EMERGING 2012

The Fourth International Conference on Emerging Network Intelligence

ISBN: 978-1-61208-239-4

September 23-28, 2012

Barcelona, Spain

EMERGING 2012 Editors

Michael D. Logothetis, University of Patras, Greece

Tulin Atmaca, IT/Telecom&Management SudParis, France

EMERGING 2012

Forward

The Fourth International Conference on Emerging Network Intelligence (EMERGING 2012) held on September 23-28, 2012 in Barcelona, Spain, constituted a stage to present and evaluate the advances in emerging solutions for next-generation architectures, devices, and communications protocols. Particular focus was aimed at optimization, quality, discovery, protection, and user profile requirements supported by special approaches such as network coding, configurable protocols, context-aware optimization, ambient systems, anomaly discovery, and adaptive mechanisms.

Next-generation large distributed networks and systems require substantial reconsideration of existing 'de facto' approaches and mechanisms to sustain an increasing demand on speed, scale, bandwidth, topology and flow changes, user complex behavior, security threats, and service and user ubiquity. As a result, growing research and industrial forces are focusing on new approaches for advanced communications considering new devices and protocols, advanced discovery mechanisms, and programmability techniques to express, measure, and control the service quality, security, environmental and user requirements.

We take here the opportunity to warmly thank all the members of the EMERGING 2012 technical program committee as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and efforts to contribute to the EMERGING 2012. We truly believe that thanks to all these efforts, the final conference program consists of top quality contributions.

This event could also not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the EMERGING 2012 organizing committee for their help in handling the logistics and for their work that is making this professional meeting a success. We gratefully appreciate to the technical program committee co-chairs that contributed to identify the appropriate groups to submit contributions.

We hope the EMERGING 2012 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in emerging technologies.

We hope Barcelona provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

EMERGING 2012 Chairs

EMERGING Advisory Chairs

Raj Jain, Washington University in St. Louis, USA
Michael D. Logothetis, University of Patras, Greece
Tulin Atmaca, IT/Telecom&Management SudParis, France
Phuoc Tran-Gia, University of Wuerzburg, Germany
Nuno M. Garcia, Universidade Lusófonas de Humanidades e Tecnologias, Lisboa, Portugal

EMERGING Industry Liaison Chairs

Krishna Murthy, Global IT Solutions at Quintiles - Raleigh, USA
Tadashi Araragi, Nippon Telegraph and Telephone Corporation – Kyoto, Japan
Robert Foster, Edgemount Solutions - Plano, USA

EMERGING Research Chairs

David Carrera, Barcelona Supercomputing Center (BSC) / Universitat Politecnica de Catalunya (UPC), Spain
Daniel Scheibli, SAP Research, Germany

EMERGING 2012

Committee

EMERGING Advisory Chairs

Raj Jain, Washington University in St. Louis, USA
Michael D. Logothetis, University of Patras, Greece
Tulin Atmaca, IT/Telecom&Management SudParis, France
Phuoc Tran-Gia, University of Wuerzburg, Germany
Nuno M. Garcia, Universidade Lusófonas de Humanidades e Tecnologias, Lisboa, Portugal

EMERGING 2012 Industry Liaison Chairs

Krishna Murthy, Global IT Solutions at Quintiles - Raleigh, USA
Tadashi Araragi, Nippon Telegraph and Telephone Corporation – Kyoto, Japan
Robert Foster, Edgemount Solutions - Plano, USA

EMERGING 2012 Research Chairs

David Carrera, Barcelona Supercomputing Center (BSC) / Universitat Politecnica de Catalunya (UPC), Spain
Daniel Scheibli, SAP Research, Germany

EMERGING 2012 Technical Program Committee

Mercedes Amor-Pinilla, University of Málaga, Spain
Richard Anthony, University of Greenwich, UK
Tadashi Araragi, NTT Communication Science Laboratories - Kyoto, Japan
Tulin Atmaca, IT/Telecom&Management SudParis, France
M. Ali Aydin, Istanbul University, Turkey
Robert Bestak, Czech Technical University in Prague, Czech Republic
Christian Blum, Universitat Politècnica de Catalunya - Barcelona, Spain
Indranil Bose, Indian Institute of Management – Calcutta, India
Mieczyslaw Brdys, University of Birmingham, UK
Chin-Chen Chang, Feng Chia University - Taichung, Taiwan
Chi-Hua Chen, National Chiao Tung University, Taiwan, R.O.C.
David Chen, University of Bordeaux – Talence, France
Dong Ho Cho, Korea Advanced Institute of Science and Technology (KAIST) - Daejeon, Republic of Korea
Carl James Debono, University of Malta, Malta
Frank Doelitzscher, Furtwangen University, Germany
Rolf Drechsler, University of Bremen, Germany
Jean-Michel Dricot, Université Libre de Bruxelles, Belgium
Dimitris Drikakis, Cranfield University, UK
Kamini Garg, University of Applied Sciences Southern Switzerland - Lugano, Switzerland

Copyright Information

Table of Contents

Improving Performance of Multithreaded Scalar Architectures for Embedded Microcontrollers

Horia V. Căpriţă
Department of Computer and Electronic Engineering
"Lucian Blaga" University of Sibiu
Sibiu, Romania
e-mail: horia.caprita@ulbsibiu.ro

Mircea Popa
Faculty of Automation and Computers
"Politehnica" University of Timişoara
Timişoara, Romania
e-mail: mircea.popa@ac.upt.ro

Ioan Z. Mihu
Department of Computer and Electronic Engineering
"Lucian Blaga" University of Sibiu
Sibiu, Romania
e-mail: ioan.z.mihu@ulbsibiu.ro

Abstract – **The primary aim followed in the development of computing systems is increasing the overall performance. The market always requires faster, more efficient and powerful products regardless of the applications that are used: high-end applications, telecommunications, automotive, low-power embedded applications, etc. Regardless of the type of application processed, the products based on simple single core systems have already shown their limitations in obtaining the desired performance. Multicore processing is currently a way to improve the performance of a computing system. Multicore devices have become ubiquitous in everyday life and are used in all areas. In this paper we present and evaluate an interleaved multithreaded scalar architecture having limited resources which supports hardware scouting technique. We will show that the implementation of hardware scouting is viable and efficient on scalar multithreaded systems, systems that have the advantage that use limited hardware resources for efficient processing. This scalar architecture will be used in our future research as a basic processing element (Base Core Equivalent) in developing of new multicore microcontrollers that will be efficient in terms of energy consumption and processing rate.**

Keywords–multithreaded architecture; multicore microcontroller; embedded systems; energy-efficient systems.

I. INTRODUCTION

The nowadays diversity of the end-user equipments that rely on a processing unit (e.g., personal computers, mobile or automotive devices) was made possible by the integration of architectural innovative solutions. The software applications for these devices require more and more computing power regardless of hardware platform that are used. As a result, more and more companies have adopted multicore technology in order to develop more efficient processors, leading to the development of more efficient end-user equipment. This is a life cycle that could be maintained by developing new architectural types to support the next generation of software applications.

The Amdahl law shows us that the fraction of sequential code within the program limits the performance of parallel machines [1] [9]. Reducing this negative effect due to portion of sequential code of a program can be done by improving the core's performance which can affect the parallel execution performance [9]. Despite this, improvement of overall performance can be done using multicore systems. The performance of a multicore system is $\sqrt{n}$, where n is the number of cores [2].

In embedded systems design it applies the same laws used in designing of general purpose systems. Fulminant development of embedded applications which demand massive calculations, led to the concept of high-performance embedded computing (HPEC) [10]. Moreover, embedded applications must comply with more stringent rules than HPC applications on supercomputers. The efficiency of energy of embedded applications is the first criterion used in evaluation. This efficiency can be achieved in different ways: by reducing the operating frequency, by implementing dynamic reconfigurable architectures like drowsy cache [7] [10], or by creating multicore processors based on simple processing elements having limited resources [5]. These techniques can be combined to obtain an efficient processor in terms of energy consumption and performance. These designing constraints lead to the development of energy-efficient high performance embedded computing applications (HPEEC) [10] in which, unlike the field of supercomputers (HPC), not only the raw performance is important, but the amount of energy consumed to achieve this performance.

The profiles of nowadays embedded applications are getting closer to those of general purpose applications. These applications can claim hardware capability to provide support for massive calculations (e.g., multimedia applications), can be parallel or distributed, can have real-

ISBN: 978-1-61208-239-4

time features (working with an RTOS), must be reliable (reliably constrained) and last, but not least, must be energy efficient.

Multicore embedded systems represent a solution to implement HPEEC applications. By combining a multicore system with multithreaded technology we can achieve more efficient systems in terms of processing rate and the energy consumption [6] [10].

In this paper, we will introduce a multithreaded scalar architecture that uses the hardware scouting technique [6]. This scalar architecture could be used as a basic processing element (Base Core Equivalent - BCE) for developing multicore microcontrollers that are efficient in terms of energy consumption and processing rate.

In Section II, we will present a short state of the art of the multithreaded architectures. Section III explains the principles of our proposed multithreaded architecture. In Section IV and V, we will present the results and the conclusions of our research.

II. Related Work

Multithreaded architectures (MT) allow execution of instructions fetched from multiple threads at a time. This paradigm is based on resource sharing when more threads compete for these resources. This thread overlapping influences in a positive way the overall processing performance [15]. Threads competition also leads to unfair resources sharing between them which can affect the processing rate of a single thread at a time. A worst case occurs when we run threads with a low hit rate in cache. Such a thread can block the reorder buffer because the instructions that follow after a data cache miss event will not be issued or will not be able to complete due the stalls imposed by the memory operations with high latencies. Moreover, there may be critical resources assigned to this thread, leading to “starvation” and stalling of other independent threads of current thread. The thread stalling during a memory access limits the exploitation of memory level parallelism (MLP). As a result, the overall performance of multithreaded processor may be affected.

The literature contains references to methods that try to reduce these negative effects. Hardware scouting [6] consists of launching a hardware thread (invisible in software) that runs in front of the main application thread. The main role of this thread is to bring data and instructions, which are necessary for the execution, in internal caches. A “load miss” event will start this hardware thread. In this case, the multithreaded architecture will create a checkpoint with the current state of the thread and then will continue to execute instructions that follow after the load, until the requested data is brought from system memory. At this point, the processor will restore the program state based on the last saved checkpoint and will rerun instructions that follow after the load using new conditions. The advantage of this method is that the instructions re-launched in the second step will be already available in the primary caches because were extracted from memory by hardware thread. Chaudry *et al.* [6] show an increase of 40% of the CPU performance when using an L2 cache of 512 KB. Using the hardware thread can be considered as a sophisticated mechanism of prefetching.

Another version of hardware scouting is presented in [12]. Ramirez *et al.* propose a method to exploit the memory level parallelism (MLP) to increase the performance of Simultaneous Multithreaded processors (SMT): Run-ahead Threads (RaT). Run-ahead execution is a method in which data and instructions are speculatively mapped in caches [11]. Ramirez *et al.* have developed the RaT method which is a strategy of fetching used to increase the performance of memory-bound threads without affecting the quantity of instruction level parallelism exploited in those threads. This method is applied to threads that could stall due to high latency memory operations. The appearance of a high-latency load instruction determines the owner thread to become a runner-ahead thread. This thread will become speculative and will use for a short time some hardware resources, so that other threads will not be limited to getting access to the CPU. At the same time, the prefetching operations from other threads will increase the degree of memory level parallelism, too. The SMT model used in simulations allows resource sharing [12]. More threads coexist in the pipeline and share structures like instruction queues, reorder buffer, physical registers, functional units and caches. This method has the advantage that will increase the performance of a simultaneous multithreaded processor by speculating load-miss events coming from different threads.

In our previous paper [4] we showed that the multithreaded model can be effective when it is implemented on a scalar processor. Our model was inspired by the models presented in [13] and [14]. It was adapted to be used in embedded multicore systems [5] in which energy savings can be made by simplifying the underlying architectural model of a BCE's.

This research focuses on scalar multithreaded architectures (having limited resources) that are capable to adapt the hardware scouting method used by others on superscalar multithreaded processors [6]. We will show that the implementation of this technology is viable and efficient on multithreaded scalar systems, systems that have the advantage that use limited hardware resources for efficient processing.

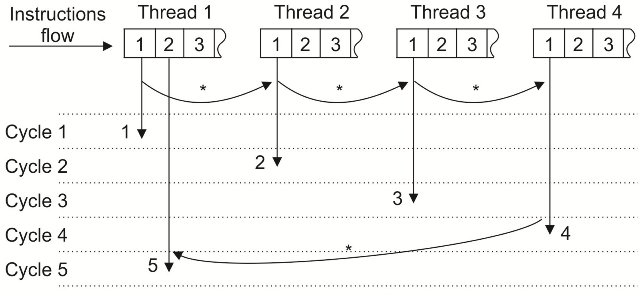
III. HSSS-IMT Multithreaded Scalar Architecture and Hardware Scouting

In this paper, we present and evaluate an interleaved multithreaded scalar architecture having limited resources which supports hardware scouting technique (HSSS-IMT architecture) [6]. This scalar architecture can be used as a basic processing element (Base Core Equivalent - BCE) in multicore microcontroller’s development process; this basic processor could be efficient in terms of energy consumption and processing rate.

The HSSS-IMT architecture is based on the SS - IMT architecture presented in [4]. SS-IMT is a multithreaded architecture based on a scalar processor [3]. SS-IMT is modified to interleave instructions that are coming from different threads in order to execute them using the same pipeline.

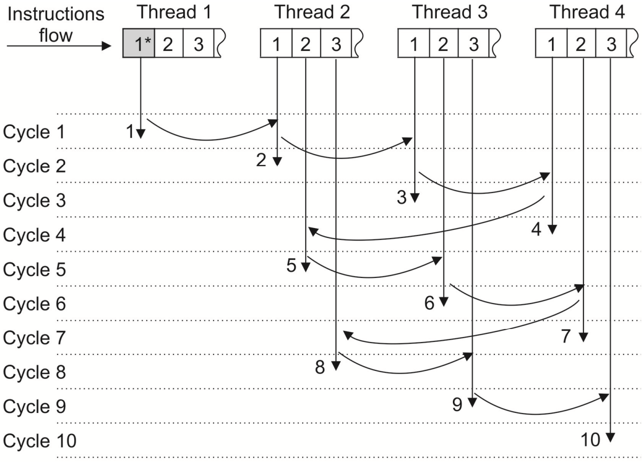
Interleaved Multithreading technique (IMT) is often called fine-grain multithreading [15]. The processor switches to another thread after each instruction fetch (Fig. 1). An instruction feeds the pipeline after the previous statement issued. IMT eliminates hazards control and data dependencies between instructions. Memory latencies are hidden by the scheduler. The thread that has generated a latency of memory will be stalled by the scheduler; the interleaving through the pipeline will continue just for instructions that becomes from other threads (Fig. 2). The instructions on the stalled thread will be scheduled again for execution when the memory transaction was done.

One way to increase the performance of this architecture is the hardware scouting [6]. The study presented in this paper is focused on hardware implementation and adaptation of hardware scouting technique to the scalar multithreaded architecture SS-IMT. When a load-miss (long latency) event occur, the new architecture, called Hardware Scouting SS-IMT (HSSS-IMT), continues to use the fetch algorithm that applies Round-Robin on all available threads, including the thread that generated the long-latency event.



* RR context switching

Figure 1. Round Robin context switching in SS-IMT.

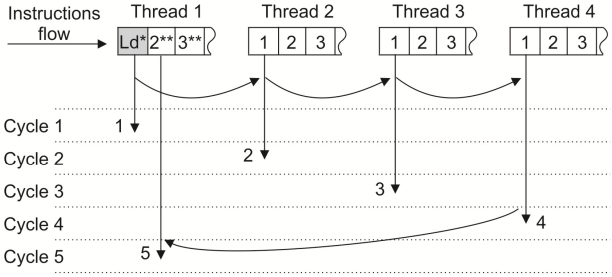


* Thread 1 is stalled by a long latency instruction

Figure 2. Stalled thread in SS-IMT.

Moreover, when a long - latency event occurs, HSSS-IMT will create a checkpoint that will contain the status of all architectural registers of the thread that generated the memory latency. Unlike SS-IMT, HSSS-IMT processor will not block the fetching of instructions belonging to the thread which generates the latency. These instructions (following load-miss instruction) become pseudo-executed instructions and will continue to be scheduled for execution under Round-Robin algorithm (Fig. 3). Thus, in the HSSS-IMT processor there are no stalled threads.

When a memory transaction ends, the instructions which follow after the Load and were pseudo-executed will be flushed from pipeline, while the thread context will be restored from the last saved checkpoint. Check pointing mechanism is implemented in "Check pointing and Flush" block (Fig. 4). Starting from now, the instructions of the thread that generated the long latency event will be rerun through all phases of the instruction cycle (instruction fetch, decode, execute, write back) and using the right context. The advantage of the hardware scouting method will be that, this time, the rescheduled instructions have been already in the instruction cache and their operands could be already loaded into the data cache. Such an instruction will be executed with maximum speed allowed by this architecture.



* Load miss instruction (Long latency instruction)
** Instructions fetched after Load (Pseudo executed instructions)

Figure 3. HSSS-IMT: when a load misses, the next instructions will be fetched, and a new checkpoint is created.

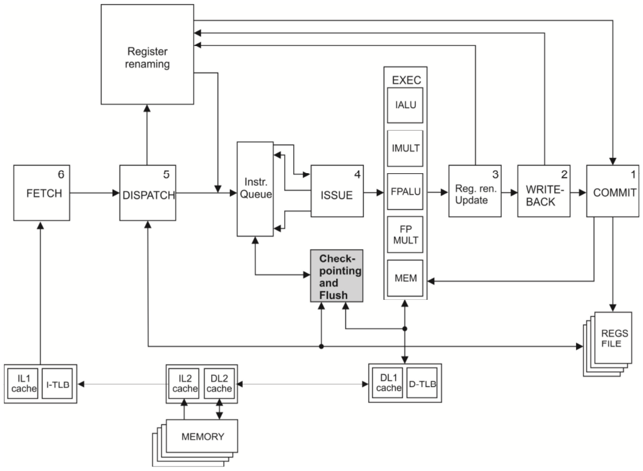


Figure 4. HSSS-IMT architecture.

IV. RESULTS

Being a multithreaded environment, we used benchmarks as threads for our HSSS-IMT evaluation. By porting SPEC CPU2000 [8] suite we obtained 7 integer benchmarks and 9 floating point benchmarks that we have used in simulations (Table I). These benchmarks represent the threads concurrently processed in our architectures (e.g., the evaluation of 8-contexts HSSS-IMT processor was done using 8 concurrent benchmarks as inputs).

TABLE I. SPEC CPU2000 BENCHMARKS PORTED TO HSSS-IMT

SPEC CPU2000 Benchmark type	Benchmark name
Integer	bzip2.ss, gcc.ss, gzip.ss, mcf.ss, parser.ss, twolf.ss, vortex.ss
Floating point	ammp.ss, applu.ss, apsi.ss, art.ss, equake.ss, mesa.ss, mgrid.ss, swim.ss, wupwise.ss

We created some groups of benchmarks named "thread sets". For example, Gr-2TH-1 thread set contains bzip2.ss and ammp.ss benchmarks, while Gr-8Th-1 thread set contains bzip2.ss, gcc.ss, gzip.ss, mcf.ss, ammp.ss, applu.ss, apsi.ss and art.ss benchmarks. We defined three versions of HSSS-IMT architectures: two hardware contexts (2 threads), four hardware contexts (4 threads) end eight hardware contexts (8 threads). We extended the number of sets for D-cache and I-cache accordingly with the number of threads that we've used in simulations. We evaluated the performance of HSS-IMT related to the performance of the basic architecture SS-IMT. Each result was obtained running 100 millions of instructions/benchmark on SS-IMT and HSSS-IMT architectures.

In [4], we proposed a multithreaded model based on Simple Scalar (SS) architecture [3]. Our model, named Simple Scalar Interleaved Multithreaded architecture (SS-IMT), was inspired by multithreaded architectures presented in [13] and [14]. It was adapted to be used in embedded multicore systems [5]. In Figure 5 and Figure 6 we depict the average performance of the SS-IMT architectures with 2, 4 and 8 contexts (threads). Each average value has been obtained using instruction and data caches with 32, 64, 128 and 256 KB per context.

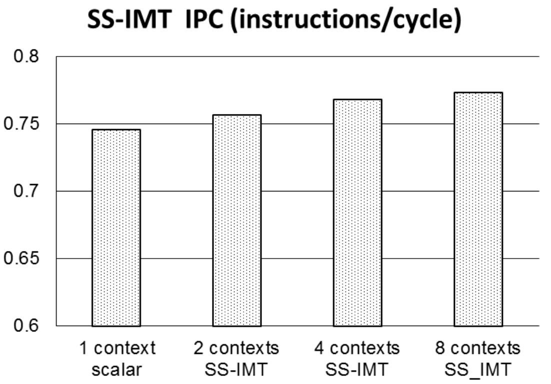


Figure 5. The overall performance (IPC) of SS-IMT architectures [4].

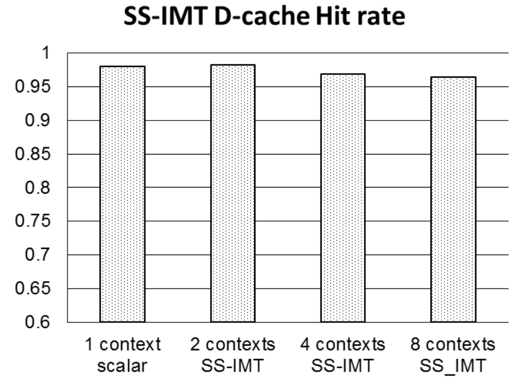


Figure 6. The overall performance (D-cache hit rate) of SS-IMT architectures [4].

As we can see from the previous figures, we note that IPC is growing together with the number of contexts (threads), while D-cache performance decreases. This decrease is due to the number of threads increasing, threads which concurrently coexist in D-cache [4].

HSSS-IMT evaluation has been done using data caches having 32, 64, 128 and 256 KB per thread. In Figure 7 we represent the performance in terms of IPC for HSSS-IMT architecture with 2 hardware contexts. For evaluation we have used 18 groups of 2 benchmarks. These groups contain benchmarks that have an integer or floating point profile or could contain benchmarks from these two categories. It could be observed that the maximum average performance is obtained using the thread set number 2 (0.8361804 IPC), while the worst average performance is given by the thread set 11 (0.784226 IPC). The difference between these two values is given by the content of the thread set (the profiles of the SPEC CPU2000 benchmarks). The average value of the performance for this HSSS-IMT configuration is 0.8130545 IPC.

Figure 8 depicts the HSSS-IMT performance having 4 hardware contexts. For evaluation we have used 22 groups of 4 benchmarks. Like in previous case these groups contain mixed benchmarks and the performance grows together with the dimension of caches.

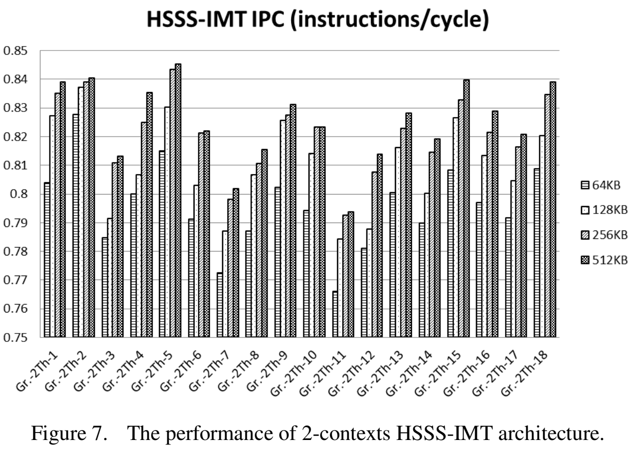


Figure 7. The performance of 2-contexts HSSS-IMT architecture.

ISBN: 978-1-61208-239-4

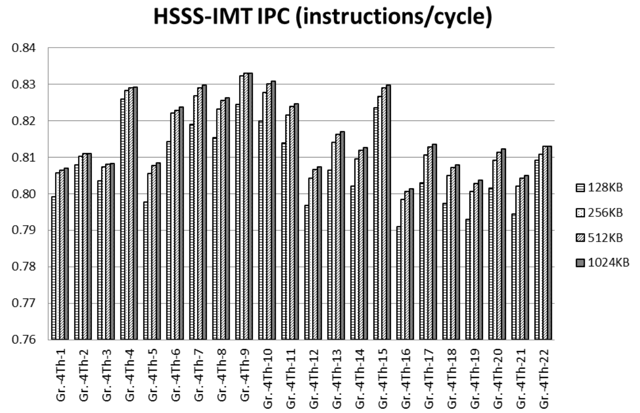


Figure 8. The performance 4-contexts HSSS-IMT architecture.

It can observe that the maximum average performance is obtained using the thread set number 9 (0.8492511 IPC), while the worst average performance is given by the thread set 19 (0.8001177 IPC). The average value of the performance for this HSSS-IMT configuration is 0.817238679 IPC.

In Figure 9, we represent the performance of HSSS-IMT architecture having 8 hardware contexts obtained with a worst case combination of benchmarks. The average value of the performance for this HSSS-IMT configuration is 0.820006152 IPC.

By comparing these results we can see that the performance of HSSS-IMT architecture is superior to the performance of SS-IMT regardless of the number of hardware contexts that are implemented.

From Figure 10, we can observe that while the number of HSSS-IMT contexts is growing, the average performance of this architecture is growing too. The global performance of HSSS-IMT architecture (0.816766446 IPC) is greater than the global performance of the SS-IMT architecture (0.766081157 IPC). In Figure 11 we represent the data cache hit rate for SS-IMT and HSSS-IMT architectures. While the number of contexts is growing we obtain a better performance that is in opposite to the results obtained on SS-IMT (Fig. 6).

This additional performance comes from the implementation of the hardware scouting technique that ameliorates the negative effect of a load-miss event [6]. The number of scouting instructions after a load-miss event will vary and depends on the memory latency.

The global performance of HSSS-IMT architecture related to the SS-IMT performance is depicted in Figure 12 and Figure 13.

Figure 12 represents a synthesis of Figure 5 and Figure 10 and depicts the IPC parameter of SS-IMT and HSSS-IMT architectures. Also, Figure 13 represents a synthesis of Figure 6 and Figure 11 and depicts the hit rate of data caches for these two architectures. The best results are obtained when we increase the number of the hardware contexts (number of threads). Average performance of this new architecture (approx. 0.816 IPC) is greater than average performance of SS-IMT (approx. 0.766 IPC) with approximately 5%. Data cache hit rate grows from approximately 0.97% in SS-IMT to 0.99% in HSSS-IMT.

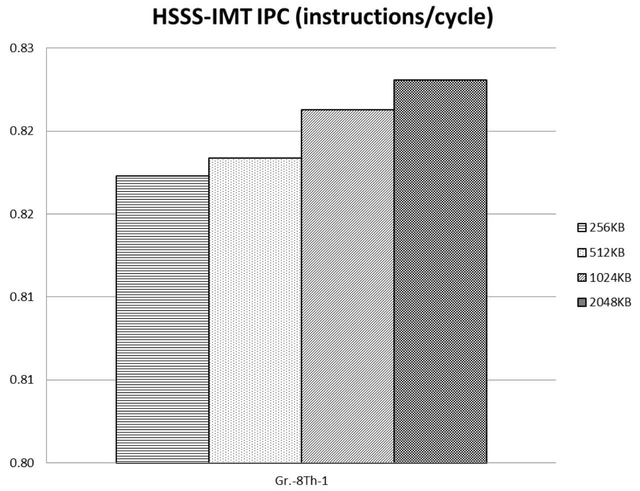


Figure 9. The performance 8-contexts HSSS-IMT architecture.

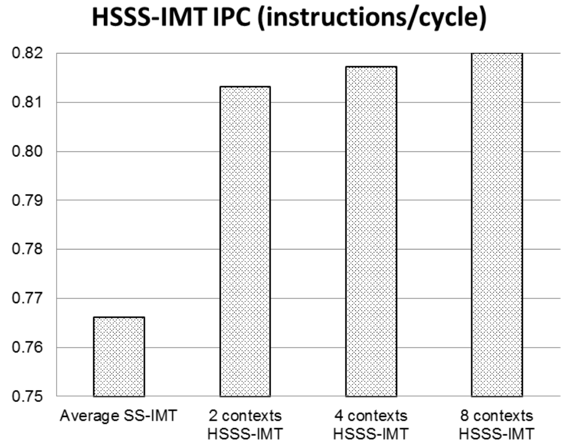


Figure 10. The overall performance (IPC) of HSSS-IMT architecture.

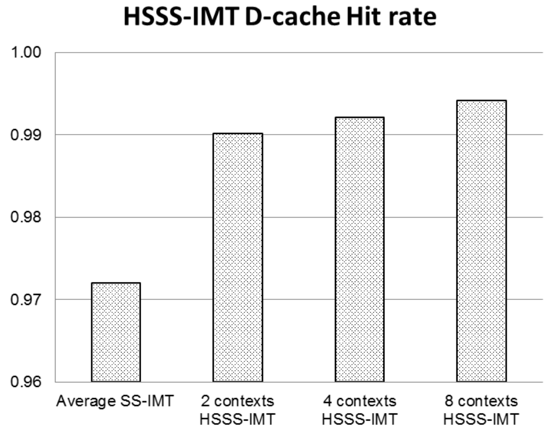


Figure 11. The overall performance (D-cache hit rate) of HSSS-IMT architecture.

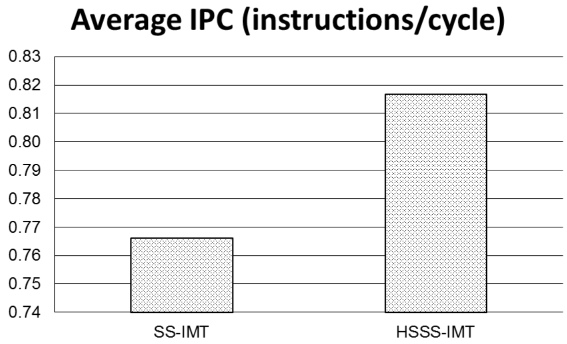


Figure 12. Average performance of HSSS-IMT vs. SS-IMT.

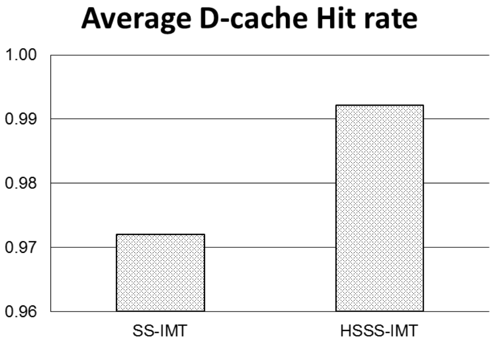


Figure 13. Average D-cache performance for HSSS-IMT vs. SS-IMT.

Implementing the hardware scouting method on our basic scalar multithreaded processor (SS-IMT) led us to a new multithreaded architecture (HSSS-IMT) that has a better performance than previous one presented in [4].

V. Conclusion And Future Work

HSSS-IMT architecture benefits of full of multithreading techniques and simultaneously of hardware scout. Moreover, this check pointing technique offers a better utilization of instruction and data cache memory by using hardware scouting. From Figure 11 we can see that average data cache hit rate of HSSS-IMT is greater with over 2% than that obtained on SS-IMT. This rising is very important for us. By implementing the multithreading technique on a basic scalar processor gave us a worse performance related to this data cache hit rate parameter [4] [5]. These performances of HSSS-IMT show us that there exists the possibility to use this kind of architectures to integrate it in bigger structures in order to create multicore processors. These multicore processors could be used in embedded systems because they contain limited resources (e.g., they have a scalar configuration) and they are able to manage more threads useful to run an RTOS.

Although investigating the power consumption of this architecture wasn't a purpose of this paper, we can anticipate that this type of architecture could be used to create low power energy BCEs due to the simplicity of the processing element [6] [11] [12].

The HPEEC domain should be sustained by developing novel transistor technologies in order to reduce the power consumption but, in the same time, the researchers have to concentrate on how to create new scheduling techniques, basic processing elements or revolutionary memory design.

Acknowledgment

This work was supported by Continental Automotive Systems Sibiu, Romania, grant no. 550/25.02.2010.

References

[1] G.M. Amdahl, "Validity of the single-processor approach for achieving large-scale computing capabilities", Proc. Am. Federation of Information Processing Societies Conf., AFIPS Press, 1967, pp. 483-485.

[2] S. Borkar, "Thousand core chips - a technology perspective", Proc. of the 44th annual Design Automation Conference (DAC 07), 2007, pp. 746-749.

[3] D. Burger and T.M. Austin, "The SimpleScalar tool set, version 2.0", ACM SIGARCH Computer Architecture News, vol. 25, Issue 3, June 1997, pp. 13-25.

[4] H.V. Căpriţă and M. Popa, "Design methods of multithreaded architectures for multicore microcontrollers", Proc. of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011), Timisoara, Romania, 2011, pp. 427-432.

[5] H.V. Căpriţă and M. Popa, "Multithreaded peripheral processor for a multicore embedded system", Applied Computational Intelligence in Engineering and Information Technology, Springer Berlin Heidelberg, 2012, pp. 201-212.

[6] S. Chaudry, P. Caprioli, S. Yip and M. Tremblay, "High performance throughput computing", IEEE Micro, vol. 25, Issue 3, May 2005, pp. 32-45.

[7] K. Flautner, N. Kim, S. Martin, D. Blaauw and T. Mudge, "Drowsy caches: simple techniques for reducing leakage power", ISCA '02 Proc. of the 29th annual international symposium on Computer architecture, vol. 30, Issue 2, 2002, pp. 148-157.

[8] J.L. Henning, "SPEC CPU2000: Measuring CPU Performance in the New Millennium", Journal of Computer, vol. 33, Issue 7, July 2000, pp. 28-35.

[9] M.D. Hill and R.M. Marty, "Amdahl's Law in the Multicore Era", Journal of Computer, vol. 41, Issue 7, July 2008, pp. 33-38.

[10] A. Munir, S. Ranka and A. Gordon-Ross, "High Performance Energy Efficient Multicore Embedded Computing", IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 4, 2012, pp. 684-700.

[11] O. Mutlu, J. Stark, C. Wilkerson and Y.N. Patt, "Runahead execution: an alternative to very large instruction windows for out-of-order processors", Proc. of International Symposium on High-Performance Computer Architecture (HPCA 9), 2003, pp. 129-140.

[12] T. Ramirez, A. Pajuelo, O.J. Santana and M. Valero, "Runahead threads to improve SMT performance", Proc. of 14th International Conference on High-Performance Computer Architecture (HPCA 14), 2008, pp. 149-158.

[13] D.M. Tullsen and J.A. Brown, "Handling long-latency loads in a Simultaneous Multithreading processor", Proceedings of the 34th International Symposium on Microarchitecture, December 2001 (MICRO 34), pp. 318-327.

[14] D.M. Tullsen, S.J. Eggers, J.S. Emer, H.M. Levy, J.L. Lo and R.L. Stamm, "Exploiting choice: instruction fetch and issue on an implementable Simultaneous Multithreading processor", Proc. of the 23rd Annual International Symposium on Computer Architecture, Philadelphia, PA, 1996, pp. 191-202.

[15] T. Ungerer, B. Robic and J. Silc, "A survey of processors with explicit multithreading", ACM Computing Surveys, vol. 35, no. 1, March 2003, pp. 29-63.

Optimization of Proxy Caches using Proxy Filters

Fabian Weber, Marcel Daneck, Christoph Reich
Hochschule Furtwangen University
78120 Furtwangen, Germany
{fabian.weber, marcel.daneck, christoph.reich}@hs-furtwangen.de

***Abstract*— Web proxy caches are a widely used tool to reduce the network load by storing often needed web pages. Increasing amounts of data transferred require new intelligent caching strategies with smarter replacement algorithms to predict, which web documents will be requested most likely in the future. However, these algorithms are relatively complex and require a lot of computing power. This paper describes an approach to design more intelligent and efficient web caches by adding a document filter that decides whether a document should be cached or whether it should be ignored in order to save disk space. The filter uses server connection information for its decision. The evaluation shows a reduction of required cache space of almost 90% compared to a traditional proxy cache.**

Keywords - Proxy; Cache; Filter; Replacement-Algorithm; LRU; LRU-threshold; LRU-MIN; SIZE; Log2(SIZE); Hybrid; MIX; GreedyDual-Size

I. INTRODUCTION

Whenever multiple users have to share a single internet connection, bandwidth bottlenecks are imminent. This problem occurs in the private as well as business sector. In the latter case, bottlenecks often harm productivity, as the employees have to wait for websites, documents etc. to be loaded. To circumvent this problem, a common practice is to use a web proxy cache server. Proxy caches store web documents that are frequently requested by web users to avoid repeated downloads of the same information from the originating web server and therefore reduce bandwidth utilization. Typically, this server is located in the local network and avoids WAN (Wide Area Network) bottleneck at the edge server.

However, the utilization of a proxy cache introduces a few difficulties when working with large amounts of data transferred [1]. The main problem is that the cache can only store a limited amount of documents, because of the limited disk space available. To cope with this problem, more intelligent cache replacement algorithms are needed to increase the efficiency of the cache. These algorithms are very complex and require a high amount of computational power, effectively limiting the efficiency of a proxy cache not only by disk space, but also by CPU power.

However, some of the documents downloaded need caching more than others. For example, files that are served with a speed almost as fast as the proxy server's connection do not benefit as much from caching as files transferred from a very slow originating server. A filter can ensure the files that benefit mostly are more likely to be cached and to remain in the cache.

This paper describes the basis to proxy caches in Section II, outlines the state of the art and technology in Section III and shows a new way to increase the efficiency of proxy caches without having to use complex replacement algorithms by filtering the web documents in Section IV. In Section V, the new proxy filter is evaluated and a conclusion can be found in Section VI.

II. TRADITIONAL PROXY CACHES

Caching is divided into three main areas: *Client-Caching* [2] is performed at the user's own system by the browser. *Server-Caching* (or *Reverse-Proxy-Caching*) [2] on the other hand is accomplished by a remote server. This technique is normally used to reduce the work load of a web server. The proxy server is therefore located in the network of the originating web server. The performance improvements are most notably with web servers that handle complex dynamic websites. The last area is *Proxy-Caching*. Here, the proxy server is located at the client's subnet, ensuring high bandwidth and low latency for this connection. Every document request of the client is sent through the proxy server. This enables the server to cache frequently accessed documents and deliver them to multiple users through the local network infrastructure, rather than through the relatively slow internet connection [3].

The utilization of proxy caching has a wide range of advantages: a) Internet users will benefit from faster page load times through a higher bandwidth and therefore have a better web experience. b) The internet-infrastructure can be improved by decreasing the outbound network traffic, effectively increasing the total network performance. Since large enterprises often buy their internet connection volume-based, reduction of the internet traffic can also save expenses. c) Web servers can benefit from proxy caching, because the work load on these servers is reduced [3]. This leads to higher performance without the need of hardware upgrades.

There are some important issues to be considered when using a proxy server:

- Even though mass storage is not very expensive, proxy cache servers do have space limitations. When the available disk space is filled, a replacement algorithm is utilized to decide which documents can be evicted in order to make room for new documents. Therefore, an optimal algorithm must be chosen depending on how clients are using the web. Several of these algorithms will be discussed in the related work section of this paper.

- Another requirement for a cache server is to ensure the consistency of the stored documents. This can be realized using one of two methods: Either the cache server asks the web server, whether the cached document has been changed in the meantime, or the web server can inform the cache server about a change of the document. The last method is not very popular, since there are no well-defined standards and it is much harder to implement.
- A last possible feature of a proxy cache is to predict the user's behavior and to pre-load documents the user is likely to request next.

This paper focuses on a method to improve the efficiency of proxy caches by using the concept of content filtering for proxy caches (proxy filters) explained in Section IV.

III. Related Work

Until now, optimization efforts in the field of proxy caching are mostly aimed to improve the underlying replacement algorithm. Such an algorithm is applied whenever a new document has to be stored while the cache storage is already full. Next, some of these replacement algorithms will be shown and analyzed.

One of the most well-known replacement algorithm is *LRU* (least recently used) [4]. As the name suggests, this algorithm always evicts the least recently used document from the cache. Therefore, a simple list is used. Upon request, a document is moved to the top, while the document to be deleted is taken from the bottom of the list. This procedure also explains the very low complexity of this algorithm at $O(1)$ [5]. The major downside of this algorithm is its simplicity of predicting which document will be requested in the future and therefore the hit rate of cached documents is rather low. Another downside is the weakness to calculate the cost of caching a requested document. This means, a large downloaded and cached document will overwrite many small and maybe more frequently used websites. Many important websites are replaced by one rather useless document.

To take countermeasures against these problems, some LRU derivatives were developed. One of these derivatives is *LRU-threshold* [6], which supports the definition of a maximum document size. Documents that exceed the given size threshold are never cached (not even if storage space is left). Apart from that, LRU-threshold acts like LRU.

SIZE [7] and *Log2(SIZE)* [7] represent two algorithms that use the document size as their primary caching decision. While SIZE always evicts the largest document first, Log2(SIZE) groups the documents by a logarithmic value of their size. Within a group, the least frequently used document is evicted (using LRU). Both of these algorithms have a complexity of $O(\log(n))$ [5].

Another popular LRU derivative is *LRU-MIN* [6]. This algorithm replaces larger documents earlier than smaller documents. Therefore, whenever a new document of size S has to be cached, all documents with size greater or equal to S are grouped. Within this group, the document is selected with LRU. If no documents remain with size $\geq S$, $S/_2$ is used as selector (then $S/_4$ and so on). The disadvantage of the algorithm is the inability to consider the cost of a document. On caches with very large storage spaces, a high amount of computing power will be required to utilize this algorithm as it has a very high complexity of $O(n)$ [5].

All the aforementioned algorithms assume that large documents (like file downloads) are less frequently requested than small documents (i.e., websites) and therefore less important to cache. This assumption may have been true a few years ago, but a study in [8] suggests an oncoming change in user behavior. With Web 2.0 and media services like YouTube even large documents (i.e., videos) will be requested frequently. Eventually, these files are also eligible for caching.

In [9], Wooster and Abrams introduce the *Hybrid* algorithm, designed to reduce the document access delay. Therefore, the algorithm considers the round trip time (RTT), the bandwidth between proxy server and originating server as well as the quantity of requests since a specific document has been stored into the cache. Using these parameters, a utility value is calculated for each document in the cache. The document with the lowest utility value finally gets replaced. This algorithm is the basis of the *MIX* algorithm, developed by Niclausse, Liu and Nain [10]. In MIX, the time since the last access of a document in the cache is added to the formula, thus introducing a possibility to remove obsolete documents like LRU does. Different to LRU, however, is that all characteristic parameters of the Hybrid algorithm are considered, too. Both methods provide benefits when documents from very slow servers are fetched. These web documents produce a very high utility value, courtesy of the low server bandwidth and therefore stay in the cache for a relatively long time. Because these files would normally be served very slowly, the performance gain is very high. On the other hand, documents that are on fast servers will be saved as well (even if the remote server speed is almost as high as the request speed from the proxy cache). These documents will ultimately be evicted in the near future, but at first they will get cached and replace other, more important documents. Additionally, both algorithms have a relatively high complexity of $O(\log(n))$ [5].

Cao and Irani introduced the *GreedyDual Size* (GDS) algorithm in [11], which is an improvement of Young's *GreedyDual* [12]. The GDS algorithm calculates the cost to cache for each document. Key parameters are the connection time, access time, transfer time and document size. Whenever a document has to be evicted, the file with the lowest value is deleted, like with Hybrid and MIX. GDS additionally incorporates a LRU-like behavior by subtracting the value of an evicted document from the values of all remaining documents. If a document is requested again while it is still remaining in the cache, its value has to be restored. This way, less frequently requested documents gradually lose their value and get evicted. The major benefit of GDS is clearly the consideration of caching costs. Therefore, large documents originating from fast servers get a relatively low value and are deleted shortly. If the connection speed of the remote server is slow, the documents are rated with a high value – even though they are relatively large – and stay in the cache for a longer time. Additionally, small documents are also treated the same way: if they are downloaded from a very fast remote server, the utility value of these documents is low (because the performance improvement of caching these documents is very low). Ultimately, these documents will be evicted soon and do not have to be kept for an unnecessary amount of time. The disadvantage of this algorithm on the other hand is the high complexity of $O(\log(n))$ [5].

In conclusion, the utilization of a complex replacement algorithm like Hybrid, MIX or GreedyDual Size results in a high hit/miss-ratio and a reasonable good selection of documents to be evicted, but with the need of high computing power (especially with large caches) caused by the high complexity of these algorithms. Furthermore, none of these algorithms takes into consideration whether the caching of a specific document entails a performance improvement in the first place.

In Table 1, an overview of the aforementioned algorithms is shown.

TABLE 1: OVERVIEW OF SELECTED REPLACEMENT ALGORITHMS [5]

Replacement algorithm	Relevant keys	Complexity
LRU	Time since last access	$O(1)$
LRU-threshold	File size Time since last access	$O(1)$
LRU-MIN	File size Time since last access	$O(n)$
SIZE	File size	$O(\log(n))$
Log2(SIZE)	File size	$O(\log(n))$
Hybrid	File size Round-Trip-Time Bandwith between proxy and server Number of hits	$O(\log(n))$
MIX	File size Round-Trip-Time Bandwith between proxy and server Number of hits Time since last access	$O(\log(n))$
GreedyDual Size	File size Connection time Access time Transfer time	$O(\log(n))$

IV. PROXY FILTER

Current proxy caches generally only limit the maximum document size. Documents that exceed this size are never cached while documents smaller than the maximum size are always cached and replace other documents if the storage is full. Admittedly, as shown in the last chapter, a wide range of more or less smart replacement algorithms can be used to select documents for eviction, but none of these considers whether it is reasonable to store a web document in the cache or not. The assumption is that it might be better not to cache a document at all, because the performance improvement of storing a document in the cache is not worth it.

For example, dynamic web pages that take a long time to generate can be cached, while images or style sheets embedded in the site are not cached because they are static files located on the originating web server and provide very low access latency. In this specific case, transfer time is a less important criterion than access time.

A. *Concept*

The decision whether a document should be cached or not is done by a proxy filter module. Several factors influence the decision and have to be taken into account by the proxy filter:

File size: The file size is an important factor, because the cache can store a lot more small files than large files. This basically means that a large file occupies the space that otherwise very many small files would use and is therefore considered less valuable. Contrary to existing algorithms, the file size is not considered an absolute limit, but it is relativized with the other factors.

Request time: This reflects the time needed to connect to the remote server and send the request. The request time is particularly interesting because servers that operate under high load and reach their connection limit cannot react to the request in a decent time frame.

Access time: High access times are mostly a result of dynamic content, which has to be computed by the server. This is, for example, the case with Web-Content-Management systems, forums or other Web 2.0 pages. The access time is measured by the timespan between sending the request and receiving the first byte of the response.

Transfer time: Another interesting factor is the transfer time. If it is an unusual high value, caching the document may result in high performance improvements despite the eventually large file size. Documents that are fetched with a low bandwidth generally achieve a higher performance gain than documents transferred through a fast connection.

A few samples of latency distributions for web requests are shown in Figure 1. The first bar represents the download of a small static style sheet file from a heavily used server (request time is relatively high). The second element represents accessing a dynamic website. Here the request time is very low, while the access time is extremely high. This document will benefit from a great performance gain upon caching. The last bar visualizes the download of a large file. The most notable factor is the transfer time. Caching benefits for this file have to be evaluated by consideration of the server's bandwidth (the file size in relation to the transfer time).

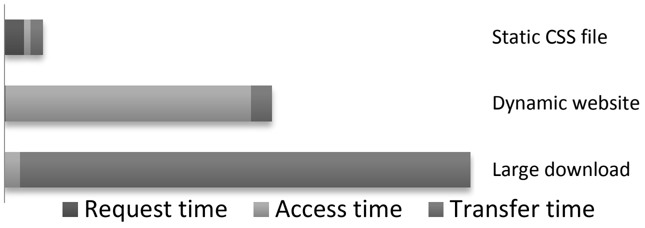


Figure 1: Exemplary latency distribution for web requests

To be able to determine, which documents should be cached, the factors mentioned above have to be weighted and summed up. Therefore, in a first step, the weighting of each factor has to be configured. The algorithm then multiplies each factor with the associated weighting and sums the (weighted) factors up. More precisely, the filter uses the formula:

$$r * w_r + a * w_a + t * w_t + s * w_s$$

where r stands for the request time, a is the access time, t the transfer time and s is the file size. w_r, w_a, w_t and w_s are the weightings for each factor. If the result is greater than or equal to zero, the document is considered relevant for caching. Documents with ratings less than zero will not be cached, because these documents would not get enough performance gain when loaded from the cache as opposed to being loaded from the internet.

Furthermore, this filter can be designed to act intelligent by setting the weightings dynamically. This way, the system would be capable of adjusting itself to changed conditions like an increase in available storage space. To implement such intelligent behavior, a background task could be set that runs at the end of the day and analyzes the hit/miss ratio, byte-hit/miss ratio etc. of the proxy cache and adjust the parameters accordingly. At the next run, this optimization job can compare the last results with the new test results and adjust the weightings again.

The proxy filter does not take the place of cache replacement algorithms. These algorithms still have to be used whenever a new document has to be saved to the already filled storage space. However, these algorithms will be used less often since documents that would get evicted again after a short timespan will never be cached in the first place.

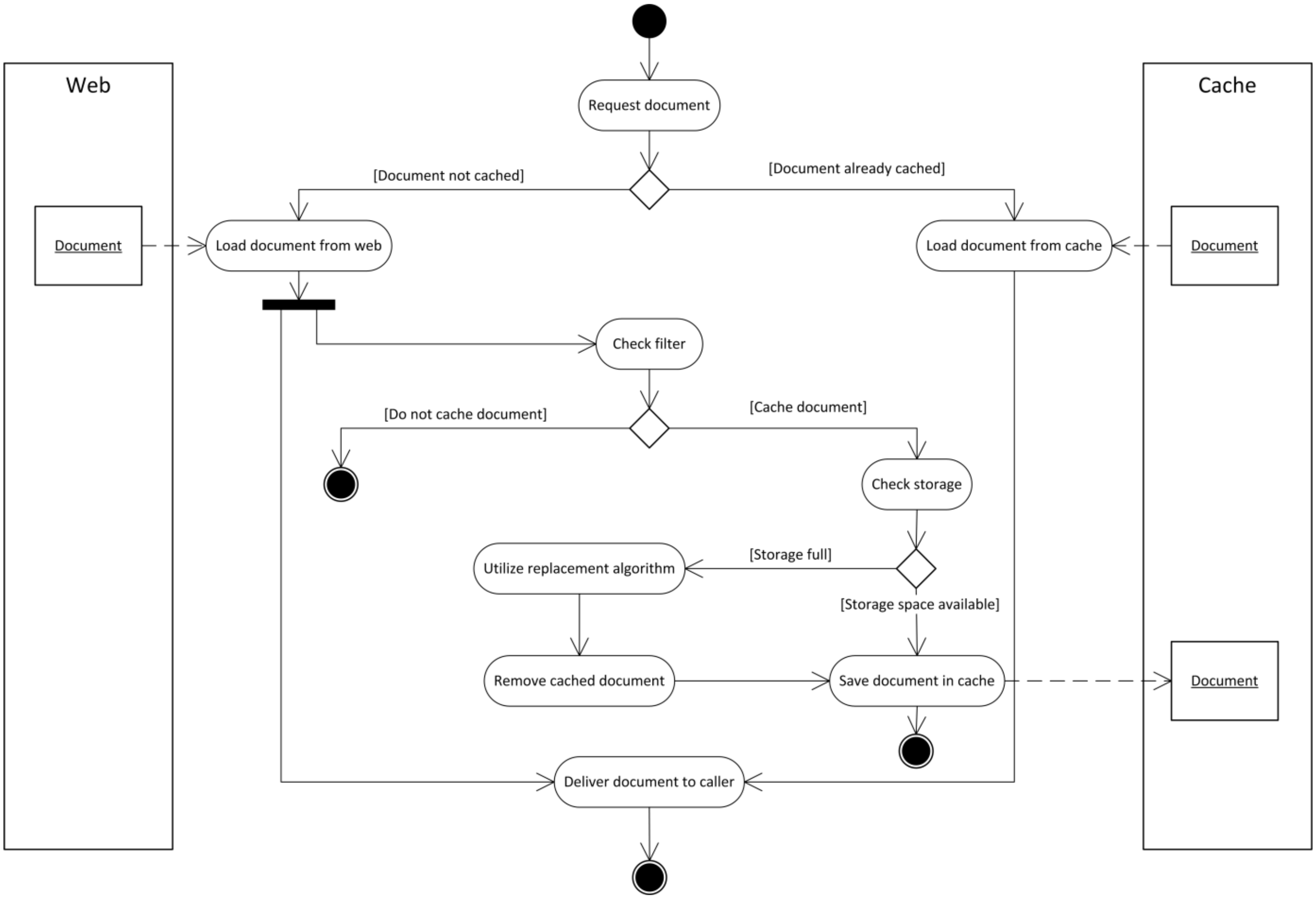


Figure 2: Activity diagram of a proxy cache with enabled filter

ISBN: 978-1-61208-239-4

The proxy filter – which is placed before the actual caching logic (see Figure 2) – improves the performance of the total proxy cache. The hit/miss ratio of the cache will not be greatly improved by utilization of a proxy filter, but the cache will work much more efficiently. One reason is that important documents are not replaced by less valuable files. Another reason is due to the lightweight algorithm for filtering, which requires much less computing power. As shown in Table 1, most of the replacement algorithms have a complexity of at least $O(\log(n))$. For big caches with many stored documents, a lot of documents have to be analyzed for replacement decision. Some of the algorithms analyze every document in the cache and therefore need a lot of compute power. The filter algorithm, however, just has to analyze the current document. The resulting complexity is $O(1)$.

Figure 2 shows how the proxy filter is integrated into the proxy cache. If a document is requested and already stored in the cache it is directly delivered. If a document is requested and not stored in the cache it is downloaded from the addressed web server logging the file size, request time, access time and transfer time. According to these parameters a decision whether this document needs to be cached or not can be reached.

V. EVALUATION

To be able to evaluate this concept, a first step was to collect proper test data. Therefore, various websites were visited, files downloaded and media streamed. Meanwhile, all web requests were logged, causing a total of 6644 data sets. Each of these data sets reflects one downloaded document of types like websites, embedded pictures, style sheets and JavaScript files, as well as video files, etc.

In total, these 6644 files take around 505 MB of space and the download time of these files (including connection time, access time and transfer time) was about 1 hour and 35 minutes.

TABLE 2: WEIGHTING OF FACTORS

Factor	Weighting
File size	-1
Request time	50
Access time	100
Transfer time	250

To evaluate the filter algorithm, for each data set the performance improvement has to be determined when it is stored in the cache. Using the filter we can decide if it is worth to store a document or if it is better not to store it and save the cache space for other documents. As mentioned above, the factors file size, request time, access time and transfer time needed to be weighted. Therefore, the weights were configured as shown in Table 2 (these values were selected experimentally).

A sample calculation in Table 3 shows, how the document rating was concluded from the factors and weights of three exemplary documents:

TABLE 3: RATINGS OF EXEMPLARY DOCUMENTS

File size [Bytes]	Request time [ms]	Access time [ms]	Transfer time [ms]	Rating
142,694	< 1	1,514	4,352	1,096,706
91,989	< 1	31	203	-38,139
10,121,411	140	250	133,693	23,333,839

The first document has a high access time as well as a high transfer time, meaning a low server bandwidth. The document rating is therefore positive and the document gets cached.

The size of the second document is even smaller than the first document, but because of its low access time and high bandwidth, the rating is negative. This document does not get cached, because caching would not bring a high performance gain (the originating server is almost as fast as the proxy cache).

Even though the third document is with almost 10 MB rather large, it is cached because of the very high transfer time indicating a server with a slow internet connection. This way, long waiting times when downloading this file are circumvented.

After applying the algorithm with the weightings from Table 2, it indicates that of the 6644 requested documents, 5682 (that is 85.5%) would have been cached. These 85.5% of files take 63 MB storage space, meaning a reduction of disk space of more than 87%.

To estimate the performance improvements achieved by caching these documents, the request time at the proxy server was set at 10 ms and access time was set to be 30 ms. In practice, these values are most likely even lower. Since the proxy cache is typically located in the physical network of the users, the bandwidth was assessed at 100 Mbit/s.

To calculate the performance gain, the total response time (request time, access time and transfer time) for each document not in the cache was summed up. For each document that was cached, a request time of 10 ms, access time of 30 ms and the transfer time according to the file size through a 100 Mbit/s connection were summed up. In total, the time required to load all documents would be barely 9 minutes. Compared to the 1.5 hours needed for the initial download this is a performance improvement of over 90%.

By changing the weightings of the factors, the ratio of storage space and performance gain can be varied. A short outlook of these variations is given in Figure 3. The chart shows clearly that the system works most efficiently while between 10% and 15% of the traffic is cached. Caching the other documents would require a huge amount of storage space while the performance improvements would be at a minimum.

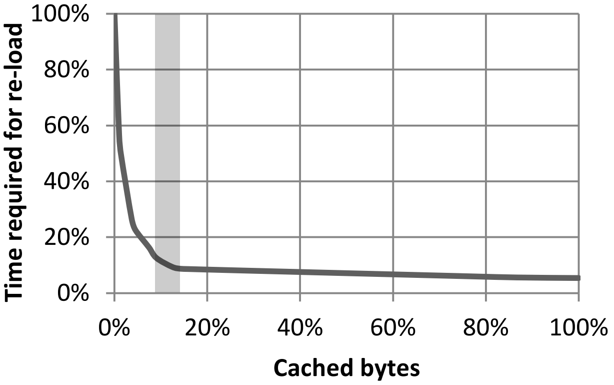


Figure 3: Performance improvement in relation to cached bytes

This chart also shows the difference between a proxy cache that uses the filter and one that does not. The proxy filter has the ability to choose the 12% of the documents that cause 90% of performance improvement by caching. A proxy cache without this filter is not able to process this information, since every document is stored, regardless of the performance gain.

VI. Conclusion

Proxy caches can utilize a wide range of cache replacement algorithms. Depending on the selected algorithm, different factors are then used to select a document for eviction whenever storage space is needed to cache a new document. However, because these algorithms only take action when a document needs to be deleted, none of them can predict whether the caching of a specific document makes sense in terms of performance improvements.

The newly introduced proxy filter fills this gap by trying to estimate the performance gain of each document upon request. Only documents that promise high performance improvements will be cached. This method highly aids the selected replacement algorithm – which can still be used without modifications – because the filter uses less computational power to execute.

As shown in the evaluation, using the proxy filter only 12% of the transferred amount of data is cached, resulting in a performance increase of over 90%. Because of this data reduction the cache has a lot less swap and the replacement algorithm is utilized less often. This improves the system responsiveness and saves resources since the replacement algorithm has to analyze every document cached to select one to evict while the filter only analyzes the current document.

References

[1] Elias Balafoutis, Antonis Panagakis, Nikolaos Laoutaris and Ioannis Stavrakakis: *Study of the Impact of Replacement Granularity and Associated Strategies on Video Caching* from: Cluster Computing, Vol. 8, No. 1, pp. 89-100, 2005

[2] Zeeshan Naseh and Haroon Khan: *Designing Content Switching Solutions*, Cisco Press, March 14, 2006

[3] Daniel Zeng, Fei-Yue Wang, and Mingkuan Liu: *Efficient Web Content Delivery Using Proxy Caching Techniques* from: IEEE Transactions on Systems, Man, and Cybernetics - TSMC , Vol. 34, No. 3, pp. 270-280, 2004

[4] Andrew S. Tanenbaum: *Modern Operating Systems*, Prentice Hall, 2nd Edition: December 6, 2001

[5] Abdullah Balamash and Marwan Krunz: *An Overview of Web Caching Replacement Algorithms* from: IEEE Communications Surveys and Tutorials - COMSUR, Vol. 6, No. 1-4, pp. 44-56, 2004

[6] Marc Abrams, Charles R. Standridge, Ghaleb Abdulla, Stephen Williams, and Edward A. Fox: *Caching Proxies: Limitations and Potentials* from: 4th International World-wide Web Conference, Dec. 1995, retrieved from: http://www.w3.org/Conferences/WWW4/Papers/155/

[7] Stephen Williams, Marc Abrams, Charles R. Standridge, Ghaleb Abdulla, and Edward A. Fox: *Removal Policies in Network Caches for World-Wide Web Documents* from: Computer Communication Review - CCR, Vol. 26, No. 4, pp. 293-305, 1996

[8] Geetika Tewari and Kim Hazelwood: *Adaptive Web Proxy Caching Algorithms*, Computer Science Group Harvard University Cambridge, Massachusetts, 2004

[9] Roland P. Wooster and Marc Abrams: *Proxy Caching That Estimates Page Load Delays* from: Computer Networks and Isdn Systems - CN, Vol. 29, No. 8-13, pp. 977-986, 1997

[10] Nicolas Niclausse, Zhen Liu, and Philippe Nain: *A New Efficient Caching Policy for the World Wide Web* from: Workshop on Internet Server Performance, June 1998

[11] Pei Cao and Sandy Irani: *Cost-Aware WWW Proxy Caching Algorithms* from: USENIX Symposium on Internet Technologies and Systems - USITS, pp. 193-206, 1997

[12] Neal Young: *The K-Server Dual and Loose Competitiveness for Paging* from: Algorithmica, Vol. 11, No. 6, pp. 525-541, June 1994

The Impact of the Floorplan of Functional units on 3D Multi-core Processors

Hyung Gyu Jeon
School of Electronics and Computer Engineering,
Chonnam National University
Gwangju, Korea
hggodman1108@gmail.com

Hong Jun Choi
School of Electronics and Computer Engineering,
Chonnam National University
Gwangju, Korea
chj6083@gmail.com

Jong Myon Kim
School of Computer Engineering and Information Technology,
University of Ulsan
Ulsan, Korea
jmkim07@ulsan.ac.kr

Cheol Hong Kim
School of Electronics and Computer Engineering,
Chonnam National University
Gwangju, Korea
chkim22@jnu.ac.kr*

***Abstract*— Interconnection delay is one of the most critical constraints in improving the performance of multi-core processors. In order to reduce the interconnection delay in the multi-core processor, 3D integration technology has been applied in designing multi-core processors. The 3D multi-core processor is composed of vertically stacked cores which are connected by through-silicon vias, leading to improved interconnection and power efficiency by reducing the physical wire length significantly. However, 3D multi-core processors have severe temperature problems caused by higher power density compared to 2D Multi-core processors. In this paper, we propose the thermal-aware floorplan schemes to solve the thermal problems in 3D multi-core processors by changing the location of functional units. According to our experimental results, the proposed floorplan schemes reduce the peak temperature by 12℃ on average with 3% performance gain.**

Keywords- Multi-core processor; 3D architecture; Temperature; Floorplan schemes

I. INTRODUCTION

Continuing advances in semiconductor technology enables increasing clock frequency, leading to the improved processor performance. Unfortunately, the increased frequency causes high power consumption [1]. Therefore, performance and power efficiency should be considered together in designing up-to-date microprocessors [2-6]. To overcome the power constraints in single-core processors, multi-core processors have been widely used. In the multi-core processors, the interconnection delay is regarded as one of the major constraints in improving the performance [7-9].

In 3D multi-core processors, multiple cores are stacked vertically and each core on different layers are connected by vertical through-silicon vias(TSVs) [10][11]. The 3D integration technology using TSVs can be a good solution in the multi-core processor because the 3D architecture has advantages in the perspective of performance and power efficiency over the 2D architecture [12]. For this reason, many researchers have focused on the 3D architecture in designing multi-core processors. However, one of the major problems in designing 3D multi-core processors is the thermal problem due to high power density. According to [13], the thermal problem is exacerbated in the 3D cases for mainly two reasons. First, the thermal conductivity of the dielectric layers between the device layers is very low compared to silicon and metal. Second, the vertically stacked multiple layers of active devices cause a rapid increase of power density. Therefore, in spite of various advantages of the 3D integration technology, it cannot be practical without proper solutions for thermal problems, because thermal problems have negative impact on the reliability of the processor [14].

Dynamic Thermal Management(DTM) techniques, which use dynamic frequency scaling(DFS), dynamic voltage scaling(DVS), clock gating, or computational migration, have been proposed to relieve the thermal problems of processors. DTM techniques keep the chip temperature under the given threshold, resulting in improved reliability [15]. Unfortunately, DTM techniques degrade the performance to reduce the temperature of the processor. In this work, we propose three thermal-aware floorplan schemes to alleviate the thermal problems of the 3D multi-core processor with little performance loss. In our previous work, we investigated thermal-aware floorplan schemes and analyzed the impact of the floorplan on the processor temperature [16]. In this work, we present more efficient floorplan schemes compared to the schemes in [16][17], leading to better performance and energy-efficiency while solving the thermal problems of 3D multi-core processors.

The rest of this paper is organized as follows: Section II describes related work and Section III presents the proposed thermal-aware floorplan schemes. Section IV describes the simulation infrastructure and methodology and Section V describes our experimental results in detail. Finally, Section VI concludes this paper.

II. RELATED WORK

A. *3D multi-core processors*

Compared to 2D multi-core processors, 3D multi-core processors have benefits in improving the performance by reducing the wire length dramatically. There are several manufacturing technologies for 3D die stacking and alignment, such as wafer-to-wafer bonding, die-to-die bonding, die-to-wafer bonding [18][19]. In the wafer-to-wafer bonding, electronic components are built on several semiconductor wafers and entire wafers are directly bonded together, where under 1.5um misalignment is achieved without significant bonding defects after process optimization [20]. Improvement of alignment accuracy also can be expected, because no deformable adhesive material is included in the bonding interface.

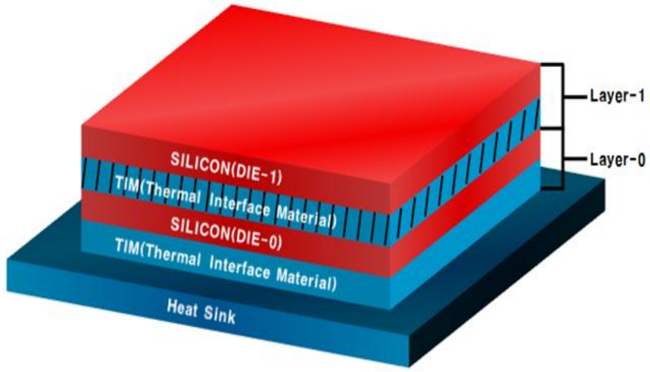

Layer-1
Layer-0
SILICON(DIE-1)
TIM(Thermal Interface Material)
SILICON(DIE-0)
TIM(Thermal Interface Material)
Heat Sink

Figure 1. 2-die 3D IC

B. Thermal Management techniques

Researchers have proposed a large number of DTM techniques to solve the thermal problems in the processor. DTM techniques can be categorized into two different groups: One is software-based techniques such as energy-aware task scheduling, OS-level task scheduling [21]. The other is hardware-based techniques, such as clock gating, dynamic frequency scaling (DFS), dynamic voltage scaling (DVS), instruction throttling, etc. Software-based techniques show lower performance degradation compared to the hardware-based techniques, but the cooling efficiency of software-based techniques is less effective than that of hardware-based techniques [22].

The existing DTM techniques for 2D multi-core processors have been used to prevent the case that the temperature of the chip exceeds the thermal limit supported by the cooling solution. However, high power density of 3D multi-core processors may require reactive DTM techniques to be engaged more frequently than the 2D multi-core processors, because the 3D architecture exacerbates the thermal problems in multi-core processors [23]. Unfortunately, DTM techniques incur performance overhead to control the temperature in the processor, resulting in performance degradation. Consequently, as DTM techniques are applied to the 3D multi-core processors, more performance degradation can be occurred compared to the 2D multi-core processors. Therefore, the thermal management for 3D multi-core processors should be used to proactively and continuously optimize performance and temperature, instead of merely reacting to emergencies [16].

C. Floorplan Technology

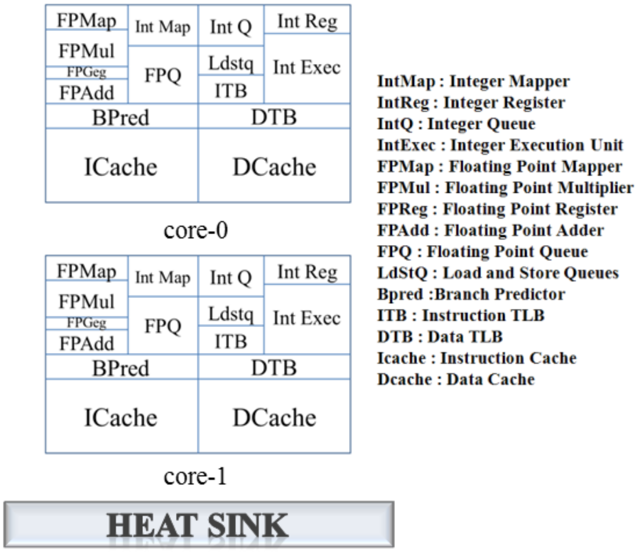
When designing up-to-date microprocessors, high performance, power efficiency, and thermal efficiency are all important design considerations. However, existing thermal-aware studies, i.e. DTM, reduce the peak temperature in the processor by sacrificing performance. For this reason, recent researches have focused on the solutions for reducing the peak temperature in the processor with little effects on the performance.

According to [17], the thermal-aware design techniques using floorplan scheme lead to peak temperature reduction with minimal performance degradation. Traditionally, floorplan schemes have been researched at the circuit-level [13]. However, as wire delay becomes the bottleneck of performance and thermal problems become critical issue, floorplan schemes have started to be looked at the architecture-level.

In floorplan schemes, heat transfer from adjacent functional units is one of the most important factors that affect the temperature distribution of a chip. The temperature distribution depends on the functional unit adjacency determined by the floorplan of the processor. In other words, the temperature on the functional units is heavily affected by the heat transfer from adjacent functional units.

III. PROPOSED THERAL-AWARE FLOORPLAN SCHEMES

In this paper, baseline floorplan model is based on the Alpha21364(EV6) [23]. We assume a 90nm technology with a supply voltage of 1.5 volts. To implement 2-die staked 3D IC, we extend it to 2 layers in our experiments.



IntMap : Integer Mapper
IntReg : Integer Register
IntQ : Integer Queue
IntExec : Integer Execution Unit
FPMap : Floating Point Mapper
FPMul : Floating Point Multiplier
FPReg : Floating Point Register
FPAdd : Floating Point Adder
FPQ : Floating Point Queue
LdStQ : Load and Store Queues
Bpred :Branch Predictor
ITB : Instruction TLB
DTB : Data TLB
Icache : Instruction Cache
Dcache : Data Cache

Figure 2. Baseline Floorplan of 3D Dual-core Processor - Baseline

Figure 2 denotes the baseline floorplan of 3D dual-core processor, which is the target processor in this work. Each core of the baseline processor is Alpha21364. We stacked two cores vertically to configure the dual-core processor. In Figure 2, core-0 represents the core located far from the heat sink while core-1 denotes the core located near to the heat sink. Proposed three floorplan schemes are shown in Figure 3 ~ Figure 5. The floorplan of core-0 is modified while the floorplan of core-1 is fixed, since the temperature of core-0 is higher than that of core-1 due to the different cooling efficiency. In this work, we implement a lot of experiments to find efficient floorplans for reducing the temperature. We describe only three efficient floorplan schemes owing to limited page.

In the proposed floorplan schemes, hottest functional units are relocated to reduce the temperature without increasing area. Compared to the Baseline in Figure 2, relocated functional units in the proposed floorplan schemes can be summarized as follows:

Floorplan I - IntReg, IntExec
Floorplan II - FPMul, LdStQ, IntReg, IntExec, IntQ
Floorplan III - FPAdd, LdStQ, IntQ, IntExec, IntReg

First floorplan scheme (Floorplan I) swaps the location of IntReg with that of IntExec because the IntReg unit is one of the hottest units in the baseline floorplan. As shown in Figure 3, Floorplan I doesn't change the location of other functional units. Second floorplan scheme (Floorplan II), shown in Figure 4, changes the location of FPMul, LdStQ, IntReg, IntExec and IntQ. As shown in Figure 5, third floorpaln scheme (Floorplan III) changes the location of FPAdd, LdStQ, IntQ, IntExec and IntReg.

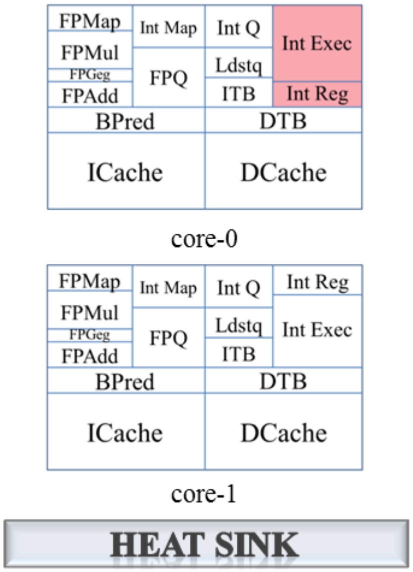


Figure 3. Floorplan I

Modifying the floorplan of the core affects the datapath in the processor, because the distance between functional units is determined by the floorplan. Data path is also considered in this work, because the floorplan in the proposed schemes is modified, significantly. We assume that the data path of Floorplan I is same to that of Baseline, since there is little difference between two floorplans. However, we changed the data path of Floorplan II and Floorplan III, because two floorplans have big difference compared to the Baseline. Especially, level 1 data cache latency related with LdStQ is increased compared to the Baseline, because the location of LdStQ is mainly changed in Floorplan II and Floorplan III.

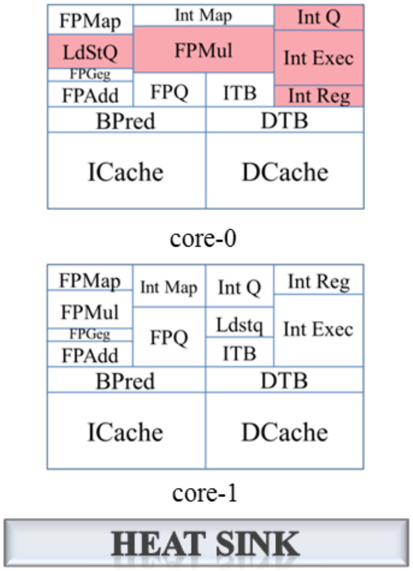


Figure 4. Floorplan II

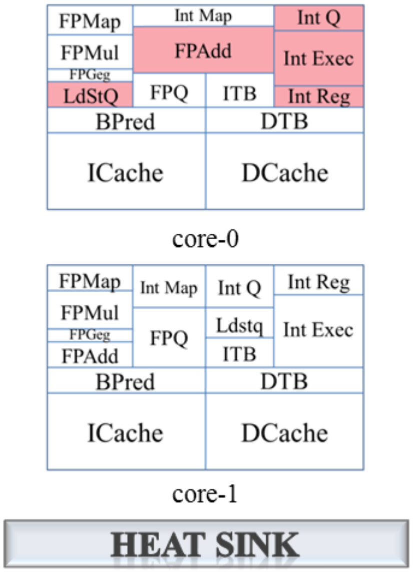


Figure 5. Floorplan III

IV. EXPERIMENTAL METHODOLOGY

In this section, we briefly describe the simulation infrastructure and thermal modeling. In order to determine the characteristics of the proposed schemes with respect to the baseline scheme, we perform applications selected from SPEC CPU2000 suite [24] using SimpleScalar [25] and Wattch [26]. SimpleScalar provides cycle-level modeling of processor in detail and Wattch is used to obtain the power trace of processor.

TABLE I. SYSTEM PARAMETERS

Parameter	Value
Functional units	**4 integer ALUs, 1 floating point ALUs 1 integer multiplier/divider 1 floating point 1 multiplier/divider**
L1 Instruction Cache	**32KB, 4-way, 32byte lines, 1 cycle latency**
L1 Data Cache	**32KB, 4-way, 32byte lines, 1 ~ 2 cycle latency**
Unified L2 Cache	**256KB, 8-way, 64byte lines, 12 cycle latency**

Table I shows the main processor and memory hierarchy parameters used in the simulation. In the up-to-date processors, DTM technique is applied to alleviate the thermal problems. Therefore, DTM techniques (initial temperature: 60℃, instruction throttling starting at 80℃, DVFS starting at 85℃) are also applied to the simulated processor.

TABLE II. BENCHMARK

Benchmark Program	*IPC*	*L2 Cache Miss Rates*	*Peak Temperature in 2D Processor (℃)*
gcc	2.48	0.03	78
mcf	2.73	0.25	81

Table II shows benchmark programs used in our simulation. Generally, benchmark programs are divided into two categories: memory-bound programs and cpu-bound programs. We use both memory-boud program(mcf) and cpu-bound program(gcc) to anaylze the temperature of the processor with various kinds of benchmark programs.

TABLE III. THERMAL MODEL PARAMETERS

Parameter	**Value**			
	TIM-0	***die-0***	***TIM-1***	***die-1***
Specific heat capacity(J/m^3K)	**4.0e6**	**1.75e6**	**4.0e6**	**1.75e6**
Thickness(m)	**2.0e-5**	**1.5e-4**	**2.0e-5**	**1.5e-4**
Resistivity(mK/W)	**0.25**	**0.01**	**0.25**	**0.01**

To evaluate the heat dissipation of 3D multi-core processors, especially inside each die, we use HotSpot Version 5.0 [27]. Hotspot is a modeling tool for developing compact thermal models. It can simulate the processor temperature in detail. And, HotSpot's grid model is capable of modeling stacked 3D chips. HotSpot takes power trace as input and generates the steady state temperature according to each functional block. Power trace is generated by using Wattch. In order to have precise experiments, the configuration parameters and the layer parameters are obtained from the material properties in CRC handbook [28]. Thermal modeling configurations of 3D dual-core processor used in our experiments are shown in Table III. In the table, die-0 represents the die located far away from the heat sink, while die-1 denotes the die located near to the heat sink.

V. EXPERIMENTAL RESULTS

In this work, we analyze the temperature and the performance according to the floorplan scheme for the 3D multi-core processor. In the results, notDTM and DTM represent the baseline floorplan without DTM technique and the baseline floorplan with DTM technique, respectively. Floorplan I, Floorplan II and Floorplan III represent the proposed floorplan schemes shown in Figure 3-5 with the DTM technique, respectively.

A. Temperature

We use the peak temperature of the processor instead of the average temperature because the major thermal problem is caused by the highest temperature. We assume that g and m represent the gcc and mcf applications obtained from SPEC CPU2000, respectively. In the graph, the first character denotes an application which is executed on the core-0 and second character denotes the application executed on the core-1. For example, the gm means that gcc is executed on the core-0 and mcf is executed on the core-1. The vertical axis represents the peak temperature of the processor.

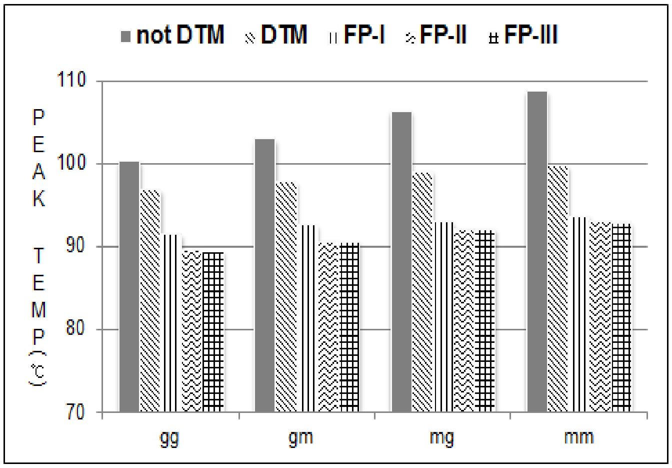


Figure 6. Peak temperature of 3D Dual-core processor

Figure 6 shows the peak temperature of the 3D dual-core processor according to each floorplan scheme. As shown in the graph, the temperature of the DTM is lower than that of notDTM. Our proposed three floorplan schemes reduce the peak temperature compared to the baseline floorplan.

In the three proposed floorplan schemes, Floorplan II and Floorplan III show lower temperature than Floorplan I. According to our simulation results, compared to notDTM, DTM decreases the temperature by 6.22℃ on average (gg:3.42℃, gm:5.16℃, mg:7.28℃, mm:9.0℃) and Floorplan I decreases the temperature by 11.9℃ on average (gg:8.8℃, gm:10.38℃, mg:13.4℃, mm:15.08℃). Floorplan II and Floorplan III reduce the temperature by 13.27℃ and 13.32℃ on average, respectively. Reduced temperature leads to higher reliability of the processor.

B. Performance

If the temperature on the processor exceeds the threshold, the DTM technique is activated, resulting in performance degradation. In the simulated 3D dual-core processor, the temperature of the core-1 is higher than that of the core-0. Therefore, the DTM is more frequently applied to core-1 than core-0. Consequently, the performance degradation of the core-0 is more serious than that of the core-1. Figure 7 shows total execution time of core-0 to analyze the performance according to each floorplan scheme. Each bar in the graphs is normalized to the execution time of notDTM.

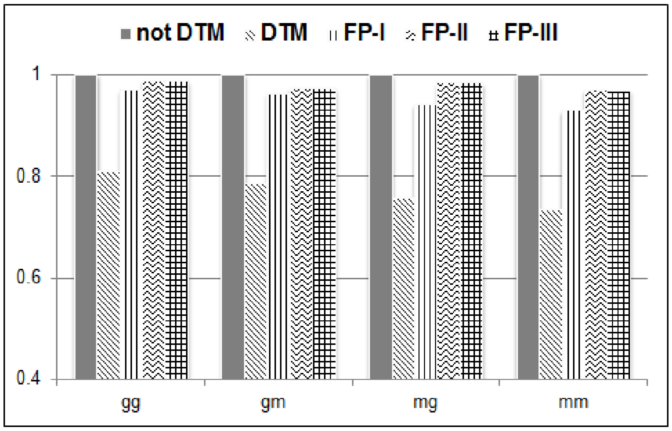


Figure 7. Performance of 3D Dual-core processor

We analyze the normalized performance according to each floorplan for 4 application combination such as gg, gm, mg and mm. For all cases, DTM shows the performance degradation by 22.89% compared to notDTM on average. In our simulation results, the first DTM technique, instruction throttling, is applied after 5.1% of the total execution time. Floorplan I, Floorplan II and Floorplan III degrade the performance by 5%, 2.19% and 2.2% on average compared to notDTM, respectively. Performance degradation in the proposed schemes is less than that of DTM, because the DTM technique is less frequently applied in the proposed schemes. Especially, even though Floorplan II and Floorplan III increase the latency for accessing the data cache(1 cycle → 2 cycle) due to modified floorplan, the performance is improved by about 3% compared to Floorplan I. According to our analysis, it comes from the fact that performance improvement obtained from reduced temperature is larger than the performance loss due to increased latency.

C. Energy consumption

As described in [29], we can see the equation for calculating the leakage power related to the area and temperature.

$$P_{leak} = \alpha \cdot Area \cdot e^{\beta(T_{current} - T_0)} \quad (1)$$

$$P_{total} = \sum P_{leak} \quad (2)$$

In the equation (1), the “α” and “β” represents a base leakage power and constant value which is changed by the technology, respectively. “$T_{current}$” is current temperature and “T_0” is temperature when the leakage power is base value. We can get the total leakage power by using the equation (2). As you can see in the equation, total leakage power decreases if “α” or “β” or “Area” or execution time becomes less. The proposed floorplan schemes reduce the execution time significantly compared to the traditional floorplan. Therefore, the proposed technique can reduce the leakage power consumption significantly.

VI. Conclusion

In this paper, we proposed thermal-aware floorplan schemes to solve the thermal problems in 3D multi-core processors. The proposed schemes reduce the peak temperature of the processor by adjusting the location of hot units to reduce the temperature increase due to heat transfer. According to our experiments, the proposed schemes reduce the temperature compared to the baseline scheme by 12.84℃ on average, leading to better reliability. Moreover, it reduces the performance loss due to the DTM technique significantly. The proposed floorplan schemes also show better power efficiency than the traditional floorplan scheme. Therefore, the proposed floorplan schemes can be a good solution for improving the performance, power-efficiency and reliability of 3D multi-core processors.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2012R1A1B4003492) and MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)(NIPA-2012-H0301-12-3005)

References

[1] V. Agarwal, M. S. Hrishikesh, S. W. Keckler and D. Burger, “Clock rate versus IPC: the end of the road for conventional microArchitectures,” Proc. the 27th International Symposium on Computer Architecture, Jun. 2000, pp. 10-14.

[2] L. Xiang, J. Huang and T. Chen, “Coordinating System Software for Power Savings,” Proc. Future Generation Communication and Networking, Dec. 2008, pp. 13-15.

[3] R. Palit, A. Singh and K. Naik, “An Architecture for Enhancing Capability and Energy Efficiency of Wireless Handheld Devices,” International Journal of Energy, Information and Communications. vol. 2, Nov. 2011, pp. 117-136.

[4] M. Chakraverty, S. Mandava and G. Mishra, “Performance Analysis of CMOS Single Ended Low Power Low Noise Amplifier,” International Journal of Control and Automation. vol. 3,Jun. 2010, pp. 45-52.

[5] S. Banerjee, M. Mukherjee and J. P. Banerjee, “Bias current optimization of Wurtzite-GaN DDR IMPATT Diode for High Power Operation at Thz Frequencies,” International Journal of Advanced Science and Technology. vol. 16, Mar. 2010, pp. 11-20.

[6] H. Naqvi, S. Berber and Z. Salcic, “Energy Efficiency of Collaborative Communication with imperfect Frequency Synchronization in Wireless Sensor Networks,” International Journal of Multimedia and Ubiquitous Engineering. vol. 5, Oct. 2010, pp. 19-30.

[7] J. W. Joyner, P. Zarkesh-Ha, J. A. Davis and J. D. Meindl, “A Three-Dimensional Stochastic Wire-Length Distribution for Variable Separation of Strata,” Proc. IEEE International Interconnect Technology Conference, Jun. 2000, pp. 7-9.

[8] L. Yeh and R. Chy, “Thermal Management of Microelectronic Equipment,” American Society of Mechanical Engineering, 2001.

[9] Z. Zhijun, L. R. Hoover, and A. L. Phillips, "Advanced thermal architecture for cooling of high power electronics," Components and Packaging Technologies, IEEE Transactions on, vol. 25, Dec. 2002, pp. 629-634.

[10] S. W. Yoon, D. W. Yang, J. H. Koo, M. Padmanathan and F. Carson, “3D TSV processes and its assembly/Packaging Technology,” Proc. IEEE International Conference on 3D System Integration, Sep. 2009, pp. 28-30.

[11] Zhu, C., Gu, Z., Shang, L., Dick, R.P., and Joseph, R. “Three-dimensional chip-multiprocessor run-time thermal management,” IEEE Transactions on Computer-Aided Design of Lntegrated Circuits and Systems, vol. 27, Aug. 2008, pp. 1479-1492.

[12] A. W. Topol, D. C. L. Tulipe, L. Shi, D. J. Frank, K. Bernstein, S. E. Steen, A. Kumar, G. U. Singco, A. M. Young, K. W. Guarini and M. Ieong, “Three-Dimensional integrated circuits,” IBM Journal of Research and Development, USA, 2006.

[13] J. Cong, G. J. Luo, J. Wei and Y. Zhang, “Thermal-Aware 3D IC Placement Via Transformation,” Proc. Asia and South Pacific-Design Automation Conference, Jan. 2007, pp .23-26.

[14] R. Mahajan, “Thermal Management of CPUs: A Perspective on Trends, Needs, and Opportunities,” Invited talk given at the 8th International Workshop on Thermal INvestigations of ICs and Systems, 2002.

[15] A. K. Coskun, J. L. Ayala, D. Atienza, T. S. Rosing and Y. Leblebici, “Dynamic Thermal Management in 3D Multicore Architectures,” Proc. Design, Automation & Test in Europe Conference & Exhibition, Apr. 2009, pp. 20-24.

[16] D. O. Son, Y. J. Park, J. W. Ahn, J. H. Park, J. M. Kim and C. H. Kim, “Thermal-aware Floorplan Schemes for Reliable 3D Multi-core Processors,” Proc. International Conference ICCSA, Jun. 2011, pp. 20-23.

[17] K. Sankaranarayanan, S. Velusamy, M. Stan and K. Skadron, “A Case for Thermal-Aware Floorplanning at the Microarchitectural level,” Journal of Instruction-level Parallelism, vol. 7, Aug. 2005, pp. 1-16.

[18] P. Lindner, V. Dragoi, T. Glinsner, C. Schaefer and R. Islam. “3D Interconnect through aligned wafer level bonding,” Proc. the Electronic Components and Technology Conference, May. 2002, pp. 1439-1443.

[19] P. Morrow, M. J. Kobrinsky, S. Ramanathan, C. M. Partk, M. Harmes, V. Ramachandrarao, H. M. Park, G. Kloster, S. List and S. Kim. “Wafer-level 3D Interconnects via Cu bonding,” Proc. the 2004 Advanced Metalization Conference, Oct. 2004, pp. 331-336.

[20] P. Leduca, F. de Crecy, B. Charlet, T. Enot, M. Zussy, B. Jones, J.-C. Barbe, N. Kernevez, N. Sillon, S. Maitrejean and D. Louisa. “Challenges for 3D IC Integration: bonding quality and Thermal management,” Proc. IEEE International Interconnect Technology Conference, Jun. 2007, pp. 21-212.

[21] X. Zhou, Y. Xu, Y. Du, Y. Zhang and J. Yang, “Thermal Management for 3D Processors via Task scheduling,” Proc. the 2008 37th International Conference on Parallel Processing, Sep. 2008, pp. 9-12.

[22] T. Pering and R. Brodersen, “Energy efficient voltage scheduling for real-time operating systems,” Proc. the 4th IEEE Real-Time Technology and Applications Symposium RTAS'98, Work in Progress Session, Jun. 1998, pp. 3-5.

[23] R. E. Kessler. “The Alpha 21364 microprocessor,” IEEE MICRO, vol. 19, 1996

[24] J. L. Henning, “SPEC CPU2000: Measuring CPU Performance in the New Millennium,” IEEE Computer, vol. 33, Jul. 2000, pp. 28-35.

[25] D. C. Burger and T. M. Austin. “The SimpleScalar tool set, version 2.0,” ACM Special Interest Group on Computer Architecrue Computer Architecture News, vol. 25, Jun. 1997, pp. 13-25.

[26] D. Brooks, V. Tiwari and M. Martonosi, “Wattch: A Framework for Architectural-level Power Analysis and Optimizations,” Proc. the 27th Annual International Symposium on Computer Architecture, Jun. 2000, pp. 10-14.

[27] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan and S. Ghosh. “HotSpot: A Compact Thermal Modeling Method for CMOS VLSI Systems,” IEEE Transactions on VLSI Systems, vol. 14, May. 2006, pp. 501-513.

[28] CRC Press, CRC Handbook of Chemistry. http://www.hbcpnetbase.com

[29] A. K. Coskun, A. B. Kahng and T. S. Rosing, “Temperature- and Cost-Aware Design of 3D Multiprocessor Architectures,” Proc. Architectures, Methods and Tools, Aug. 2009, pp. 27-29.

A Plug-in Node Architecture for Dynamic Traffic Control

Wangbong Lee, Dong Won Kang, Joon Kyung Lee
Division of Next Generation Internet Research
Electronics and Telecommunications Research Institute
Daejeon, Korea
emails {leewb, dwkang, leejk}@etri.re.kr

***Abstract*—Accurate traffic classification is fundamental to many network activities such as an analysis of the application usage, anomaly detection, and accounting. However, the architecture of most network devices is strictly fixed. We need a dynamic architecture to deal with increasing traffic. We therefore propose a plug-in node for dynamic traffic control. A plug-in node is designed to load and unload traffic processing plug-in modules on the fly. Plug-ins can be implemented with various functions including packet classification, anomaly detection, and application monitoring.**

Keywords-Plug-in; Traffic Classification; Performance model.

I. INTRODUCTION

A greater understanding of Internet traffic has become more important over the past few years [1]. Moreover, mobile data traffic is expanding more rapidly now than ever before. The number of mobile subscribers and applications continues to increase globally [2]. To deal with the current expansion of data traffic, network/mobile operators need to optimize their networks and collect key performance data through traffic measurement and monitoring techniques.

Research on Internet traffic classification has generated creative and novel approaches. Accurate traffic classification is fundamental to many network activities [6] such as an analysis of application usage, anomaly detection, and accounting. A correct analysis of Internet applications can provide valuable information to network operators for building efficient networks. Detecting anomalies in a network is a critical activity for network operators and end users in terms of service availability. Traffic classification is also important for application-based billing and user-based services.

A work by Arthur Callado et al. introduced and classified traffic classification techniques [1]. Summarizing this research, there is no silver bullet technique in terms of accuracy and completeness. A work by Alberto Dainotti et al. discusses recent achievements and future directions in traffic classification [3]. This research suggests several strategies for tackling unsolved challenges. Their recommended strategies focus on a mixture of traffic classification techniques, speed increments of network links, rigorous evaluations, and open source implementations.

In 2011, KaKaoTalk, one of most famous social network service (SNS) application in Korea, overloaded the 3G mobile network through its highly frequent control messages [4]. One major Korean network operator, SKT, recognized the importance of network traffic optimization, and they proposed a network traffic mitigation method [5]. However, newly proposed techniques are not easy to apply to current network devices such as access points, edge routers, and gateways

For this research, we designed a traffic classification system supporting high-speed links based on a commercial network processor and its architecture. The system has a dynamic plug-in node for supporting various traffic control functions.

Internet traffic is always changing. The variety and complexity of modern Internet traffic exceeds anything imagined by the original designers of the fundamental Internet architecture. As the Internet has become our most critical communications infrastructure, service providers are attempting to improve the functionality, including security, reliability, privacy, and multiple service qualities, into a best-effort architecture. To prioritize, protect, or prevent certain traffic, service providers need to implement a technology for traffic classification. In this paper, a plug-in system is proposed for adopting various traffic classification techniques. This system is designed for a user-defined network service using a network processor based network interface card (NIC). Our proposed conceptual model of a plug-in node is shown in Figure 1.

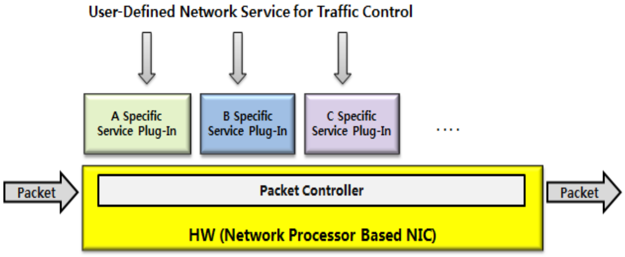

User-Defined Network Service for Traffic Control
A Specific Service Plug-In
B Specific Service Plug-In
C Specific Service Plug-In
....
Packet
Packet Controller
Packet
HW (Network Processor Based NIC)

Figure 1. Conceptual model for a plug-in node

The packet controller is similar to a network switch. It must perform packet classification at high speeds to efficiently implement various functions. Packet classification requires matching each packet against predefined rules, and forwarding the packet according to the matching results. When a packet is matched against predefined rules, it is forwarded to a specific plug-in module that is dynamically loaded or unloaded by users. Service plug-in modules include a Peer to Peer (P2P) traffic control plug-in, traffic classification plug-in, traffic monitoring plug-in, and so on. Service plug-ins can be implemented as traffic classification techniques. Thus, the plug-in node is considered a multi-traffic classifier system. A newly proposed method of traffic identification can apply to proposed node owing to its plug-in feature.

The rest of this paper is organized as follows. We begin by reviewing proposed plug-in node architecture in Section II. In the next section, we provide the test result in its prototyping system. Finally, some concluding remarks and a description of future work are given in Section IV.

II. PLUG-IN NODE ARCHITECURE

To deal with the rapid growth of Internet traffic, current backbone network devices such as routers and switches have to manipulate millions of packets per second at each port. To achieve such a high packet processing rate, hardware devices such as an application-specific integrated circuit (ASIC), field-programmable gate array (FPGA), and ternary content-addressable memory (TCAM) are usually adopted in the design of current network devices. Current Internet traffic processing systems need to support a variety of emerging network applications while guaranteeing a high packet processing rate. To achieve the requirement of a high packet processing rate, network processors are introduced as a promising solution for building network devices [7]. Our proposed node is implemented on a Tilera network processor board in a general x86 Linux server. The Tilera network processor board is designed as an NIC, and has a peripheral component interconnect express (PCI-e) interface and 4 x 10 Gb Ethernet ports, as shown in Figure 2 [8].

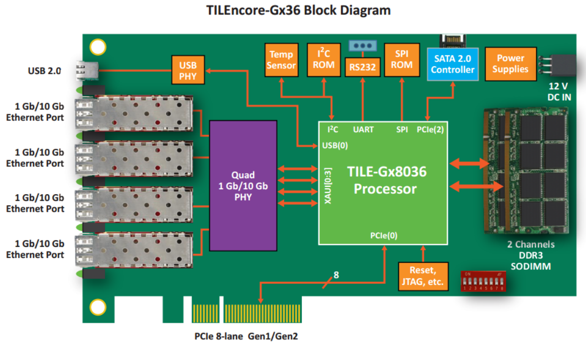


Figure 2. Tilera network interface card

We designed the main components, which manipulate packets and hardware resources, into a network interface card. The service plug-in adapter manages the plug-ins and isolates their resource space, which prevents abnormal crashes from creating a failure in another plug-in. Figure 3 shows the main components: a command line interface (CLI) parses the user's commands and transmits the control messages to the appropriate modules. A virtual switch implements the general switch functions including queue management and packet forwarding among the interfaces ports. It manages the packet flows through a user-defined rule, which is based on the 6-tuple classification rule. The plug-in manager starts and stops the plug-in modules using their service profilers. The resource monitor reports the hardware resource information such as memory usage and processor usage. The connector is a communication channel between the service plug-in adapter (SPIA) and service plug-in panel (SPIP), which is implemented in x86 Linux server, as shown in Figure 4. This channel uses a PCI-e interface and external management Ethernet interface.

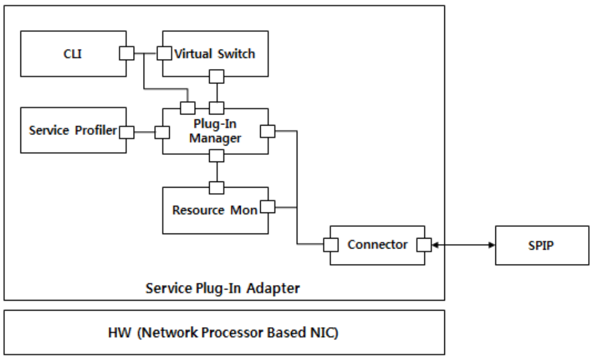


Figure 3. Service plug-in adapter

A service plug-in panel consists of a platform manager, service manager, and log manager. The service manager controls the plug-ins using their service profile information. Users can build plug-in programs and their own profiles using the software developer's kit (SDK) [8]. Plug-in SDK uses a Tilera network processor's compiler and simulator, as well as plug-in libraries. The log manager reports and saves errors and resource information from the service plug-in adapter. The connector has the same functionality as the service plug-in adapter.

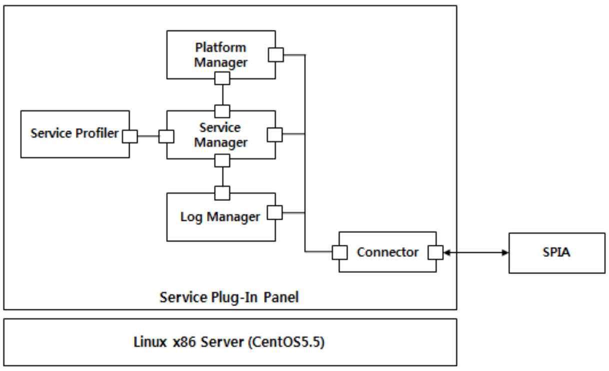


Figure 4. Service plug-in panel

III. EVALUATION

We implement the plug-in node to demonstrate its feasibility. This proof of concept is implemented on a Tilera multi-core processor. We implement the core functions of the plug-in node including the switch function, plug-in

ISBN: 978-1-61208-239-4

manager, and resource manager. Figure 5 shows the test topology. Various types of traffic are generated by tcp replay running on a regular Linux operating system [9][10]. The results of the experiments are shown in figure 6. The average latency of a 1,024-byte packet for five plug-ins is 9.7 msec, while that of a 1,024-byte packet for no plug-in is 2.4 msec.

Chertov et al. [11] estimated that a perfect router has under a 0.2 msec packet delay for a 1,024-byte packet, while a Cisco router has under a 0.35 msec packet delay. A perfect router is hypothetical, as it has zero processing and queueing times. Our current experiment values are not adequate to apply the commercial product. The current prototype uses Linux network APIs for a proof of concept. Thus, we have room to optimize the average latency during the development phase

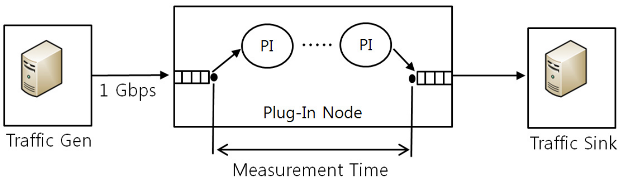
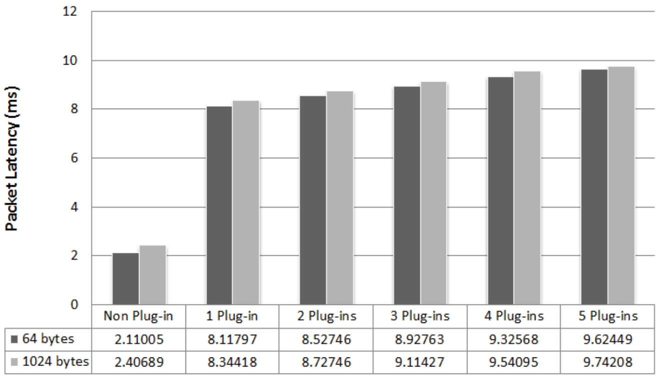


Figure 5. Test Topology



	Non Plug-in	1 Plug-in	2 Plug-ins	3 Plug-ins	4 Plug-ins	5 Plug-ins
■ 64 bytes	2.11005	8.11797	8.52746	8.92763	9.32568	9.62449
■ 1024 bytes	2.40689	8.34418	8.72746	9.11427	9.54095	9.74208

Figure 6. Packet Latency in Plug-in node

IV. CONCULSION AND FUTURE WORK

The goal of this study was to provide a plug-in node architecture. It is designed to load and unload traffic processing plug-in modules during runtime. Plug-ins can be implemented with various functions including packet classification, anomaly detection, and application monitoring. In this paper, we implemented the prototype of plug-in node and provided the test result. Our current experiment values are not adequate to apply the commercial traffic control product. We are going to optimize the code using network processor specific APIs during the development phase.

We plan to evaluate the performance of our proposed node under various realistic application workloads using Tmix [10]. Also, we are going to study on a packet latency analysis model based on a queuing system to evaluate our proposed node.

For our future work, we consider the software defined networking (SDN) in terms of dynamic traffic control. Thus, we are going to research on how to interact SDN control plane such as NOX, and how to implement SDN packet forwarder as a plug-in.

REFERENCES

[1] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, and D. Sadok, “A Survey on Internet Traffic Identification,” Communications Surveys & Tutorials, IEEE , vol. 11, no. 3, pp. 37-52, 3rd Quarter 2009

[2] http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html [retrieved: July, 2012]

[3] A. Dainotti, A. Pescape, and K. C. Claffy, “Issues and future directions in traffic classification,” *Network, IEEE* , vol. 26, no. 1, pp. 35-40, January-February 2012

[4] http://www.seoul.co.kr/news/newsView.php?id=20110402015019 [retrieved: July, 2012]

[5] M. W. Kim, D. G. Yun, J. M. Lee, Y. J. Shim, and S. G. Choi, “Network traffic mitigation method using TCP signalling delay algorithm,” Advanced Communication Technology (ICACT), 2012 14th International Conference on , vol., no., pp.730-733, 19-22 Feb. 2012

[6] A. W. Moore and D. Zuev, “Internet traffic classification using bayesian analysis techniques,” Proceedings of the 2005 ACM SIGMETRICS, pp. 50-60, June 06-10, 2005, Banff, Alberta, Canada

[7] http://en.wikipedia.org/wiki/Network_processor [retrieved: July, 2012]

[8] http://www.tilera.com/ [retrieved: July, 2012]

[9] http://tcpreplay.synfin.net/ [retrieved: July, 2012]

[10] M. C. Weigle, P. Adurthi, F. Hernandez-Campos, K. Jeffay, and F. D. Smith. “Tmix: A tool for generating realistic application workloads inns-2,” ACM Computer Communication Review, vol. 36, pp. 65–76, July 2006.

[11] R. Chertov, S. Fahmy, and N. B. Shroff, “A Device-Independent Router Model,” *INFOCOM 2008. The 27th Conference on Computer Communications, IEEE* , pp. 1642-1650, 13-18 April 2008

Emerging Function Concept Applied to Photonic Packet Switching Network

Antonio de Campos Sachs, Ricardo Luis de Azevedo da Rocha, Fernando Frota Redigolo, Tereza Cristina Melo de Brito Carvalho.

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)
Escola Politécnica da Universidade de São Paulo (EPUSP)
São Paulo, Brazil
antoniosachs@larc.usp.br; luis.rocha@poli.usp.br; fernando@larc.usp.br; carvalho@larc.usp.br

***Abstract*—Aiming to contribute to a solution for the number of nodes scalability problem, a transparent Optical Packet Switching (OPS) network is treated in a new approach that considers the network as a complex system. This treatment allowed the investigation of a network based on large number of autonomous OPS nodes connected in a mesh topology. The network operates in a bottom-up organization and the long distance signalization, one of the main factors responsible for the number of nodes limitation, is avoided and the scalability is enabled. Desirable characteristics (scalability, traffic organization, protection, restoration, etc.) result from simple rules executed by individual nodes. All those characteristics are referred to as Emerging Functions. It is possible to create those simple rules, or fundamental individual functions executed by individual nodes, in order to potentiate those Emerging Functions. As an example, a set of node interactions with their next neighbors is described. That will result in a Protection Emerging Function able to maintain the network operation after failure and to confine the network degradation just around the failure position. That segregation of the failure effects represents a new feature that could be observed due to the new approach.**

Keywords-complexity; emerging function; photonic packet switching; protection

I. INTRODUCTION

One important limit factor for the scalability of the number of nodes in a network is the long time necessary for communication between two distant nodes. The current approach, which treats the OPS network as a complex system and the network nodes as autonomous entities, was previously described in [1]. That approach avoids long distance signalization. A packet is sent from source to destination without any previous path determination. Routing activities needs to use a shortest path table previously calculated at the moment of the network initialization. From that shortest path table each node knows the number of the output port that corresponds to the shortest path connecting itself to any other node. Each packet carrying the destination address can find the path from source to destination from node to node in a multi hop schema using the output port corresponding to the shortest path or the alternative port in those cases in which the preferred one is not available.

Simple switching device without optical buffers that forwards the arriving packet without delay to the preferential output port or to the alternative one is a procedure referred to as "hot potato routing" [2]. That operation can be performed by using an optical sample removed before a FDL (fiber delay lane). This sample can be converted into electrical media for the logical treatment performed by conventional electronic circuitry and the optical switch, based on SOA (Semiconductor Optical Amplifiers) devices, can be positioned before the arrival of the packet that is traveling through the FDL. Such operations have been adopted since the precursor projects KEOPS [3] and DAVID [4]; but, nowadays, the conversion from optical to electrical media is not necessary and new photonic devices can do all the jobs (including logical operations), and the switching operation can be performed in a fully optical process [5]. The network described herein works for any technology utilized for reading the address and forwarding the packet to the output port. Apart from the technology used inside the node, the network can operate as a complex system, with a bottom-up organization. Each node has the autonomy to carry on the switching operation, performing its work exclusively with locally obtained information.

The utilization of large number of nodes with large number of alternative paths, provided by the mesh topology, is known to be important for the network survivability. Since the beginning of the digital telecommunication technology, Baran [2] worked with mesh topology and got very strong robustness for a network with a large number of nodes. Today, the survivability of a complex network is associated to the intrinsic robustness of complex systems [6, 7]. Carlson and Doyle [6] claims that all complex systems are intrinsically robust for the most frequent daily events; however they are very fragile due to rare and unexpected environment events. Reference [7] declares that “hubs make the network robust against accidental failures but vulnerable to coordinated attacks”. All agree that complexity is intrinsically related to robustness.

Next, Section II describes generically aspects related to Emerging Functions applied to a network and Section III describes the network operation. Section IV describes the adopted theory, calculations aspects and discussions analyzing the failure distribution effect for a 256-node case study. Section V presents the final conclusions.

II. EMERGING FUNCTIONS APPLIED TO A NETWORK

The term Emerging Function is utilized in a number of different areas, such as physics, chemistry or biology. Although there is no single formal definition for the term, two main definitions can be inferred:

- A function that is not regularly present in a system and appears or is activated automatically in an emergency situation;
- A function that is always present in a system (it characterizes the system) and emerges from simple operations executed by its individual parts.

An emerging function is associated to the whole system and not to its individual parts although its emergence is the result of small changes in the normal operations (first definition) or of regular operations of individual parts of the system (second definition).

A system based on emerging functions can be characterized as a bottom-up organization system or, equivalently, a self-organized system [8] and it is associated with a complex system composed by a large number of individual units following simple operation rules. It is difficult to deal with such complex systems, with a large number of elements, in a classical and reversible treatment that calculates all the possible events in all the system components. The models considering the probability of transition from one state to the next one to describe the system evolution seem to be a more feasible strategy. That is also the same strategy found in the chaos theory [9], in which the final consequences cannot be derived from the initial conditions because there is a high sensitivity to tiny fluctuations in those initial conditions.

The network routing function herein is not controlled by the network layer (OSI network layer 3). It emerges from simple fundamental functions executed by each node individually. There is no high entity accounting for switches operation or for the path followed by each packet in the network. Instead, the node operation is based on the local situation and on the packet header information: each packet is sent to the preferred output port, or sent to the available port if the preferred one is occupied. This operation rule, by itself, turns the network auto-organized or bottom-up organized, and provides autonomic network operation. Therefore, it is possible to consider “routing” as a function emerging from individual nodes operations or, in other words, that routing is an Emerging Function.

Traffic distribution, which can be considered the set of all routes, is also an Emerging Function. As the shortest path is not always the one which is chosen, the traffic distribution obtained is better than the one obtained utilizing only the shortest path.

The access to the network is made only if there is a time interval to accept the new packet without collision. This is possible because of a fiber delay line (FDL) positioned before any input port. Collision avoidance can also be interpreted as an Emerging Function, since it is not executed by any higher protocol layer, but it results from the careful local insertion procedure.

Protection is an important network function that can be enabled through the insertion of an extra individual node operation function based on a backward signalization sent to all the input ports. The output ports integrity can be checked through the signalization received from the next node. Protection can also be considered as an Emerging Function and its architecture is presented in detail in Section III.

III. Network Operation

The network architecture is based on the “Hot Potato Heuristic Routing Doctrine” [2] made up by network nodes executing simple well-defined rules. A set of Emerging Functions arise from those simple rules. The network complexity is related to its size and the number of nodes. Each node, in contrast, is idealized to be simple. The first simplification is the omission of optical buffers. Without optical buffers, it is necessary to use symmetrical nodes in order to avoid packet losses. In symmetrical nodes, with the same number of input and output ports, there is always a free output port for any arriving packet.

Manhattan-Street Network (MSN) [10] was chosen as the main topology for the development of this work, but any other mesh topology can be considered. This particular choice facilitates the calculations for increasing the number of nodes without changing the network symmetry.

To implement the protection emerging function, it is necessary to differentiate the two output ports in order to define links sub-domains as described in [1]. Figure 1 is a MSN showing clockwise and counterclockwise sub-domains. Each node belongs to two sub-domains and each sub-domain contains four nodes.

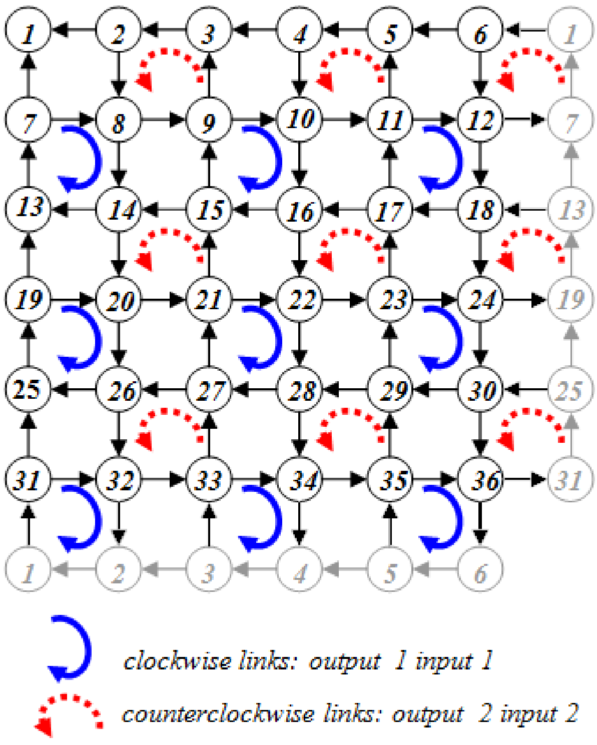

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
clockwise links: output 1 input 1
counterclockwise links: output 2 input 2

Figure 1: MSN organized with clockwise and counterclockwise link sub-domains [1].

After organizing the network links in small sub-domains, it is possible to create the protection function by including an operation rule for all network nodes. This operation rule is composed by a continuous optical signal which is sent backward through the two input ports and the operation of reading the arriving signal from the two output ports. This signal is named integrity signal, as the integrity of a link sub-domain is signalized by the optical signal continuously traveling in the opposite direction of the optical packet signal. When a failure occurs in one link, the backward integrity signal is interrupted and the node, which is just before that link, is immediately aware of the failure and no longer uses that output port. The action rule for all nodes is to turn off the integrity signal forwarded to the input port belonging to the failed sub-domain.

In this work, a second signalization was added, sent backward to inform the nodes outside the failed sub-domain that the link is working properly but the next node belongs to a failed sub-domain.

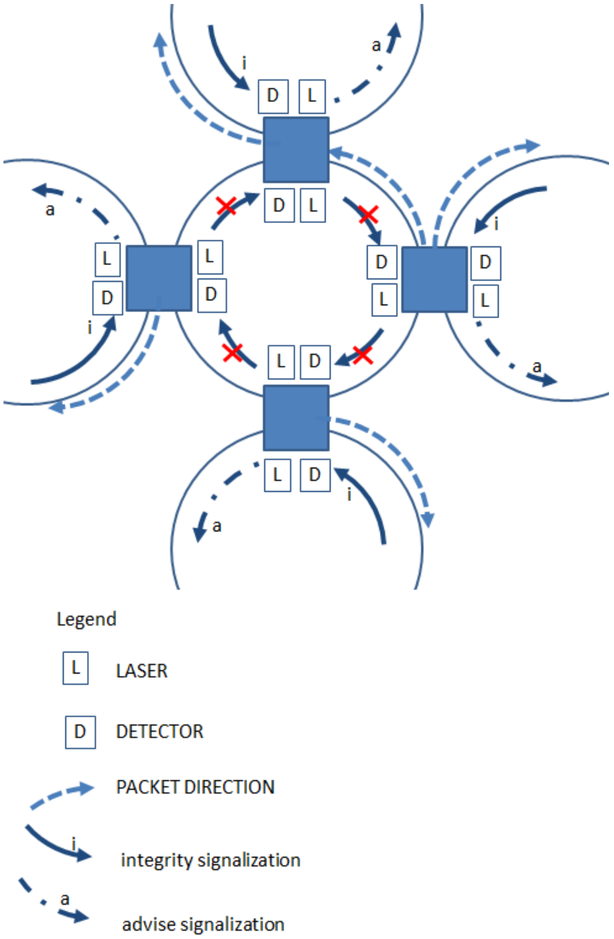


Figure 2: First signalization (link integrity DC laser signal) and second signalization (square wave advise signalization).

Figure 2 shows four nodes with four links in a counterclockwise sub-domain. Each node has a DC laser sending a continuous laser signal in clockwise direction (opposite direction of the counterclockwise packets) and has a detector placed to receive the laser signal sent by the next neighbor belonging to the same counterclockwise sub-domain. In case any one of those four links is interrupted, the detector that first stops receiving the signal turns off the laser corresponding to the same sub-domain. All the four links in the ring are forced to be interrupted. The four lasers are turned off. Each node belongs to two sub-domains and uses a second laser as well as a second detector for integrity signalization of that second sub-domain. In a normal process, without failure, all signalization is of the first type (continuous laser signal), but in the case of failure, the signalization is interrupted in the failed sub-domain and changed from continuous signal to square wave signal in the second sub-domain (four next neighbor sub-domains). The failure causes four links to stop the first signalization (integrity signalization) and to start the second signalization (advise signalization) outside the failed sub-domain.

The implementation of that second level of information can be performed by a square wave light signal replacing the DC light signalization or, alternatively, a DC laser can be used with half optical power to differentiate from the full optical power of the regular link integrity signalization. The implementation of both signalization types, indeed, can also be made by smart photonic devices in a fully optical process.

All the nodes at a failed sub domain operate with only one input and one output. All the other nodes, far from the failure, have no information about the failure and their procedure remains the same, including the utilization of the same preferential output port matrix. The action of the node receiving the second signalization is to deflect all packets to the other output port (the port that is receiving the first signalization type), with the exception of the packets addressed to those nodes that are sending the second signalization type. That is the only way a failed sub-domain node can receive a packet. That deflection corresponds to an adaption in the preferential port table.

As an example, consider a failure in the counterclockwise sub-domain connecting nodes 17, 16, 22, 23 in Figure 1. Those four nodes extend the information failure to the remaining input by changing the backward continuous wave light source to square wave light-source. That signalization is a sign for the preferential port adaption in nodes 18, 10, 21 and 29.

That new feature was implemented in the calculations, and the results are shown in Figure 3 for 16 nodes (N=16) and 256 nodes (N=256). The caption termination “F” refers to the first level protection schema, characterized by not using the second signalization type. The caption termination “F2” refers to the second level protection schema that includes a second signalization type. For 16 nodes, the second level protection schema (N=16F2) shows that the interference of the failure is remarkably smaller than that observed at the first level protection schema (N=16F).

One additional feature is the correction of a strange behavior for low charge condition. In that region (Link Load < 50%) the failure causes a large number of hops enhancement and it is quite odd to see the number of hops decreasing for higher load condition (curve N=16F in Figure 3). That behavior can be explained by the fact that in the low load condition, the packets take the preferential output port more often as compared to the large charge condition and are forced to proceed through the failed region. The failure is more efficiently avoided with the second signalization, minimizing such effect. All the unnecessary trial through the failed region is avoided.

IV. Calculations

To deal with scalability, the number of nodes can be higher than practical calculations can support. It is impossible to implement calculations for an arbitrarily large number of nodes. In order to minimize the time and memory utilization, connection matrix “c” and preferential output port matrix “pp”, were calculated separately. Data were saved in files that could be interpreted by the main program. The shortest path calculation is presented in sub-section A. The algorithm description for the mean number of hops calculation is presented in sub-section B. The model validation carried out by comparison with the simulation model is presented in sub-section C. One important result, the segregation of the failure effect, is presented in sub-section D.

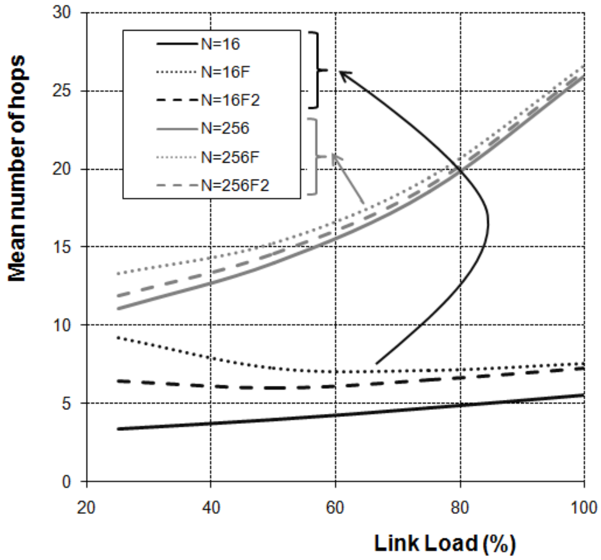


Figure 3: Number of hops increases after failure for two types of signalizations.

A. Shortest Path Calculation

The shortest path to reach the destination is calculated once for a non-failed topology. As the packet can be deflected to any output port, it must be able to find out the destination shortest path from any place in the network and not only from the origin. The packet is informed about the shortest path through a preferential port matrix “pp” with dimensions *NxN*, were *N* is the total number of nodes. Each column of the pp matrix represents the actual position of a packet and the matrix elements are numbers indicating the best option: number 1 for output 1 or number 2 for output 2 or number 3 to indicate that there is a shortest path starting from both outputs.

Before the calculation of the preferential port matrix pp, it is necessary to know the connection matrix “c”. That matrix is a sparse *NxN* matrix representing the considered topology. Each column in the c matrix represents the actual position of a packet and each line represents the destination to be reached in one hop. With the exception of the two directly connected nodes, all the *N-2* elements in any column of the c matrix are equal to zero. In addition to the information of the directly connected nodes, the non-zero elements also inform the link sub-domain type number. The positions of the matrix elements correspond to the directly connected nodes and the matrix element itself represents the sub-domain type 1 (output 1) or sub-domain type 2 (output 2). Type 1 could be, as an example, the clockwise type and type 2 the anticlockwise type (see Figure 1).

Starting from connection matrix c, the preferential port matrix pp is constructed. This is done column by column, in an adaptive tree procedure [11] that is nondeterministic just at the beginning of the algorithm. That procedure is nondeterministic because it is necessary to calculate the smallest path starting from all possible packet positions.

B. Mean number of hops calculation

The network performance is measured by the mean number of hops *<H>* a packet completes from origin to destination. The main program, utilized for the main number of hops calculation, is based on the evolution of a vector *P(x)* with *N* dimensions. The *x* variable is the discrete position for the packet *(x = 1, 2, …, N)*. Each vector represents the probability of finding a hypothetical packet in etch node. That is called probability distribution vector. The mathematical treatment for the evolution of a probability distribution along time corresponds to the application of an operator *“U”* to the probability vector $P_t(x)$ at any instant of time *“*t*”* to obtain the probability vector $P_{t+1}(x)$ at the instant of time “t+1” after a discrete time interval. The unitary increment of time corresponds to one hop from one node to the following in the packet traveling from source to destination.

$$P_{t+1}(x)=UP_t(x) \qquad (1)$$

Operator *U* is analogous to the “Perron-Frobenius operator” utilized in the chaos theory for the calculation of the time evolution of a probability distribution [9]. An analogy can be constructed with the chaos theory, in which the idea of trajectory is abandoned and replaced by the evolution of a probability distribution. In this work, the idea of a path that a packet should follow from its origin to its destination is replaced by the probability distribution vector time evolution described in [1]. Acampora and Shah [12] consider similar statistical procedure to describe the behavior of a store-and-forward routing as a comparison with hot-potato routing. Due to the fact that the probability to go directly from one node to the other is zero for almost all nodes except for the two directly connected nodes, most of the elements in operator *U* are zero. Each column has only two non-zero elements. The preferential output port has probability *Ppp* and the alternative port, corresponding to the deflection port, has probability *Pd* given by:

$$Pd = 1 - Ppp \tag{2}$$

A packet is sent to the preferential port in tree cases:

a) There is no other packet in the competitor link that could arrive before it.

b) There is another packet that could arrive before it, but that has a local final address and is going to be removed before competition.

c) There is another packet arriving before it that is not a local packet, but it has a different preferential output port.

The link occupation probability *Poc* defines the probability of the first case to be $1 - Poc$. Given that case a) is not true, the local packet probability *Plp* defines the second case probability term as Poc_*Plp. Finally, given that case a) and case b) do not apply, considering *Pop* as the probability of the competitor packet to have a different preferential port (another port), the third term is defined as $Poc_*(1\text{-}Plp)_*Pop$. The final probability of a packet to go through the preferential port *Ppp* is given by:

$$Ppp = 1 - Poc + Poc_*Plp + Poc_*(1 - Plp)_*Pop \tag{3}$$

In (3), *Poc* is the occupation probability that is associated to the link load. It is considered that a fully loaded link (not considering the FDL length) corresponds to $Poc=1$. The probability of a packet preference pointing to another port *Pop* is assumed to be 50% and $Pop=0.5$ in all cases. The local packet probability *Plp* is evaluated to be $1/\langle H\rangle$, with $\langle H\rangle$ calculated as a preliminary mean number of hops obtained with a first guess value $Plp=1/(N-1)$.

The $Plp=1/\langle H\rangle$ hypothesis is originated by the fact that all packets, at any time, belong to his path from origin to destination. The mean number of hops in all possible paths is $\langle H\rangle$ and the packet is considered to be a local packet only in the last of those hops. That means that *Plp* is the probability of a packet to be positioned at the last hop of its path from origin to destination.

Without failure, the Manhattan Street network architecture belongs to a symmetry group called automorphism [10]. In this group, it is impossible to differentiate any node from the other concerning its position in the network. The mean number of hops is the same regardless the position of the final address node. But introducing a failure, the symmetry is broken and the mean number of hops may assume different values for different final destinations. In this case, it is necessary to calculate the mean number of hops for all possible destinations and to adopt the arithmetic mean of those values as the final network mean number of hops.

One more consideration should be made about the “don’t care” nodes. They are already identified and signalized by number 3 in the preferential port matrix pp. In that case, it is considered that the packet plays no role in the decision of the preferable output port. The position of the switch may be adjusted to the preferred output port of the packet eventually arriving in the competitor link. That procedure corresponds to considering $Ppp=Pd=50\%$ in all “don’t care” situations.

The number of hops is obtained recursively by (1) starting with $P_1(x)$, that represents the probability to reach the destination with one hop, to calculate $P_2(x)$, that represents the probability to reach the destination with two hops. That procedure is repeated *k* times while the total probability is less than 100%, with an arbitrary criteria chosen to be $\Delta P=10^{-6}$. Further reduction of that criterion interferes only with the calculation time and no change is observed in the results for $\Delta P=10^{-5}$.

As an example, for a network with four nodes ($N=4$), the initial probability to find a packet addressed to node number one, in any place is considered to be zero to node number one and 1/(N-1) for all the other nodes. That condition is represented by the initial probability vector $P_1(x)$ given in (4).

$$P_1(x) = \begin{bmatrix} 0 \\ 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \tag{4}$$

The mean number of hops for each destination *x* is calculated by the equation:

$$< H >= \sum_{1}^{k} tP_t(x) \tag{5}$$

With the condition:

$$1 - \Delta P < \sum_{1}^{k} P_t(x) \le 1 \tag{6}$$

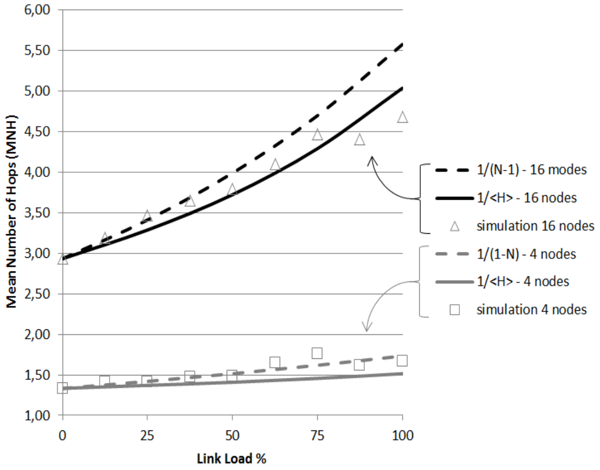


Figure 4: Analytical and simulation models.

C. Simulation model

The time domain simulation model (TDSM) was developed over the OMNET++ platform. The simulation model considers all the nodes sending packets to all the others and following the same rules used for the analytical model. Every packet arriving to one *2x2* node is addressed to the better output port, unless the node is already occupied with a competitor packet. In that case, the packet is sent to the available output port. The destination and the exact instant of packet generation are randomly chosen. Each link load condition is governed by the packet size. A packet with half the link size is used to simulate the 50% link load condition. Each packet that reaches the destination stimulates the insertion of a new one, addressed to a new randomly chosen destination. That procedure insures the maintenance of the link load condition all along the simulation time. A 40Gbps bit rate and one kilometer link length were considered. The delay line fiber length is considered to be equal to the link length, the same hypothesis utilized in the analytical model. Figure 3 shows the simulation results compared to the analytical model results for two hypotheses utilized for the evaluation of the local packet probability *Plp*. The agreement between models is better for hypothesis *Plp=1/<H>* as compared to the hypothesis of the first guess *Plp=1/(N-1)*. In fact, that first guess is very close to simulation results for the small number of nodes but tends to decrease faster than *Plp=1/<H>* producing wrong results for a higher number of nodes. The simulation time is far higher than the analytical calculation time, limiting its utilization for scalability issues.

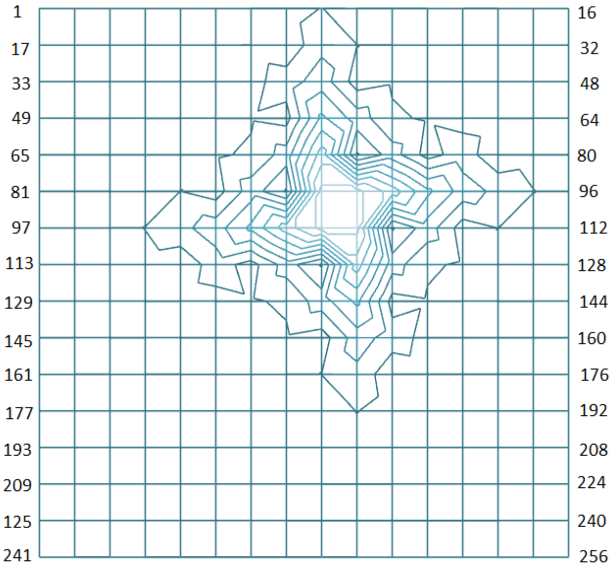


Figure 5: Failure segregation. Each contour line corresponds to one more hop from source to destination. Outside the contour lines, the Average Number of Hops (ANH) is less than 26. Inside all lines, the ANH is less than 34.

D. Failure effect distribution map

The last calculation performed was the failure distribution effect. In case of failure, the symmetry is broken and the mean number of hops is no longer the same for any destination. Then, it was necessary to calculate the mean number of hops executed by an arbitrary packet addressed to all the 256 nodes. The overall mean value was considered to be the arithmetical median of those previously calculated values. Considering the full load traffic condition (100% link load), the map in Figure 5 shows an important distribution characteristic. The map shows nodes 1 to 16 in the first line and 16 nodes per line up to node number 256. The failure occurs in a link belonging to the clockwise sub-domain connecting nodes 89, 90, 106 and 105. Most of the destination nodes are not perturbed by the failure and remain with the same average number of hops (ANH) they had before failure (ANH<26). The ANH increases only for the destinations near the failure. Outside the contour lines, the ANH is less than 26. Crossing one contour line, the ANH is less than 27. Increased by one unit after crossing each contour line, the ANH will be less than 34, near failure, after crossing 8 contour lines.

V. CONCLUSION AND FUTURE WORK

The approach of treating large number of nodes network as a complex system, working as a bottom-up organization system, was analyzed with a statistical analytical model and a simulation model. It was possible to investigate a Protection Emerging Function. Protection is achieved by local signalization that modifies only the nodes operations around the failure. No signalization needs to be transmitted over a long distance regardless of the size of the network. A map with the number of hops after failure illustrates that the network performance degradation occurs only around the failure. The segregation of the failure effects represents a new feature that could be observed due to the new approach. All results can be reproduced for any topology. Reference [13] shows preliminary results for the National Science Foundation Network (NFSnet) treated as a complex system in a bottom-up type of organization. Future work will furnish more details about the complex behavior trough the utilization of the same statistical analysis. With more nodes new Emerging Functions can be emphasized [1]. Several new features can be proposed or investigated. Traffic distribution, protection and restoration functions can also be analyzed as Emerging Functions. The bottom-up organization and the complex system treatment permits better performance and to increase the robustness of large number of nodes network.

ACKNOWLEDGMENT

The authors thank FAPESP – The State of São Paulo Research Foundation – sponsor of the KyaTera Project. The authors also thank Lucas Pauli Simões for the contributions to the simulation results.

REFERENCES

[1] A. Sachs, C.M.B. Lopes, and T.C.M.B. Carvalho, Protection schema for optical packet switching network with large number of nodes, Microwave and Optoelectronics Conference (IMOC) 2009 SBMO/IEEE MTTS-International, 3-6 Nov 2009, pp. 47-50.

[2] P. Baran, On Distributed Communications Netwoks, IEEE Transactions on Commnications Systems, CS-l2 (1964), pp. 1-9.

[3] C. Guillemot, M. Renaud, P. Gambini, C. Janz, I. Andonovic, R. Bauknecht, B. Bostica, M. Burzio, F. Callegati, M. Casoni, D. Chiaroni, F. Clerot, S.L. Danielsen, F. Dorgeuille, A. Dupas, A. Franzen, P.B. Hansen, D.K. Hunter, A. Kloch, R. Krahenbuhl, B. Lavigne, A. Le Corre, C. Raffaelli, M. Schilling, J.C. Simon, L. Zucchelli, Transparent Optical Packet Switching: The European ACTS KEOPS Project Approach, J of Lightwave Technology, 16 (1998), pp. 2117-2134.

[4] L. Dittmann, C. Develder, D. Chiaroni, F. Neri, F. Callegati, Member, IEEE, W. Koerber, A. Stavdas, M. Renaud,, A. Rafel, J. Solé-Pareta, W. Cerroni, N. Leligou, Lars Dembeck, B. Mortensen, M. Pickavet, N. Le Sauze, M. Mahony, B. Berde, and G. Eilenberger; The European IST Project DAVID: a Viable Approach towards Optical Packet Switching; JSAC Special Issue on High-Performance Optical/Electronic Switches/routers for High-Speed Internet II. IEEE Journal on Selected Areas in Communications, 21 (2003), pp 1026 – 1040.

[5] C. Stamatiadis, M. Bougioukos, A. Maziotis, P. Bakopoulos, L. Stampoulidis and H. Avramopoulos, All-Optical Contention Resolution using a single optical flipflop and two stage all-optical wavelength conversion, paper OThN5 Proceedings of OSA / OFC/NFOEC 2010. Available at: <http://www.photonics.ntua.gr/PCRL_web_site/OFC_10_OThN5.pdf>. Retrieved: July, 2012.

[6] J.M. Carlson and J. Doyle, Complexity and Robustness, Proceedings of the National Academy of Sciences - PNAS, February 19, vol. 99, suppl. 1, 2002, pp. 2538–2545.

[7] A.L. Barabási, The Architecture of Complexity, IEEE Control Systems Magazine, 27(2007), pp. 33-42.

[8] D.L. Turcotte and J.B. Rundle; Self-organized complexity in the physical, biological, and social sciences, in Proceedings of the National Academy of Sciences – PNAS, February 19, vol. 99, suppl. 1, 2002, pp. 2463–2465.

[9] I. Prigogine, Le leggi del caos, Roma-Bari, Editori Laterza, 1993.

[10] A.G.Greenberg and J.Goodman, Sharp approximate models of adaptative routing in mesh networks. Telegraffic Analysis Computer Performance Evaluation. Elsevier Science -North Holland, pp. 255-269, 1986.

[11] H. Pistori; J.J. Neto; M.C. Pereira, Adaptive Non-Deterministic Decision Trees: General Formulation and Case Study. INFOCOMP Journal of Computer Science, Lavras, MG, 2006. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.1885&rep=rep1&type=pdf>. Retrieved: July, 2012.

[12] A.S. Acampora and S.I.A. Shah, Multihop lightwave network: a comparison of store-and forward and hot potato routing, IEEE Transactions on Communications, 40(1992), pp. 1082-1090.

[13] A.C. Sachs, Rede auto-organizada utilizando chaveamento de pacotes ópticos. 2011. Doctoral Theses (Digital Systems) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2011. Available at: <http://www.teses.usp.br/teses/disponiveis/3/3141/tde-05082011-152444/>. Retrieved: July, 2012.

ISBN: 978-1-61208-239-4

Fighting Botnets - A Systematic Approach

Nuno G. Rodrigues, António Nogueira and Paulo Salvador
Instituto de Telecomunicações/University of Aveiro
Campus de Santiago, 3810-193 Aveiro, Portugal
E-mail: nuno@ipb.pt, {nogueira, salvador}@ua.pt

***Abstract*—The increasing impact of Internet in the global economy has transformed botnets into one of the most feared security threats for citizens, organizations and governments. Despite the significant efforts that have been made over the last years to understand this phenomenon and develop detection techniques and countermeasures, this continues to be a field with big challenges to address. The most important detection approaches and countermeasures that have been proposed are usually oriented to address some specific type of botnet threat or fight botnets in particular scenarios or conditions. This paper proposes a generic and systematic model to describe the network dynamics whenever a botnet threat is detected, defining all actors, dimensions, states and actions that need to be taken into account at each moment. We believe that the proposed model can be the basis for developing systematic and integrated frameworks, strategies and tools to predict and fight botnet threats in an efficient way.**

Keywords*-network security; malware; botnet; network resilience.*

I. Introduction

In the last decades, communication networks, and Internet in particular, incredibly expanded their usage, importance and impact levels in the global economy. Nowadays, significant parts of our daily lives are directly or indirectly related with the Internet, with the use of services like the e-mail, online news or entertainment, teleworking, business transactions, home banking, social networks and much more. This level of dependence raised this network to the level of a global critical infrastructure, where possible failures and disruptions have a tremendous impact in the global economy. If the Internet relevance in current society increases very fast, motivations for launching cyber-attacks and the Internet vulnerability level increase even faster. In many aspects, the new level of importance was not accompanied by the increase of reliability, availability and security [1] or, in other terms, of the network resilience [2].

From the three disciplines that mainly characterize network resilience, security is the most challenging. In fact, the range of security threats that can affect Internet is immense and increasingly complex, reinforced with the beginning of a new era where cyber-war between nations is a reality. One recent example of this situation were the massive Distributed Denial of Service (DDoS) attacks deployed against Georgia governmental Web sites during the summer of 2008, coinciding with the movement of Russian troops into the Georgian province of South Ossetia [3] or the recently discovered Stuxnet botnet [4] that was specifically developed to sabotage the Iranian uranium enrichment infrastructure.

Network security is a very broad topic that includes issues like confidentiality, authenticity, integrity, authorization or non-repudiability. The lack of security of computers and networks is created, in a first instance, by the existence of vulnerabilities that can become a threat. Threats can become attacks, which can result in compromised systems. One of the most common security threats in current networks and computer systems is the use of software with malicious functionalities, known as *malware* [6]. Malware is a generic term that encompasses specific malicious pieces of software like rootkit, virus, worm, spyware, trojan horse, sniffer and many others. A large set of infected computers (bots) that is remotely and coordinated controlled by an attacker (botmaster) is known as a botnet. Although botnets are used for many different malicious purposes, nowadays the most relevant uses are for political and financial benefits [6].

During the last years, several techniques were developed to detect botnets in local networks. These techniques are usually divided among passive and active [6]: passive techniques are only based on monitoring and observation, acting transparently without interfering with the botnet environment, while active techniques use approaches that interact with the environment under observation and monitoring. Whenever a botnet is detected, it is necessary to deploy appropriate countermeasures that should limit the threat and/or eliminate it. Countermeasures can be grouped into three main categories: technical, regulatory and social methods [7].

Although the identification of possible countermeasures that can fight and remove botnet threats in a local network is nowadays reasonably well achieved, their systematic application needs to be significantly improved. Cleaning infected machines using anti-virus software, applying traffic filtering rules or blocking network elements' ports are relatively common measures taken by network administrators in the case of a botnet detection. However, since these threats become more and more complex and sophisticated, the fighting procedures need to be systematized and automated. Besides, having the ability to model all network states (from a security perspective) can help predict future network states/behaviors based on available (input) events. This systematization will facilitate the deployment of automated countermeasures for any detected threat. This paper proposes a generic network model that is able to describe the different network dynamics under the presence of a botnet threat: all actors, dimensions, states and actions that need to be taken into account at each moment will be defined, allowing the development of appropriate inference procedures that can infer the values of different model parameters based on real data.

The paper is organized as follows. Section II presents the most relevant background on botnet infrastructures, detection approaches and countermeasures; Section III presents the network modeling approach, including all possible network states and all the actions that originate state transitions, besides discussing the necessary steps to infer the model parameters and use it to help network managers and administrators; finally, Section IV presents the main conclusions and topics for future work.

II. Background on Botnets

A botnet is a large collection of computing systems that is infected with the same piece of malware (bot) and is remotely controlled by one or more attackers (botmasters), using a specific command-and-control (C&C) infrastructure [1], with the purpose of performing malicious actions like sending spam email, triggering distributed denial-of-service attacks (DDoS), capturing private information or propagating other types of malware. Infected computers and networks become unstable and, frequently, unable to operate normally.

Nowadays, it is estimated that millions of infected systems exist in the Internet, being part of thousands of botnets [8]. According to Fossi *et al.* [9], the Rustock botnet, for example, controlled more than 1 million bots. If in the last years economic benefit has been the major motivation for botnet deployment, recently we are witnessing its use for political purposes [3], [4] and for several underground cybercrime activities [6]: unsolicited mass mailing (spam), click frauds and pay per install, identity theft, DDoS.

A. Botnet infrastructures

The botnet C&C infrastructure includes bots and a control entity, using an addressing mechanism and one or more protocols to maintain a communication channel and distribute commands between the infected computers and the botmasters [10]. The C&C infrastructure can have a centralized, decentralized or locomotive based architecture.

In a centralized architecture, bots act only as clients, connecting and receiving commands from one or more servers. This architecture is based on a client-server communication model, where HTTP and IRC are the most common communication protocols [11]. Centralized infrastructures can be based on single central C&C servers or in a multi-layered structure of servers and bots. In this second alternative, servers can be divided into different roles: some can be used for command and control and others for delivering contents to bots. Bots can also perform different roles in the botnet structure.

In decentralized architectures, also known as peer-to-peer architectures, there is no differentiation between clients and servers. All nodes participating in the botnet perform the same set of roles, being known as peers. The communication protocol is also based in peer-to-peer models. With this architecture, botmasters control bots by inserting commands and updates in an arbitrary point of the botnet, which makes their localization almost impossible and provides a very high degree of anonymity. There are no central servers to mitigate and disable. However, the propagation of commands through the botnet is slower when compared to centralized approaches. There are some botnets that use hybrid infrastructures, with a centralized infrastructure as the primary option and an alternative peer-to-peer backup channel.

Locomotive botnets use a central C&C infrastructure that is constantly moving over time. This means that the C&C servers are continuously changing, with the support of the DNS service [6].

A highly complex DNS-based technique was used by botnet developers to increase botnet resilience and anonymity: the so called fast-flux service [6]. With this service, it is possible to use several bots as proxy servers to transparently forward malicious communications from clients to a malicious server. The proxy servers hide to the outside the malicious services that are available in the malicious server. The main characteristic of this mechanism is the use of round-robin DNS with very short TTL values associated with the DNS resource records in order to rapidly and continuously change the IP addresses of the bot proxies, being extremely difficult to follow and intercept these communications.

B. Botnet detection and countermeasures

Since botnets act with discretion, their detection is very challenging. One of the solutions that have been used for botnet detection and tracking is based on honeynets [12], a set of honeypots. A honeypot is an intentionally insecure computational system that is placed in the network with the objective of detecting and capturing traffic from botnets in order to understand their characteristics and *modus operandi*. The most important botnet detection techniques that have been proposed are based on passive monitoring and analysis of the network traffic, and can be classified into four main categories [13], [14]:

- Signature-based: these techniques are based on previous knowledge about malware and botnets [15]. One known example is the Snort [16] tool, an open source intrusion detection system (IDS). The main drawback of this type of systems is that they can only detect known botnets and malware.
- Anomaly-based: these techniques are based on the detection of traffic anomalies, like high volumes of traffic, high delay or jitter, unusual ports or unusual system behaviour [8]. However, if the botnet traffic seems to have normal patterns, this type of methods cannot detect it. Botsniffer [17] is an anomaly-based detection tool.
- DNS-based: these techniques apply the same principles of the anomaly-based techniques to the specific case of DNS traffic.

- Mining-based: Since the other techniques are not effective to detect C&C traffic, this approach uses data mining techniques to perform this identification. Masud *et al.* [18] presented a very promising data mining identification methodology.

When a botnet is detected, it is necessary to do all the possible to mitigate the threat, taking measures to shut it down if possible. Because of the dissimulated nature of these systems, this is a challenging task. The most common approach is based on searching for central weak points in the botnet infrastructure that can be disrupted or blocked. In general, two main approaches exist: classical countermeasures and offensive strategies [10]. In the classical countermeasures group, the three most common used techniques are:

- Taking down the C&C server. Whenever possible, this is the most effective and fast way to shut down the botnet. However, it is only applicable to botnets with a central infrastructure and if the location of the C&C server is known. The cooperation of the service provider where the server is connected to is fundamental in this step. Besides, depending on the botnets, bots can be prepared to spread and perform tasks autonomously, without communicating with the C&C server.
- Sinkholing malicious traffic. If shutting down the C&C server is not possible, the traffic between bots and this server can be redirected to a sinkhole. This can be done at the routing level, either in a local or global scale, obviously depending on the cooperation between organizations and ISP's.
- Cleaning infected systems. Although, this is the most sustainable measure to eliminate a botnet threat, it is also the most difficult due to the extremely large spectrum of client systems that normally are infected, covering many different geographical areas, different types of users, etc. The most common approaches are based on the use of up-to-date anti-virus and personal firewalls in the end user systems. However, usually these tasks are not controlled by the network and system administrators, which makes them so difficult to implement.

The effective implementation of classical countermeasures clearly depends on the organizational and political cooperation between different entities, which is usually a slow process when compared to the urgency that is required to fight these threats. Additionally, the most recent botnet threats use increasingly sophisticated obfuscation techniques that make the application of classical countermeasures even more difficult. To solve these limitations, some new proactive offensive approaches have been proposed [10]:

- Mitigation: an offensive approach against the botnet infrastructure, similar to temporary DoS attacks to C&C servers, trapping and blocking connections from infected machines or malicious domains.
- Manipulation: this approach relies on bugs found in bots to access the C&C channel, intercepting commands and forge new fake commands to change their behavior. In the limit, fake commands can order the bots self-destruction.
- Exploitation: this approach explores bugs in the C&C servers or even in the bots to gain control over them and promote their destruction from inside.

Despite being technically feasible and very effective, these types of techniques raise several ethical and legal questions, as the name (offensive) suggests. If fact, the use of these techniques usually implies the unauthorized access to infected machines and infrastructures, which means using the same (and many times illegal) rules as the attackers. An example of this complexity is the recent action of FBI to take the control of the C&C servers of the DNSChanger trojan.

Chainey [19] proposed a new approach for collective cyber threat defense efforts based on the public health models that are used in several countries. In this proposal, the authors defend the use of health certificates for all systems connected to the Internet. These certificates demonstrate the health condition of each device and can be used by service providers to allow or block access to specific resources (like home banking platforms, for example). Despite being an interesting theoretical approach, many practical questions need to be addressed to implement this model, ranging from the specification of certificates and protocols to the construction of a global infrastructure that can manage the system.

Another different and innovative approach is described in [20], where where Li and Liao proposed the idea of using virtual bots to create uncertainty in the attack capacity of each botnet. This study advocates that this uncertainty has a significant impact on the profits of botmasters and attackers, which means that the economic benefits can be destroyed or mitigated and the corresponding interest in using the botnet will automatically decrease.

III. Botnet fighting - modeling the response

Formal models that support systematic and methodical approaches are an important tool to improve computer and network security in general [21] and, we believe, to efficiently fight botnets. This section will present a network model that can describe the different network states, according to the degree of botnet infection that is detected, and the actions that lead to state transitions. The finite state machine model that is proposed includes a detailed characterization of the possible states of all network elements (hosts, switches, routers), allowing a rigorous and precise knowledge of the network operation details at any given time instant. The nature of the proposed model allows its use in the prediction of the network states at future time instants.

As a base discipline that affects network resilience, security issues can be addressed using the two-phase ResiliNets strategy D^2R^2+DR described in [2]. The first phase of this strategy (D^2R^2) runs in real-time and corresponds to the **D**efend, **D**etect, **R**emediate and **R**ecover steps, while the second phase (DR) runs in background and includes the **D**iagnose and **R**efine steps. Considering this strategy, our current work is based on the following assumptions:

1) Network and host defenses can be broken and hosts can be infected by malware, becoming members of botnets;
2) Actual techniques and resources can detect the infection of hosts and the presence of botnet activities in a local network.

This means that the work will be focused in modeling the *Remediate* and *Recover* steps of the ResiliNets strategy, in the presence of botnet threats.

A. The network model

In a first step, the problem will be limited to the perspective of a local network, where it is necessary to model the response behavior of the following actors: switches (from core, distribution and access layers), routers and hosts. A local network that is facing a possible botnet infection can be described by a sub-set of the following states and transitions:

- *Normal state*: in this state, the network is working according to its baseline, without strange events originated by the presence of malware running on hosts. The transition to another state is affected by the following transitions:
  - *Botnet Infection*: if a botnet infection is detected, the network changes from the Normal to the Impaired state;
  - *Massive Botnet Infection*: if an unexpected massive botnet infection is detected, the network changes directly from the Normal to the Generalized Infection state;
- *Impaired state*: some infections on local hosts were detected but their impact in the overall network performance and security is not very significant. The transitions that affect this state are:
  - *Increased Botnet Infection*: if the previously detected botnet infection increases significantly, the network needs to change from the Impaired state to the Generalized Infection state;
  - *Recovery measures*: the deployment of adequate recovery measures was able to eliminate the security threat, allowing the network to recover to the Normal state.
- *Generalized Infection state*: a significant infection was detected on local hosts, with a big impact on the overall performance of the local network. The transitions from this state are affected by the following actions:
  - *Remediation measures*: the deployment of remediation measures that confine the problem inside certain acceptable levels allow the network to return to the Impaired state;
  - *Recovery measures*: the deployment of adequate recovery measures that definitively eliminate the threat allow the network to recover to the Normal state.
- *Quarantine state*: the previous detection of a significant infection on local hosts implied the quarantine of the network, blocking all traffic exchanged with other IP networks in the gateway. The transitions from this state are affected by the following action:
  - *Recovery measures*: the deployment of adequate recovery measures that definitively eliminate the threat allow the network to recover to the Normal state.

Figure 1 graphically represents the finite state machine that includes the four states that were presented and the transition actions between them.

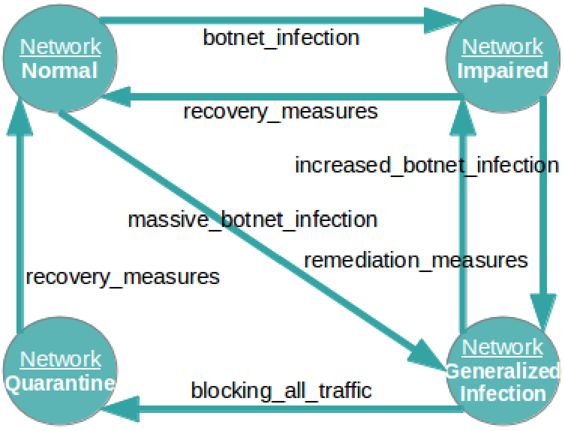


Figure 1. Finite state machine of the local network.

When considering the perspective of an individual host of the local network, the following states and transitions can be identified:

- *Normal state*: the host is not infected with malware. The following transition action will affect this state:
  - *Malware infection*: the detection of malware implies the change of the host to the Infected state.
- *Infected state*: some piece of malware was detected at the host. This state if affected by the following transition actions:
  - *Automatic clean system*: if automatic defenses are able to fight this infection, the system can return to the Normal state;
  - *Filtering malicious traffic*: if the defensive actions cannot automatically clean the system, the malicious traffic must be filtered and the host state will change to Quarantine.
- *Quarantine state*: if the infection cannot be automatically removed, the host must be quarantined. This state can be changed by the following actions:
  - *Manual clean system*: if a manual cleaning of the system with existing tools (like anti-virus) is successful, this implies the host transition to the Cleaned state;
  - *Block all network traffic*: if manual cleaning with existing tools is not possible and additional and more complex tasks are needed, the host transits

to a disconnected mode, with the consequent blocking of all network traffic in the corresponding switch port.

- *Disconnected state*: if the infection cannot be controlled in a short time and is affecting the security and performance of other external elements, then the host must be temporarily disconnected from the network. This state can be changed only by following action:
  - *Offline clean system*: the system is cleaned with the available tools and resources, definitively eliminating the threat. In some cases, a complete system formatting and re-installation might be necessary.
- *Cleaned state*: after the quarantine or disconnected period, the host transits to the cleaned state, where all the previously applied contention measures are removed. The following action will change the system to the Normal state:
  - *Permit all network traffic*: when the threat is definitively eliminated from the host, all the traffic filters that were previously activated can be removed and the host will transit again to normal operation.

The finite state machine that represents all these states and transitions is represented in Figure 2. The dashed lines correspond to actions that occurred in other actors: *filtering_malicious_traffic* is applied at the gateway, while *block_all_network_traffic* is applied at the switch interface. The action *permit_all_network_traffic* is applied in both the gateway and the switch.

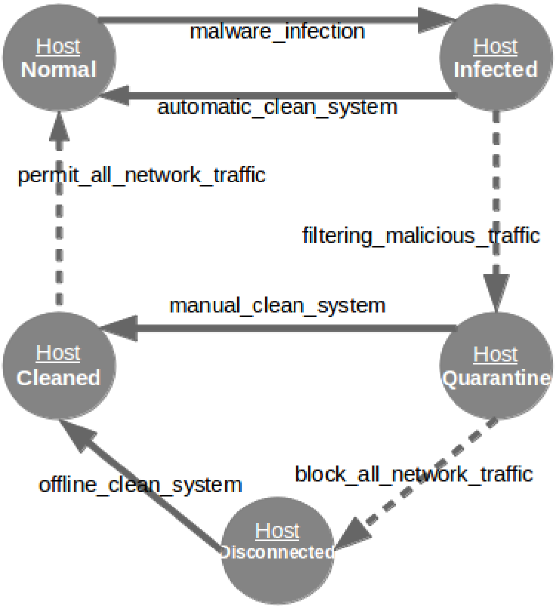


Figure 2. Finite state machine of an individual host.

In the same way, it is relevant to identify and characterize the states of layer two devices (LAN switches). Since these devices physically interconnect the network hosts, they represent the first point available to control the connect/disconnect tasks corresponding to each host:

- *Normal state*: if no actions are taken to disconnect a host from the network, all switch ports are in normal operation (enabled). The following action changes this state:
  - *Block interface*: if the host transits from the Quarantine to the Disconnected state, the corresponding switch interface needs to be blocked (disabled), transiting the switch to the Blocking state.
- *Blocking state*: if a host needs to transit from the Quarantine to the Disconnected state, the corresponding switch port is disabled, blocking the physical connectivity for that host. This state remains active until no more switch interfaces are disabled due to this reason. The following action changes this state:
  - *Release interface*: if no more switch ports are disabled, the switch will come back to the Normal state.

Figure 3 shows the finite state machine corresponding to LAN switches.

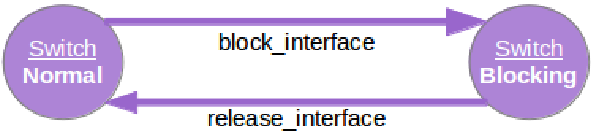


Figure 3. Finite state machine of the LAN switches.

The last relevant actor is the router that interconnects different IP networks of the LAN. The states and actions that characterize this device are:

- *Normal state*: if no malware activities were detected in the local network, the router is operating in the normal state. The following action will change its state:
  - *Filtering malicious traffic*: if malicious activities were detected in the local network and cannot be automatically removed, it is necessary to activate filters that can prevent malicious traffic from going outside.
- *Filtering state*: in this state, the router is filtering malicious traffic and its state can change due to the following actions:
  - *Remove traffic filters*: this action occurs if the threat was definitively removed from the local network. This implies changing the router state to Normal.
  - *Blocking all traffic*: if the threat increased significantly and cannot be contained by using only filters for malicious traffic, it can be necessary to activate more restrictive filters that block all traffic until the threat is eliminated. In this case, the router transits to the Blocking state.
- *Blocking state*: the router is in this state if one or more interfaces need to block all traffic. The Router leaves this state by the influence of the following action:

- *Permit all network traffic*: this action removes the filters that are blocking all traffic from one or more router interfaces and is activated whenever the threats that previously implied the activation of these filters are definitively eliminated.

Figure 4 shows the finite state machine corresponding to the router.

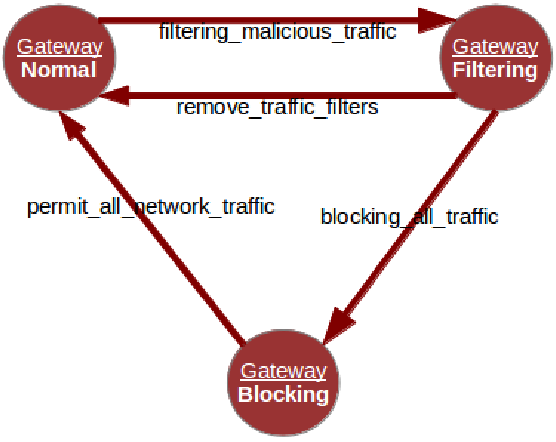


Figure 4. Finite state machine of the local router.

From this discussion, it is clear that the states and transition actions corresponding to the three identified actors are completely interrelated. Figure 5 tries to map the finite state machine of each individual actor with the finite state machine of the network as a whole. The dashed lines represent transitions of an actor from one state to another caused by actions that occurred in another different actor. For example, the Host transits from the Infected to the Quarantine state by the effect of action *filtering_malicious_traffic* that is applied in the Gateway.

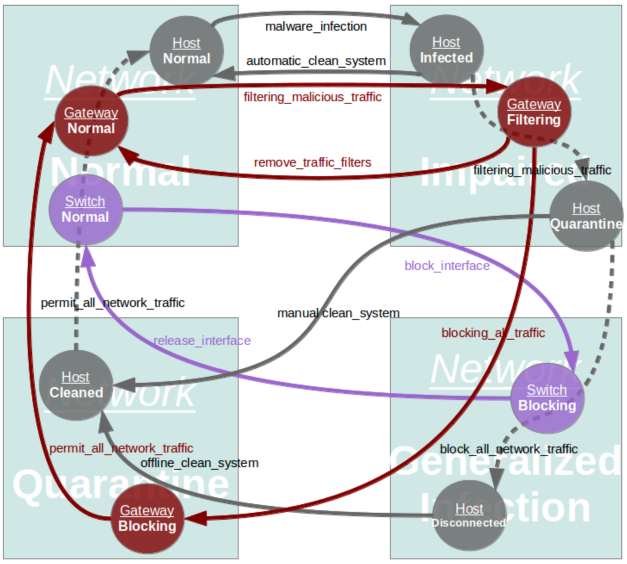


Figure 5. Finite state machine of the overall local network.

The knowledge of the real network state, influenced by the presence of botnet activities, is fundamental to take the right decisions and apply the most effective countermeasures. This knowledge is only possible after inferring all the network model parameters from real and/or reliable network data.

B. From the inference of the model parameters to network management

In a first phase, network data reflecting normal activity and anomalous behaviors induced by the presence of different botnet types should be collected, analyzed and correlated in order to understand which anomalies have occurred and how they can be characterized. The characterization of each anomaly should be as complete as possible, including the amount of data that is generated (alert messages, traffic amount on the different network links, anomalous information on log files, etc.), the timing parameters associated to the anomaly (like, for example, the duration of its characteristic segments) and the transition probabilities between the different states that characterize the anomaly, among other relevant statistics. The data collection step should involve the deployment of laboratorial testbeds where the different security threats can be easily installed in a controlled environment, analyzed and characterized.

The network modeling framework is a multistage space state process able to model the number of error or alert messages and the different states of the network in terms of security threats. Each state is characterized by the type of generation process (deterministic, exponential or other) and its corresponding parameters. The dynamics of the state transitions are heterogeneous and can be ruled by deterministic or exponential processes that define the time of permanence in each state and the destination of the next transition. The modeling framework parameterization will agree with the assumption that state transitions can follow a deterministic or random distribution. State transitions are ruled in parallel by two (or more) parametric matrices that define, respectively, the next transitions after a deterministic amount of time and the probabilistic transitions after a random period of time. The probabilistic/random transitions can follow an exponential distribution (like happens in Markovian models [23] [24]) or any other distribution. The information generation processes associated with each state will also be parameterized by two (or more) vectors defining, respectively, the deterministic values and distribution function parameters for the rates and amount of alert messages generated.

The chain modulated nature of the modeling framework will allow the use of traditional mathematical tools to obtain the model resulting from the superposition of several models or predict the network state at future time instants. The superposition of multiple models (corresponding to different independent networks or different network segments where a certain level of independence can be assumed) can be easily calculated using simple Kronecker sum and product operations [25]. Besides, the chain nature of the resulting model will facilitate the prediction of future network states.

Taking these advantages into account, we believe that the developed network model can be the basis for new tools that can be intensively used in several network operational and management tasks. The proposed framework can help network managers plan short-term or long-term network reconfigurations and upgrades or design new strategies for network management, traffic routing, service provisioning and other critical network operational issues. The correct planning and location of network failures due to security flaws can greatly increase network operation efficiency and optimize Quality of Service (QoS) parameter values.

IV. Conclusion and Future Work

This paper proposed a network model that is able to describe all network states and the network dynamics in the presence of security threats, specially those originated from botnets, being the first step for a more embracing objective, the development of an integrated framework that is able to identify threats and deploy appropriate countermeasures. All actors, dimensions, states and actions that need to be taken into account at each moment were defined, allowing the future development of appropriate inference procedures that can infer the different model parameters based on real data. Having the ability to model all network states (from a security perspective), events and transitions will be extremely important for network administrators and end users, helping them choose the most appropriate actions/countermeasures for each specific situation. The next step in this work will involve the identification of all relevant network actors and events and the inference of the finite state machine parameters, including the event generation distribution corresponding to each state and the transition probabilities between states.

Acknowledgement

This research was supported by Fundação para a Ciência e a Tecnologia, under research project PTDC/EEA-TEL/101880/2008.

References

[1] S. Goodman and H. Lin, *Toward a Safer and More Secure Cyberspace*, National Academies Press, 2007.

[2] J. P. G. Sterbenz, D. Hutchison, E. K. Cetinkaya, A. Jabbar, J. P. Rohrer, M. Scholler, and P. Smith, *Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines*, Elsevier Computer Networks, vol. 54, Issue 8, pp. 1245-1265, June 2010.

[3] S. W. Korns and J. E. Kastenberg, *Georgia's Cyber Letf Hook*, 2009.

[4] N. Falliere, L. Murchu, and E. Chien, *W32.Stuxnet Dossier*, Version 1.4, Symantec Security Response, February 2011.

[5] C. E. Landwehr, *Computer Security*, International Journal of Information Security, 1, pp. 3-13, 2001.

[6] D. Plohmann, E. Gerhards-Padilla, and F. Leder, *Botnets: Detection, Measurement, Disinfection & Defense*, European Network and Information Security Agency (ENISA), 2011.

[7] *ITU Botnet Mitigation Toolkit*, ICT Applications and Cybersecurity Division, Policies and Strategies Department, ITU Telecommunication Development Sector, 2008.

[8] B. Saha and A. Gairola, *Botnet: An Overview*, CERT-In White Paper CIWP-2005-05, 2005.

[9] M. Fossi, G. Egan, K. Haley, E. Johnson, T. Mack, T. Adams, J. Blackbird, M. Low, D. Mazurek, D. Mckinney, and P. Wood, *Symantec Internet Security Threat Report: Trends for 2010 (Volume 16)*, Symantec Corp, April 2011.

[10] F. Leder, T. Werner, and P. Martini, *Proactive Botnet Countermeasures: an Offensive Approach*, The Virtual Battlefield: Perspectives on Cyber Warfare 3. pp. 211-225, 2009.

[11] M. Fossi, D. Turner, E. Johnson, T. Mack, T. Adams, J. Blackbird, S. Entwisle, B. Graveland, D. Mckinney, J. Mulcahy, and C. Wueest, *Symantec Global Internet Security Threat Report: Trends for 2009 (Volume XV)*, Symantec Corp, April 2010.

[12] Honeynet Project and Research Alliance website. [Online]. Available: http://www.honeynet.org [Accessed: 17 April 2012].

[13] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and A. Arbor, *A Survey of Botnet and Botnet Detection*, Third International Conference on Emerging Security Information, Systems and Technologies, IEEE Computer Society, 2009.

[14] M. Feily, A. Shahrestani, and S. Ramadass, *A Survey of Botnet Technology and Defenses*, CATCH '09 Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security, IEEE Computer Society, 2009.

[15] Y. Xie , F. Yu , K. Achan , R. Panigrahy , G. Hulten, and I. Osipkov, *Spamming Botnets: Signatures and Characteristics*, ACM SIGCOMM Computer Communication Review, vol. 38 no. 4, 2008.

[16] Snort website. [Online]. Available: http://www.snort.org [Accessed: 2 May 2012].

[17] G. Gu, J. Zhang, and W. Lee, *BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic*, Proceedings of 15st Annual Network and Distributed System Security Symposium, 2008.

[18] M. Masud, T. Al-khateeb, L. Khan, B. Thuraisingham, and K. Hamlen, *Flow-based identification of botnet traffic by mining multiple log files*, Proceedings of International Conference on Distributed Frameworks & Applications, Penang, Malaysia, 2008.

[19] S. Charney, *Collective Defense: Applying Public Health Models to the Internet*, Security & Privacy, IEEE , vol. 10, Issue 2, pp. 54-59, 2012.

[20] Z. Li and Q. Liao. *Botnet economics: uncertainty matters* In M. Johnson, ed., Managing Information Risk and the Economics of Security: 245-267. Springer, 2008.

[21] C. Landwehr. *Formal Models for Computer Security*, ACM Computing Surveys (CSUR), vol. 13 no. 3, pp. 247-278, 1981.

[22] P. Smith, D. Hutchison, J. Sterbenz, M. Schöller, A. Fessi, M. Karaliopoulos, C. Lac, and B. Plattner, *Network Resilience: A Systematic Approach*, IEEE Communications Magazine, July 2011.

[23] A. Nogueira, P. Salvador, R. Valadas, and A. Pacheco. *Fitting self-similar traffic by a superposition of MMPPs modeling the distribution at multiple time scales*, IEICE Transactions on Communications, E84-B(8), 2134-2141.

[24] A. Nogueira, P. Salvador, R. Valadas, and A. Pacheco. *Hierarchical approach based on MMPPs for modeling self-similar traffic over multiple time scales*, Proceedings of the First International Working Conference on Performance Modeling and Evaluation of Heterogeneous Networks, 2003.

[25] R. A. Horn, and C. R. Johnson. *Topics in Matrix Analysis*, Cambridge University Press, p. 208, 1994.

A Cluster Head Election Method with Adaptive Separation Distance and Load Distribution for Wireless Sensor Networks

Cosmin Cirstea, Mihail Cernaianu, Aurel Gontean,
Applied Electronics Department
"Politehnica" University of Timisoara, Electronics and Telecommunications Faculty
Timisoara, Romania
cosmin.cirstea@etc.upt.ro, mihai.cernaianu@etc.upt.ro, aurel.gontean@etc.upt.ro

***Abstract*— Cluster based wireless sensor networks have the advantage of reduced energy consumption and increased message delivery compared to the situations where no hierarchical communication is used. Cluster heads (CHs) play the important role of performing data gathering and aggregation from surrounding nodes and thus must be efficiently chosen. This paper describes a CH election algorithm for wireless sensor networks (WSNs) based on LEACH that uses adaptive separation distance and load distribution (LEACH-ASDLD) in order to enhance network lifetime and message delivery. The proposed algorithm considers the number of neighbors in the vicinity of each node as well as the expected packet size to be transmitted in electing the appropriate CH, thus distributing network load among key sensors within the network rather than evenly distributing the load among all nodes. Using adaptive separation distance determines the number of CHs per round and ensures their uniform spread over the observation area. In order to determine the importance of using adaptive separation distance combined with load distribution we have performed Matlab simulations and compared our algorithm with a minimum separation distance (MSD) algorithm entitled Improved Minimum Separation Distance (IMSD), an enhancement to MSD. Our simulations show that using the proposed algorithm can extend the lifetime of the network and provide increased message delivery by up to 15% depending on the simulated network and packet sizes.**

Keywords-clustering, adaptive separation distance, load distribution, wireless sensor networks

I. INTRODUCTION

Wireless sensor networks are intelligent networks comprised of hundreds, even thousands of nodes that collaborate to perform various sensing tasks. Their unique characteristics such as low cost, reduced size, low power consumption and rapid deployment make WSNs the best solution for numerous applications such as military surveillance, detection of chemical activity, environmental and healthcare monitoring, to mention just a few. Deploying such networks in dangerous and inaccessible environments allows for the extraction of information which would otherwise be very difficult if not impossible to obtain. However, due to the fact that sensor nodes are battery powered energy consumption at the node and network level is of high importance and represents a major challenge in the design of WSNs. Network lifetime is defined as the time elapsed until the first, half, or the last node in the network consumes its energy. To address the issue of energy efficiency at the network level, intelligent routing protocols are adopted which currently fall under two categories: flat routing protocols and hierarchical routing protocols.

Flat routing protocols consider all nodes within the network as equal and routes are generated through feedback information instead of using a hierarchical management mechanism. The main advantage of this technique is that network traffic is evenly distributed among network nodes [1]. Flat routing protocols include techniques like gossiping and flooding [2], spin [3] and directed diffusion [4].

Hierarchical routing protocols divide the observation area (OA) into smaller areas named clusters and assign representative nodes entitled cluster heads (CHs) that manage the communication within nodes in the cluster and transmit the obtained data to the base station (BS) or to other CHs along the path to the BS. The way clusters are defined, how the CHs are elected and how the communication with the BS is performed, depends on the elected routing protocol. Several representative hierarchical routing protocols for WSNs are LEACH (Low-Energy Adaptive Clustering Hierarchy) [5], TEEN (Threshold sensitive Energy Efficient sensor Network protocol) [6], PEGASIS (Power-Efficient GAthering in Sensor Information Systems) [7] and others more recent that are mostly improvements to the previously mentioned ones.

The rest of this paper is structured as follows: Section II provides a description of the LEACH [5], MSD [8] and IMSD [9] protocols and Section III describes the proposed algorithm LEACH-ASDLD. Section IV describes the experimental setup, as well as the obtained results and Section V presents conclusions and future work.

II. RELATED WORK

A. The LEACH protocol

The Low-Energy Adaptive Clustering Hierarchy (LEACH) [5] is a cluster based protocol that reduces the energy consumption of the sensor network through several key features such as localized coordination and control for the CHs, local compression and randomized rotation of the CHs. The operation of LEACH is divided into rounds (a predefined interval of time during which cluster and inter-cluster communication takes place) and each round begins with a cluster set-up phase preceded by a steady-state phase where data transfer to the base station occurs. The set-up phase is organized as follows:

The advertisement phase – each node individually decides if it becomes a cluster head based on a suggested percentage (P) of cluster heads determined a priori as well as based on the number of times the node has been cluster head so far. By choosing a random number between 0 and 1 the node (n) can elect itself as cluster head if this number is less than a threshold $T(n)$ calculated as follows [5]:

$$T(n) = \begin{cases} \frac{P}{1-P*(r*\,mod\,\frac{1}{P})} & if\ n \in G \\ 0 & otherwise \end{cases} \qquad (1)$$

where r is the current round and G is the set of nodes that have not been cluster heads in the last $\frac{1}{P}$ rounds.

After each round the probability of the remaining nodes must increase since there are fewer nodes eligible to become cluster heads.

Cluster set-up phase – after the cluster heads have been elected, each of them informs neighboring nodes so that each node decides the appropriate cluster head to attend to, based on the received strength of the cluster head advertisement.

Inter cluster communication – each CH creates a TDMA schedule and informs each node from the cluster when to communicate acquired data. To avoid interference with neighboring clusters, each CH also randomly chooses a CDMA code from a list of codes and informs all nodes in the cluster to use the given code.

For simulation purposes the authors of LEACH [5] have chosen the first order radio model where the radio dissipates $E_{elec} = 50\,nJ/bit$ to run the transceiver circuit, $E_{amp} = 100\,pJ/bit/m^2$ for the transmit amplifier and $E_{DA} = 5\mu J/bit$ for data aggregation and fusion. CHs collect k-bit long messages from attending n nodes and compress the data using a compression coefficient c, thus resulting in c•n k-bit messages sent to the BS. A path loss coefficient, $\alpha = 2$, has been considered for each communication. All nodes have the same energy in the beginning, $E_0 = 50mJ$. Thus the energy needed to transmit a k-bit long message over distance d is [5]:

$$E_{Tx}(k,d) = E_{Tx-elec}(k) +\ E_{Tx-amp}(k,d)$$
$$E_{Tx}(k,d) =\ E_{elec} * k + e_{amp} * k * d^2 \qquad (2)$$

The energy required to receive a message is:

$$E_{Rx}(k) = E_{Rx-elec}(k) = E_{elec} * k \qquad (3)$$

Also, the maximum communication range of sensor nodes is 100 meters.

The steady-state (data transmission) phase – all nodes transmit sensed data to the elected CH according to the TDMA schedule they have received. After a certain time determined a priori the next round begins and the protocol resumes from the advertisement phase.

One significant disadvantage when using LEACH is that electing CHs the way previously described does not provide an even distribution of the CHs among the OA. This downside has also been observed by the authors of the MSD protocol which have come up with a solution which we will further describe.

B. The MSD and IMSD protocols

Hansen et al. [8] argue that there should be a minimum separation distance between the CHs in order to provide an even distribution of the CHs throughout the network. In order to test the impact of using a minimum separation distance between the elected CHs the authors have devised an algorithm briefly described Figure 1 [8].

```
MSD = Minimum Separation Distance
dc = Number of desired cluster heads
energy(n) = Remaining energy for node n
avg = $\frac{\sum \text{energy(n)}}{\text{number of alive nodes}}$
eligible = {n| energy(n) ≥ avg}
assert (|eligible| ≥ dc)
CD = {}
while (|CH| < dc)
    if ∃n: n ∈ eligible Λ (∀  m  ∈ CH, dist (m,n)) ≥ MSD
        add (n, CH)
        remove (n, eligible)
    else
        n ∈ eligible
        add (n, CH)
        remove (n, eligible)
    end
end
```

Figure 1. MSD algorithm proposed by Hansen et al. [8]

Their algorithm is based on a variant of LEACH, LEACH-C (Centralized) meaning that the algorithm for the cluster head election is performed by the BS which informs each node in the network about the elected CHs for the current round. In turn, all nodes are obliged to inform the BS about their position information and current energy level each round. Based on this information the BS runs the algorithm and determines which nodes are eligible for becoming CHs for the current round by calculating the average energy remaining in the network. Only nodes with remaining energy above the average threshold become eligible for being CHs for that round. After determining the eligible nodes, CHs are randomly elected based on the minimum separation distance criterion until the desired number is attained. If this number cannot be obtained, random nodes are chosen from the remaining eligible nodes to perform the CH role. After the algorithm has been successfully executed, the elected CHs are informed by the BS of their new status and clusters are formed. Until the next round when the process is repeated the network performs communication the same way as in LEACH.

Simulations have been performed by the authors on a 400x400 meter network and the results have shown that depending on the number of desired CHs, using the MSD protocol results in increased number of messages received by the BS with up to 80% when compared to LEACH.

A more recent research by Chalak et al. [9] proposes an Improved MSD (IMSD) algorithm that solves the MSD issue when the desired number of CHs cannot be obtained without electing CHs that do not obey the minimum separation distance criterion.

To solve this issue the authors claim that the smallest distance allowed between two distinct CHs is the minimum separation distance, which can be smaller than a desired separation distance but should not be larger. If the desired number of CHs cannot be obtained the minimum separation distance is reduced by a pre-defined percentage and the algorithm is implemented again until the number of CHs is obtained. Using this method improvements are obtained compared to the MSD algorithm in terms of network lifetime and overall packet delivery.

Through simulations we have observed that by using either of these two algorithms the desired number of CHs is constantly maintained. An advantage of IMSD over MSD is that the CHs are more evenly spread over the entire area. The eligibility criterion employed by both algorithms gives network nodes equal possibilities to become CHs thus distributing the network load among all nodes. This practically means that the vast majority of nodes will remain without energy at about the same time.

We consider that CHs should be elected also based on other criterions such as the number of neighboring nodes in the area defined by the separation distance and also based on the size of the packet with respect to the allowed maximum size as most WSNs are event driven and modify the transmitted packet size when an event occurs. As simulations presented in the next chapter show, choosing CHs this way can have significant impact on network lifetime and packet delivery.

Using a centralized approach such as LEACH-C can introduce several disadvantages such as nodes that are far away from the BS will have difficulties in sending their status to the BS. If this role is assumed by the CHs for the current round it will induce further strain on those nodes. Either way using a centralized approach will result in increased overhead and communication latency. We consider that a local approach can be more suited as the energy consuming task of sending and receiving messages will be replaced by local computations thus reducing latency and overhead. Also a local approach will allow for scalability in situations where the sensor network is spread over larger areas that extend over the communication range of a node with the BS.

III. Proposed algorithm

To address the previously mentioned issues we have developed the LEACH-ASDLD algorithm which we will further describe.

LEACH-ASDLD is a round based protocol structured on 3 layers of communication:

- Layer 1 represents a neighborhood reconnaissance procedure during which, in the first round nodes inform neighbors of their position information and in all other rounds nodes that remain without energy or newly added nodes inform neighboring nodes of their new status.
- Layer 2 is reserved for sensing and data gathering by network nodes which perform only intra cluster communications based on the TDMA/CDMA proposed schemes.
- Layer 3 is restricted to inter cluster head and cluster head to BS communication.

The following assumptions are made about the network model:

- The network consists of 100 randomly deployed nodes.
- We have performed simulations on areas having sizes ranging from 50x50m to 200x200m with a variation step of 25x25m.
- We assume that knowledge of the observation area size is previously known.
- All nodes are homogeneous, with the same hardware and software architectures and the same battery power.
- Energy consumption constraints are as described in Section I at the description of the LEACH protocol.
- The network is noise and error free.
- Network nodes are synchronized (using an RT Clock for example).

Based on these assumptions we will next provide a more detailed description about the proposed algorithm which can be divided into 4 steps.

1) Neighborhood reconnaissance – each node broadcasts a message with its position if a localization device is present or a dummy message so that other nodes can calculate the distance between themselves and the sending node by using the received signal strength indicator (RSSI) method. This procedure is performed in a TDMA fashion previously defined and only during the first round. In all other rounds the time span for it is reduced and this time window will serve as advertisement space for nodes that do not have enough energy to perform their tasks any more or for newly added nodes to broadcast their position information.

2) Cluster set-up phase – as the size of the OA is known, defining the separation distance between the CHs will actually determine the number of clusters that will be formed each round. Based on the desired number of clusters we have calculated the separation distance between the CHs using the following formula:

$$SD = \sqrt{\frac{L^2}{N}} \qquad (4)$$

For a given square OA of size LxL, the question posed is, if we want to fit N squares within the area, what is the side length of each square (SD). Where N is actually the desired number of CHs.

Using the distance information obtained from Step 1, each node will calculate the ratio between the number of nodes within the separation distance and the number of neighbors in its range. Also each node will calculate the ratio between the expected packet size and the maximum allowed payload size which is 127 bytes according to the IEEE Standard 802.15.4 [10]. Each node calculates a threshold value using the following formula:

$$Th = \left(\frac{E_r}{E_0}\right)\left(1 - \frac{N_{SD}}{N_R}\right)\left(1 - \frac{P_{crt}}{P_{max}}\right) \qquad (5)$$

where E_r is the remaining energy, E_0 is the initial energy, N_{SD} represents the number of nodes in the separation distance, N_R the number of nodes in the sensing range, P_{crt} is the current payload and P_{max} is the maximum payload.

The CH election phase is performed as follows. Initially each node is eligible for becoming a CH if it has enough energy to perform this task. Each node will generate a random number and set a timer according to it. The node with the smallest timer value will be the first advertised CH. Depending on the network size and node sensing range, several CHs can be elected throughout the OA. Nodes that are closer to a CH than the separation distance cannot advertise themselves as CH. Nodes that are farther than the separation distance are still eligible and the node with the smallest T_h will elect itself as CH also using a timer. The procedure is repeated until the entire OA is covered and there are no eligible nodes left. Using the smallest T_h value for electing CHs means that nodes with larger number of neighbors in the separation distance have higher probabilities of electing themselves CH. Using the described method can have a significant impact on the overall message delivery and network lifetime as we will show in Section IV.

Steps 3 and 4, ***Inter cluster communication*** and ***The steady-state (data transmission)*** phases are performed the same way as specified by the LEACH algorithm (Section II).

IV. Simulation Results

In order to determine the impact of the proposed algorithm on the network lifetime and message delivery, we have performed several simulations in different scenarios which we will describe in the following subsections.

A. Network lifetime

As previously mentioned, using an algorithm as MSD [8] or IMSD [9], where the CHs are chosen based on the maximum amount of remaining energy, the overall load of the network is evenly distributed among sensor nodes, which means that the vast majority of nodes will remain without energy at approximately the same time. In LEACH-ASDLD cluster heads are elected with the minimum threshold value, thus straining nodes in key places of the network. To determine the impact of LEACH-ASDLD over the IMSD algorithm in terms of network lifetime we have performed several Matlab simulations on random distributed networks over areas with different sizes that range from 50x50 meters to 200x200 meters with a step of 25x25 meters and a desired number of 10 CHs for each simulation. The results can be seen in Figure 2.

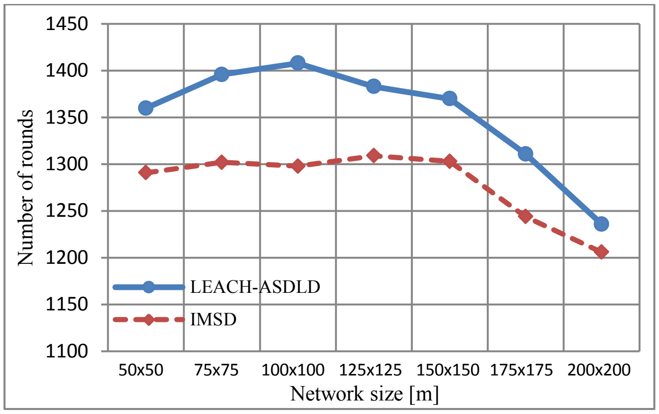


Figure 2. LEACH-ASDLD half nodes die vs. IMSD last node dies.

If we consider the first node dies (FND) metric IMSD outperforms our proposed algorithm, however if we consider the half node dies (HND) metric, as we can see from Figure 2, our proposed algorithm outperforms IMSD as when half of the nodes have died in LEACH-ASDLD, the entire network of nodes using the IMSD algorithm has already depleted all its energy. We can observe that by stressing key nodes in the network LEACH-ASDLD provides extended monitoring time of the OA when compared to the IMSD algorithm which is actually desired in a WSN.

An interesting behavior can be noticed in Figure 2. As mentioned in Section II, CHs serve the purpose of performing data aggregation and fusion operations. Also the maximum communication distance between sensor nodes is of 100 meters and the energy required to send a message is in direct correlation with the distance between the sender and receiver. Nodes send messages either directly to the BS or to the CHs, depending on which is closer. When both are within communication range, sending messages to the CHs can be more costly because of aggregation and fusion operations performed and this behavior can be noticed in Figure 2. This behavior can be avoided by electing the appropriate number of CHs per round depending on network size, however this optimization type is not of the purpose of this paper.

B. Message delivery

To determine the impact of the proposed algorithm on the overall message delivery of the network we have performed several simulations in different scenarios which we will further describe.

1) *Different network sizes*

We have performed simulations on randomly deployed WSNs over square areas of different sizes ranging from 50x50 meters to 200x200 meters with a variation of 25x25 meters. The packet size has been considered static (200 bits/packet) and each node sends a total of 20 packets per round. The number of desired CHs elected per round was 10. The results can be seen in Figure 3.

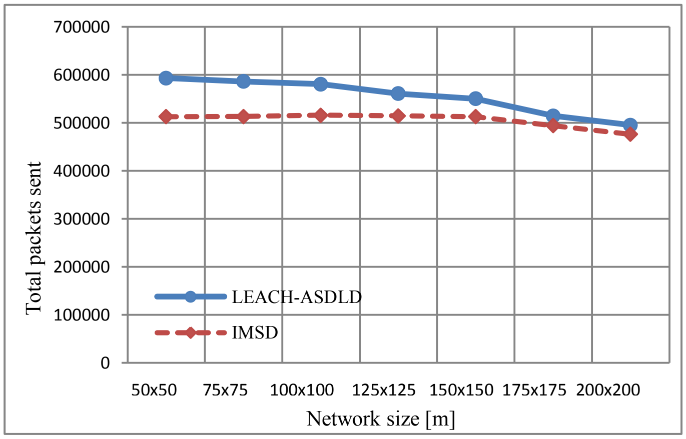


Figure 3. LEACH-ASDLD vs. IMSD – total number of packets sent for different network sizes

As can be seen from Figure 3, there is an increase of packets sent in the network dependent on the size of the OA that ranges from 15% for the 50x50 meters network to 4 % for the 200x200 meters network. This however can be an issue of electing the correct number of CHs as can be seen in the next section.

2) *Different number of CHs*

Electing the optimum number of CHs is another important research direction in the field of WSNs but it does not represent the purpose of this paper. To determine the performance of the proposed algorithm we have performed simulations on a 100x100 meters network with a different number of CHs ranging from 3 to 10 and the obtained results can be observed in Figure 4.

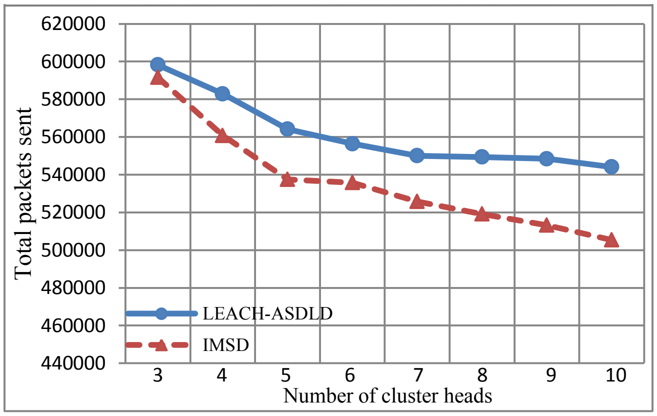


Figure 4. LEACH-ASDLD vs. IMSD – total number of packets sent for different number of CHs

We can see from Figure 4 that varying the number of CHs can have a significant importance over the message delivery of the entire network. Both protocols obtain the highest number of messages delivered for 3 proposed CHs and our proposed algorithm provides only a small improvement of 2% in terms of more messages transmitted. However as the number of CHs is increased LEACH-ASDLD obtains better performance with a maximum of 13% more packets for 10 elected CHs.

3) *Different packet sizes*

In order to determine the impact of the presence of an event in the network which would require an increase in the amount of communicated data we have spread the OA into 4 quadrants, the same as in the Cartesian coordinate system. For a specified number of 300 rounds we have increased the packet size from 200 with a specified percentage (25, 50, 75 and 100) in quadrants 2 and 4 while maintaining the OA at a fixed size of 100x100 meters. During our simulations we have observed that using the IMSD protocol when there is an increase in packet size, due to the election method of the CHs in which only nodes with remaining energy level above the network average are eligible for becoming CHs, the area in which the event takes place is left without any CH, which is not the case in LEACH-ASDLD. We have also noticed through simulation tryouts that increasing the number of CHs in these regions can lead to increased packet delivery as can be seen in Figure 5.

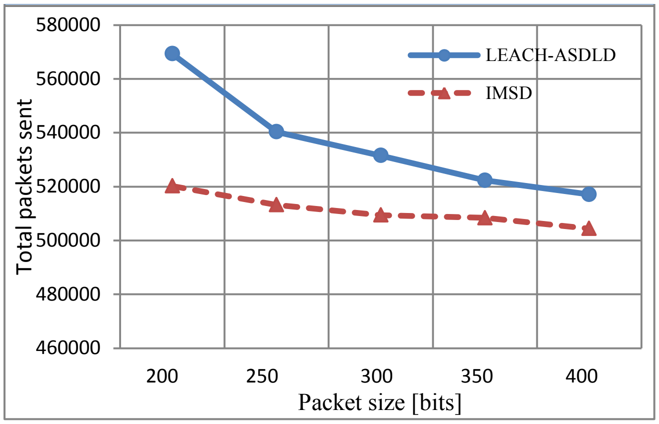


Figure 5. LEACH-ASDLD vs. IMSD – total number of packets sent for different packet sizes

The number of CHs within a certain region can be increased by using an adaptive minimum separation distance with respect to the packet size. We have obtained the best results for the current network distribution by reduced the minimum separation distance with a percentage of 7 (for the 250 bits packet) to 28% (for the 400 bits packet) with a step of 7%. However we have yet to find a direct correlation between optimum number of CHs and the number of nodes in the region/packet size. The obtained results show an increase in packet delivery that ranges from 7% for the 250 bits packet size to 3% for the 400 bits packet size when comparing LEACH-ASDLD with IMSD.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have described a proposed algorithm based on LEACH, LEACH-ASDLD that considers adaptive separation distance between CHs based on the expected packet size and also performs the election of CHs using information about surrounding nodes. We have performed simulations in various network distributions and conditions which we have compared with an improvement to LEACH that also considers a separation distance between CHs, IMSD.

Our simulation results have shown that further improvements can be obtained in terms of network lifetime and messages transmitted throughout the entire network. We have shown that using the proposed method in which key nodes are selected as CHs rather than evenly distributing the energy consumption throughout network nodes can provide extended network lifetime when using the HND metric. Also, the number of packets sent is increased by a factor of 4 to 15% depending on network size, 2 to 13 % depending on the number of desired CHs and 3 to 7% in correspondence with the packet size.

We have also argued that using a centralized protocol such as LEACH-C where network nodes have to inform the BS each round about their status can induce latency and overhead and also does not allow for scalability of the network. Our approach solves the problem of scalability by introducing a time slot in which newly added nodes but also nodes that do not have enough energy can inform neighboring nodes of their status. Using this local information approach expels the need for sending messages over long distances and reduces overhead. There are also several downsides to using our method such as it may require more time before all CHs are elected and also the number of CHs per round is not as stable as when using a centralized approach but will vary slightly (we have observed a maximum variation of ± 10% obtained for 10 desired CHs).

Future work includes determining a correlation between the packet size and the number of CHs (separation distance) as the size of the packet is increased. Also other metrics should be taken into consideration when electing the CHs such as packet frequency variation, expected throughput etc. The energy model used is incomplete and strictly refers to the send/receive of packets and data aggregation/fusion operations performed by cluster heads. This issue should also be improved and other energetic aspects should be taken into consideration such as the power consumed by the microcontroller in different working modes, energy for communication with peripheral devices (sensors), transceiver on/off etc.

ACKNOWLEDGMENT

This work was partially supported by the strategic grant POSDRU 88/1.5/S/50783, Project ID 50783 (2009), co-financed by the European Social Fund - Investing in People, within the Sectorial Operational Programme Human Resources Development 2007-2013.

REFERENCES

[1] S. Dai, X. Jing, L. Li, “Research and Analysis on Routing Protocols for Wireless Sensor Networks”, Proceedings of the IEEE International Conference on Communications, Circuits and Systems, pp. 407-411, 2005

[2] J. N. Al-Karaki, “Routing Techniques in Wireless Sensor Networks: A Survey”, IEEE Wireless Communications Magazine, Vol 11, Issue 6, pp. 6-28, 2004

[3] W. R. Heinzelman, J. Kulik, H. Balakrishnan,“Adaptive Protocols for Information Dissemination in Wireless Sensor Networks”, Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom '99, 1999

[4] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, “Directed diffusion for wireless sensor networking”, IEEE/ACM Transactions on Networking, Vol 11, Issue 1, pp. 2-16, 2003

[5] W. R. Heinzelman, A. Channdrakasan, H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks”, Proceedings of the 33rd Hawaii International Conference on System Sciences, HICSS’00, 2000

[6] A. Manjeshwar, D. P. Agrawal, “TEEN: A routing protocol for enhanced efficiency in wireless sensor networks”, Proceedings of the 15th Parallel and Distributed Processing Symposium, San Francisco, IEEE Computer Science Society, 2001.

[7] H. S. Lindsey, C. Raghavendra, “PEGASIS: Power-Efficient gathering in sensor information systems”, In Proceedings of the IEEE Aerospace Conference 2002, pp. 1-6

[8] E. Hansen, J. Neander, M. Nolin, and M. Björkman, "Energy-efficient cluster formation for large sensor networks using a minimum separation distance", in The Fifth Annual Mediterranean Ad Hoc Networking Workshop, June 2006.

[9] A. R. Chalak, S. Misra, M. S. Obaidat, “A cluster-head selection algorithm for Wireless Sensor Networks”, 17th IEEE International Conference on Electronic Circuits and Systems, ICECS’10, 2010, pp. 130-133.

[10] IEEE Standard 802 for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements, Part 15.4 Wireless Medium Access Control and Physical Layer Specifications for Low-Rate Wireless Personal Area Networks, IEEE Computer Society, 2006, pp 298.

Energy Efficient Wireless Sensor Network System for Localization

Sila Ozen
Department of Computer Engineering
Istanbul Technical University
34469, Maslak, Istanbul, Turkey
ozens@itu.edu.tr

Ture Peken
Department of Computer Engineering
Istanbul Technical University
34469, Maslak, Istanbul, Turkey
peken@itu.edu.tr

Sema Oktug
Department of Computer Engineering
Istanbul Technical University
34469, Maslak, Istanbul, Turkey
oktug@itu.edu.tr

***Abstract*—This paper introduces an enhanced overlapping connectivity-based localization technique. Constructed experiments showed that using the information obtained from closer anchor nodes in the communication range results in better position estimations. In this work, received signal strength indication (RSSI) measurements are used to estimate the distance of an anchor node to a mobile node. The optimal number of anchor nodes to be used in localization is determined empirically with the performed experiments. The enhanced localization technique is implemented on a wireless sensor network system consisting of GenetLab sensor nodes. Each node has MSP430 processor and TinyOS operating system deployed. In order to increase the energy efficiency of the wireless sensor network system, the anchor nodes are operated in active-state or semi-active state based on the position of the mobile nodes. It is shown that such an approach enhances the energy efficiency of the system under various movement scenarios and network topologies.**

Keywords-sensor network; localization; energy efficiency

I. INTRODUCTION

Wireless sensor networks have attracted great interest over the last decade. Low-cost, low power and multifunctional small sensor nodes have become available through the recent developments in electronics and wireless communications [1].

Accurate and low power localization of sensor nodes is crucial for many applications to work efficiently. Various localization techniques have been introduced so far. In general, localization techniques can be classified as measurement and non-measurement based techniques. Additionally, both of these classes are also divided into two sub classes as single-hop and multi-hop techniques. In measurement based techniques, localization is performed by using measured distance with the other anchor nodes using the techniques such as time of arrival (ToA), time difference of arrival (TDoA), and angle of arrival of signals (AoA) [2]. Non-measurement based localization techniques which are also called connectivity-based techniques do not use distance measurements. In the single-hop localization, only one-hop neighbors are considered for localization, whereas in the multi-hop localization, multi-hop distanced anchor nodes are participating. The connectivity-based single-hop localization techniques are easier to implement, more energy efficient than the other techniques and also less affected from the environment. Because of these advantages, in this work, we focus on one-hop distance localization techniques.

The overlapping connectivity-based localization algorithm which is defined by Bulusu et al. [3] is taken as the reference point in our work. The overlapping connectivity uses the location information obtained from anchor nodes to evaluate the location of a mobile node. The mobile nodes estimate their locations in x and y coordinates by taking average of obtained locations of the anchor nodes as shown in (1):

$$(X_{est}, Y_{est}) = \left(\frac{X_{i1} + ... + X_{ik}}{k}, \frac{Y_{i1} + ... + Y_{ik}}{k} \right) \quad (1)$$

where X_{est} and Y_{est} denotes x-coordinate and y-coordinate estimates of the mobile node, X_{i1}......X_{ik} and Y_{i1}......Y_{ik} denote x and y coordinates of the k anchor nodes within the coverage area of the mobile node. The coverage area spans by the anchor nodes within one-hop distance to the mobile node.

In our work, the overlapping connectivity technique is improved by using some of the anchor nodes instead of using whole anchor set within the coverage area of a mobile node. The criteria for selecting an anchor node is based on the received signal strength indication (RSSI) measurements and the direction of the mobile node. Using all of the anchor nodes in one-hop distance to calculate (1) gives rough estimates for localization as seen in simulation results. On the other hand, localization techniques based on only RSSI measurements can give erroneous results depending on environmental conditions. We have implemented overlapping connectivity localization method with a modification of using only a portion of the neighboring anchor nodes by using the RSSI measurements. Based on the results of our simulations we decide about the number of neighboring anchor nodes to be used in our calculations. Moreover, for considering energy efficiency, the anchor nodes are designed to work in two states: semi-active state and active state. In the introduced system, the anchor nodes, which have a mobile node around, broadcast their location information periodically (in the active state). Otherwise, they switch to semi-active state.

Sensor node applications are highly vulnerable to the environmental conditions due to the wireless communication medium, special harsh conditions of the deployment terrains of the sensor nodes and their limited processing capabilities. Therefore, implementation of an algorithm for WSNs with a real testbed becomes more important in order to get more realistic results likewise the testbed environment in our work.

The rest of the paper is organized as follows: Section II overviews the localization algorithms in WSNs. Section III describes the testbed employed. In Section IV, the suggested localization technique is described and the test results by using various topologies are given. In Section V, the energy efficiency of the employed WSN system is enhanced. Finally, Section VI concludes the paper.

II. LITERATURE SURVEY

Sensing an environmental stimuli is the main functionality of a WSN. The coupling location information is important for many WSN applications to process the sensed data. Therefore, the location accuracy is fundamental in many WSN applications. Energy and cost efficient localization techniques are important in WSN because of the sensor nodes' limited processing capabilities. The location of a sensor node can be obtained from the global positioning system (GPS) or can be embedded to the sensor node at the deployment phase. GPS is not an efficient choice for determining the location because of the cost and energy restrictions. Furthermore, GPS is not compatible with indoor environments. Instead of using GPS, sensor nodes can find their locations with the help of some other nodes which know about their location.

Measurement based localization techniques are based on ToA, TDoA, AoA and RSSI. ToA, TDoA and AoA have constraints imposed by hardware requirements. AoA needs directional antenna. ToA and TDoA require a slow signal transmission which can be estimated. Since RSSI does not need any additional hardware, it is widely used.

An RSSI-based localization algorithm using a single mobile anchor node is proposed in [4], in which authors employ un-located sensor nodes whose locations are determined by calculating beacon points according to beacon information taken from the mobile anchor node. Beacon points are chosen according to maximum RSSI points. It is claimed that the mobile node that is equipped with GPS, does not violate energy-constraint.

Non-measurement based localization techniques are centroid algorithm, DV-hop algorithm, amorphous algorithm, APIT algorithm, etc. The most known non-measurement based localization techniques are centroid algorithm and DV-hop algorithm. In centroid localization algorithm, the localization error is increased when number of anchor nodes are not sufficiently high or the deployment of the anchor nodes is irregular. Different centroid algorithms are proposed to improve location estimations. In [5], mobile nodes use the information of its cluster whose nodes are weighted according to received RSSI and weighted centroid localization (WCL) is executed. In [6], weighted centroid algorithm and RSSI value taken from unknown nodes for determining distances are combined and tested on 2D and 3D simulation environments. Location of unknown nodes are determined by anchor nodes which are located in the range of circle whose center is an unknown node. They weighted anchor nodes which are in the defined circle by 1, rest of the anchor nodes are weighted by 0. Then, the relationship between localization error and the number of anchor nodes used in localization process is experimented and also it is observed that localization error is highly affected by the distribution of anchor nodes.

Non-measurement localization techniques are cost and energy efficient but location error is much more than measurement localization techniques [5]. These algorithms usually do not try to optimize or minimize the location estimation errors while producing a reasonable estimation with a very low algorithm complexity. Therefore, the localization error is higher in these techniques compared to localization techniques using distance measurements. We propose combining measurement and non-measurement based localization techniques to get lower localization error. Additionally, dependency of localization error with respect to the number of anchor nodes and the deployment of anchor nodes are investigated. Localization error, as in [6], is tried to be decreased by using a determined percentage of anchor nodes in localization process.

III. TESTBED EMPLOYED

In this work, we implement a WSN environment at the Computer Network Research Laboratory of Istanbul Technical University. Sensor nodes, which are used in our work, consist of two modules: sensor-L and node-RF. Sensor-L module contains sensors and node-RF module has processor and RF communication units [7]. RF antennas, LEDs, EEPROM, power inputs for battery, 41 pin socket which is helpful for parallel and serial port connections are placed on the node-RF module of the sensor node. Accelerometer, temperature sensor, light sensor, magnetic field sensor are placed on the sensor-L part of the sensor node. These sensor nodes employ TinyOS operating system. Programs for these sensor nodes are written in NesC programming language which is compatible with TinyOS operating system [8, 9]. In the sensor network system, anchor nodes are deployed according to a predefined pattern. They are used to route data and also the necessary information to localize the mobile nodes in the environment.

Distance Vector Routing algorithm (DVR) is employed as the routing algorithm . Count-to-infinity problem is solved by defining a timer. If a node does not send any packet before the timer expires, it is considered as offline [10].

Figure 1. Node RF card and battery

ISBN: 978-1-61208-239-4

A. *Types of Sensor Nodes*

Anchor nodes- They have static locations, send routing packets to the neighboring anchor nodes and mobile nodes. Furthermore, they send localization packets which contain x-y coordinates periodically. Mobile nodes use their localization packets for localization. Additionally, the anchor nodes route data packets to the sink node.

Mobile nodes- They move in the WSN. They determine their location using the packets sent by the neighboring anchor nodes. These nodes send their localization calculations to sink node periodically.

Sink node- It is connected to computer with a serial port. It sends the obtained data packets to the computer.

B. *Types of Packets*

In order to manage the system and transmit data, the following packet types are employed.

Routing Packets- Routing packets are sent by all of the nodes except the sink node. They are used to form the routing tables according to DVR. This routing packets include node id, next hops and hop counts to other nodes.

Localization Packets- Localization packets are sent by anchor nodes periodically. They include node id, X and Y coordinates.

Mobile-to-sink Packets- Mobile-to-sink packets are sent by mobile nodes to report their location to sink node. These packets are routed to the sink by using DVR routing tables. These packets contain node id, source node id, destination node id (sink node id), X and Y coordinates.

Data Packets- These packets are formed by the nodes in order to transmit the information related to the application to the sink node.

The wireless sensor network environment employed is given in Fig. 2.

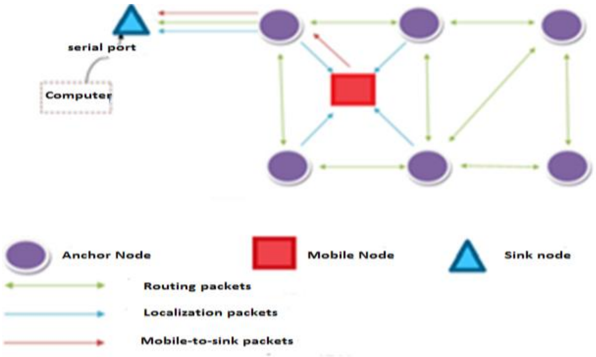


Figure 2. Wireless Sensor Network Structure

IV. LOCALIZATION TECHNIQUE INTRODUCED

Simulation results and testbed results are obtained.

A. *Enhancing The Precision Of Localization Estimates*

In this work, initially we study the effect of the number of neighboring anchor nodes used in the precision of the location estimates by simulation. Simulations are done using various topologies to find the necessary number of anchor nodes. One-hop distance is taken as 55 meters in the simulations. The distance between anchor nodes are changed between 20-50 meters. The anchor nodes are regularly placed as shown in Fig. 3.

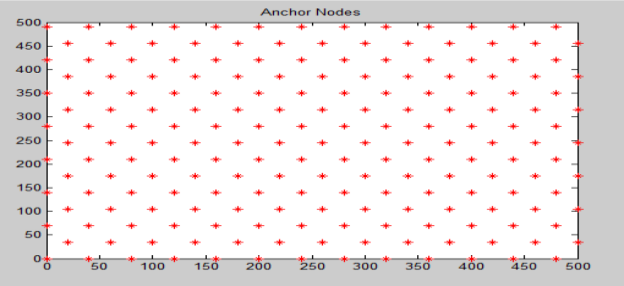


Figure 3. Anchor Nodes Placement in a 500*500m^2 region

Mobile nodes move randomly. Simulations are based on three random paths with 100 discrete movements of the mobile node. Mobile nodes move in one of four directions (+x, -x, +y, -y) in time unit by random distances (between 15m and 45m). A random 100-step movement scenario is given in Fig. 4.

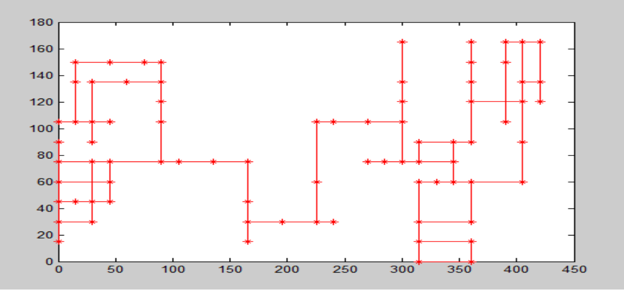


Figure 4. Random movement of mobile node

Simulations are repeated by changing the distance between anchor nodes. Average number of the anchor nodes, which are one-hop distance to mobile node, is given for different cases in Table 1.

TABLE I. CASES

Case No	Distance Between Anchors	Average number of neighboring anchor node
1	20m	≈23
2	30m	≈10
3	40m	≈5
4	50m	≈4

Above mentioned simulation scenarios are employed for location estimations by using the connectivity-based localization [3] first.

Then we modified the localization technique by reducing the number of anchor nodes employed. Initially we dropped one anchor node, then two anchor nodes and so on. The

ISBN: 978-1-61208-239-4

ones dropped selected according to the RSSI measurements. So the anchor nodes, which are far away, are not considered in the location estimates.

It is shown in Fig. 6 that as the number of neighboring anchor nodes increases the error decreases. As we drop the anchor nodes that are far away from the mobile node. The results become more accurate except case 4, where the average number of neighboring anchor nodes is four only.

We may conclude that nearly two third of the neighboring far anchor nodes could be omitted with the concern that the number of considered anchor nodes should not be smaller than three.

B. Testbed Results Obtained

The localization technique is implemented on the wireless sensor networks using 7 and 17 nodes. Considering the number of neighboring anchor nodes and the directions given in Section III, three anchor positions are used for position estimations.

Strength field in TOS_Msg struct of CC2420 module is used for RSSI measurement as below:

RSSI value (dBm) = (int8_t) TOS_Msg->strenght-45 (2)

Moreover, movement direction of mobile node is tried to be estimated. This is determined basically using the difference between previous two estimations of the mobile node.

The snapshot of the environment for 17-node case is shown in Fig. 5. Here, every anchor node is located 30 cm far from each other. One hop is approximately 40 cm when output power ratio module is taken as -25dBm.

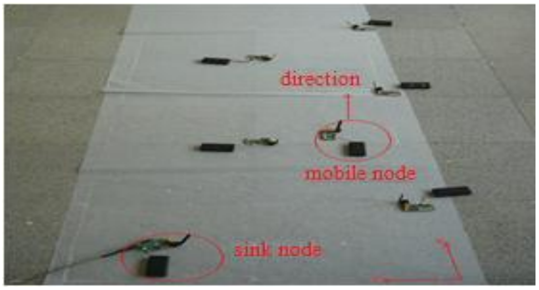
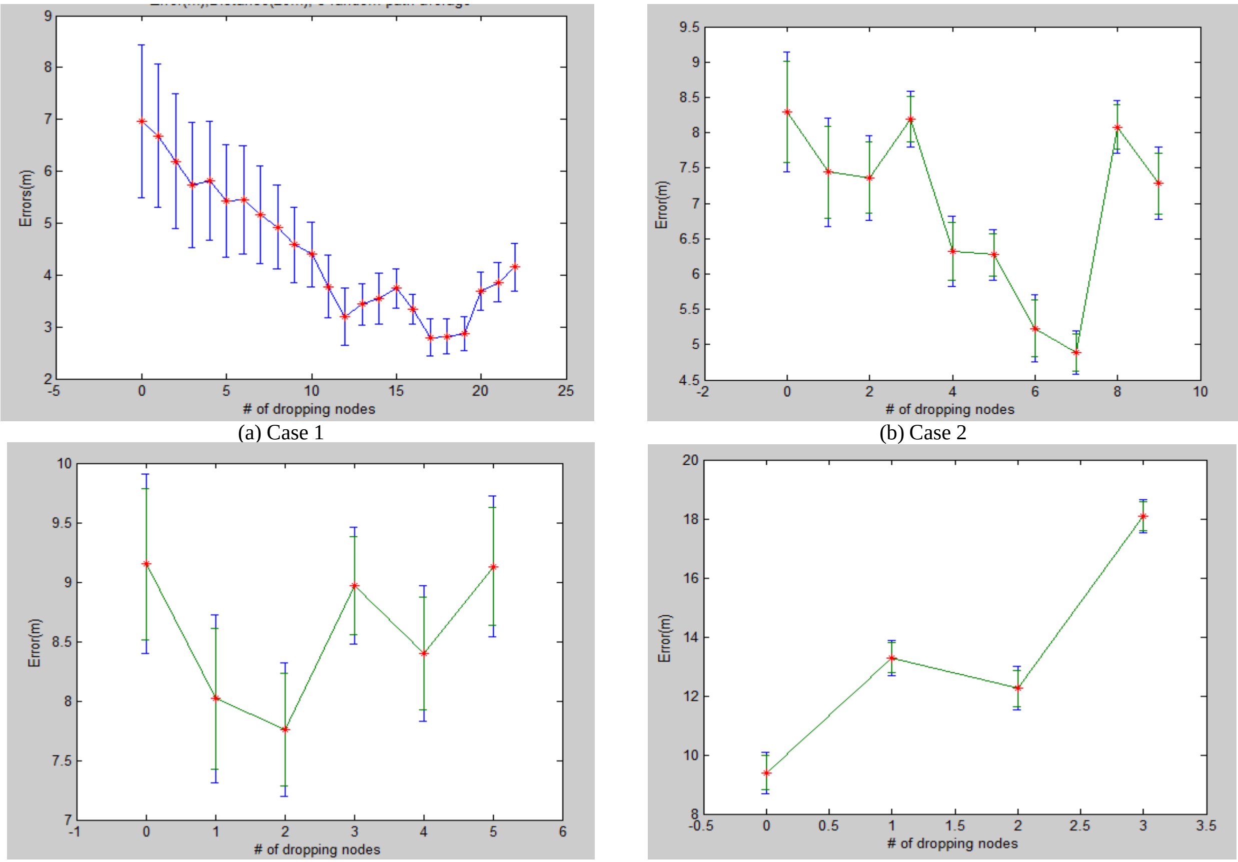


Figure 5. Test topology with 5 anchor nodes, 1 mobile node and 1 sink nodes

The results for the x-y coordinate estimates for 7-node and 17-node topologies are given in Fig. 7 and Fig. 8, respectively.



(a) Case 1

(b) Case 2

(c) Case 3

(d) Case 4

Figure 6. Error values for the cases studied

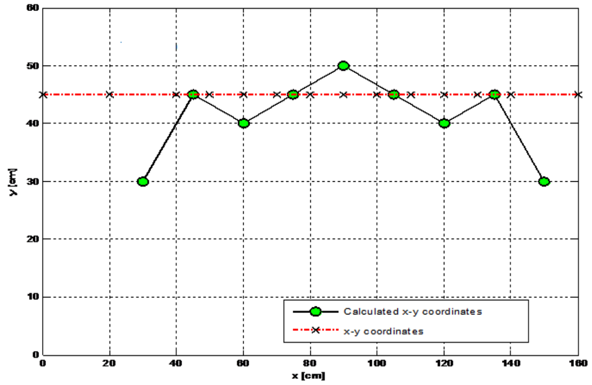


Figure 7. Estimated x-y coordinates and real x-y coordinates of mobile node (7 nodes)

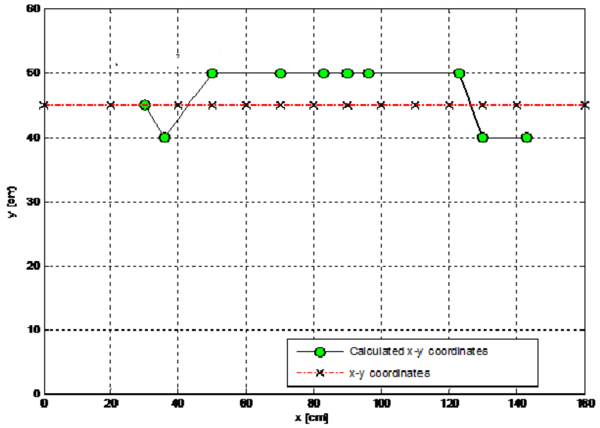


Figure 8. Estimated x-y coordinates and real x-y coordinates of mobile node (17 nodes)

Average numbers of neighboring anchor nodes are 3-4 in the 7-node topology and 7-9 in the 17-node topology. We get the best localization estimations when nearly two third of the neighboring far anchor nodes are omitted with the concern that the number of considered anchor nodes should not be smaller than three as explained in the previous sections. In the 7-node topology, the location estimations of the mobile node is calculated by using minimum number of anchor nodes (three anchor nodes). In the 17-node topology, two third of the neighboring nodes are omitted. Here, the location estimations of the mobile node is calculated by using three anchor nodes also. The 17-node topology gives better estimations as expected.

Environmental effects such as collisions and wifi networks disturb the efficiency of the results. However, despite of environmental conditions, our main assumptions are supported by the testbed results.

V. ENHANCING THE ENERGY EFFICIENCY OF THE NETWORK DEPLOYED

Energy efficiency is an important necessity for wireless sensor networks which have to work with energy constraint.

In suggested localization technique, packets which carry localization information are sent by anchor nodes periodically. However, the mobile node can only get such packets from anchor nodes in its coverage area. Therefore, localization packets should not be transmitted by anchor nodes, if anchor nodes are not in the communication range of a mobile node. We call this as *Energy Efficient WSN*. Here, the anchor nodes work in two states: *semi-active state*, and *active state*. The anchor nodes having a mobile node around broadcast their location information periodically (in the active state). Otherwise they go into semi-active state in which anchor nodes only send routing/data packets as necessary.

Anchor nodes keep IDs of all mobile nodes in the topology. It is not memory-consuming process even in large topologies. If an anchor node receives a routing package from a mobile node, it moves into the active state. Furthermore, a timer is set and starts to count. When the timer expires, the anchor node changes its state from active to semi-active.

Energy gain of the energy efficient WSN is determined by studying the decrease in the number of localization packets generated by the anchor nodes. Firstly, the power consumption to send localization packets is calculated as below [1].

$$P_{ct}= N_T\, [P_T\, (T_{on}+T_{st})+P_{out}T_{on}] \qquad (3)$$

where P_T denotes power consumption of CC2420 radio module when it is on transmission mode. This value is 62.64 mWatt for CC2420 when output power is 0 dBm. T_{st} is start-up time of CC2420 which is given as 1 ms. P_{out} is output power of CC2420 which is taken as 0 dBm. T_{on} is time duration which CC2420 stays on transmission mode. It can be calculated as L/R, where L denotes packet size and R data rate which are 20 byte and 250 kbps, respectively. N_T denotes number of activation of CC2420 radio module as transmitter. When these values are placed in (3), P_{ct} for 1 minute test duration becomes as seen in (4) where $N_T(5t)$ number of activation of CC2420 radio for transmission of localization packets in 5 seconds. The number of activations is evaluated from number of localization packets sent to the collector node in every 5 seconds during 1 minute test duration.

$$P_{ct}(5t) = \sum_{n=0}^{11} N_t\,(5t) \times [\left(62.64mW \times \left(\left(\frac{20byte}{250kbps}\right) + 0.001s\right)\right) + \left(1mW \times \left(\frac{20byte}{250kbps}\right)\right)] \qquad (4)$$

ISBN: 978-1-61208-239-4

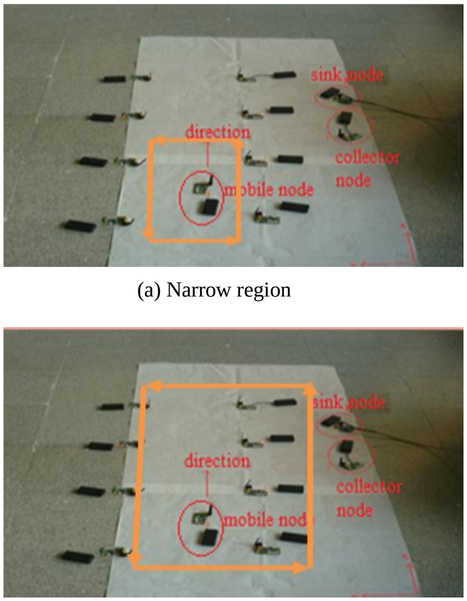
Power consumption during transmission of localization packets according to (4) is observed on two different test scenarios called Energy Efficient Localization in narrow region and wide region. Narrow region and wide region can be seen in Fig. 9. In the wide region scenario, the mobile node could move in overall region. In the narrow region scenario, the mobile node could move only a part of the region.

We run the introduced localization technique on the sensor nodes firstly. Then, the energy efficient WSN with the localization technique introduced is run on the sensor nodes by using both the wide area scenario and narrow area scenario. Power consumption considering transmission of localization packets is shown in Fig.10 for 60 second period. Energy Efficient WSN system uses 15 mWatt less power than the suggested localization technique during 1 minute test duration. Moreover, energy gain increases to 20 mWatt if the mobile node is moved in a narrow area. Because number of anchor nodes that send localization packets decrease in third test case.

To sum up these results, using mobile nodes in a narrow area gives more energy efficient results. Using more mobile nodes in dedicated areas gives more energy efficient results.

VI. CONCLUSION

In this work, a range-free one-hop localization technique is improved by using the location information obtained from the closer neighboring anchor nodes. RSSI measurements are taken to determine the distance of the anchor nodes to a mobile node. An approach is introduced in order to use the location information coming from only some of the neighboring anchor nodes. Tests are done for various topologies. The system introduced is turned into a more energy efficient one by assigning two states to the anchor nodes: active state and semi-active state. Due to the environmental effects the test environment scaled down.



(a) Narrow region

(b) Wide region

Figure 9. Test topology with 8 anchor nodes, 1 mobile node, 1 collector node and 1 sink node

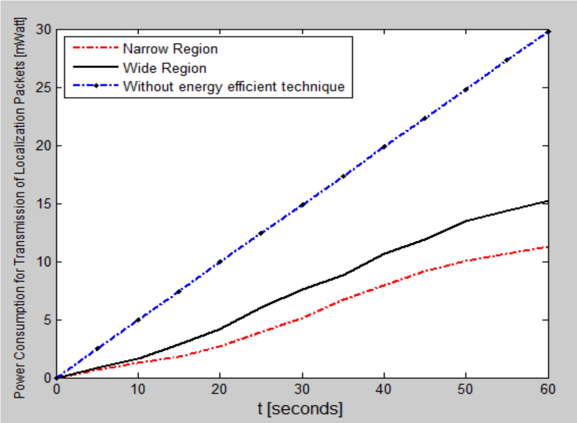


Figure 10. Power consumption comparison

Currently, we are operating the network by using 50 meter as the communication range of the sensor networks. Unfortunately, environmental effects are accumulating and disturbing the results.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey", Computer Networks, vol. 38, Januarv 2002, pp. 393-422, doi: 10.1016/S1389-1286(01)00302-4.

[2] G. Mao, B. Fidan, and B. D. O. Anderson, “Wireless Sensor Network Localization Techniques”, Computer Networks, vol. 51, Julv 2007, pp. 2529-2553, doi: 10.1016/j.comnet.2006.11.018.

[3] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less Low-Cost Outdoor Localization for Very Small Devices”, IEEE Personal Communications, vol.7, October 2000, pp. 28-34, doi: 10.1109/98.878533.

[4] Y. Zhu, B. Zhang, F. Yu, and S. Ning, “A RSSI Based Localization Algorithm Using a Mobile Anchor Node for Wireless Sensor Networks”, in: Computational Science and Optimization, CSO 2009, April 2009, pp. 123-126, doi: 10.1109/CSO.2009.287.

[5] H. Ahn and S. Rhee, "Simulation of a RSSI-Based Indoor Localization System Using Wireless Sensor Network", in: Ubiquitous Information Technologies and Applications (CUTE), December 2010, pp. 1-4, doi: 10.1109/ICUT.2010.5678179.

[6] L. Xu, K. Wang, Y. Jiang, F. Yang, Y. Du, and Q. Li, “A Study on 2D and 3D Weighted Centroid Localization Algorithm in Wireless Sensor Networks”, in: Advanced Computer Control (ICACC), January 2011, pp. 155-159, doi: 10.1109/ICACC.2011.6016388.

[7] Genetlab Information Technology, "Sensenode –low power wireless sensor module–datasheet", December 2005. Retrieved June 2012, from http://www.genetlab.com/images/urunler/SENSENODE.pdf

[8] P.Levis and D. Gay, "TinyOs Programming", July 2009. Retrieved June 2012, from http://csl.stanford.edu/~pal/pubs/tos-programming-web.pdf

[9] C. F. Fok, "TinyOs Tutorial", 2006. Retrieved June 2012, from http://web.eecs.utk.edu/~xwang/ece455/tutorial.pdf

[10] G. Meyer and S. Sherry, "Triggered Extensionsto RIP to Support Demand Circuits", RFC 2091, IETF Network Working Group, January 1997.

A Context-Relational Approach for the Internet of Things

Jamie Walters*†, Theo Kanter†
*Department of Information Technology and Media
Mid Sweden University, Sundsvall, Sweden
email: jamie.walters@miun.se
†Department of Computer and System Science
Stockholm University, Forum 100 Kista, Sweden
email: {kanter, jamiew}@dsv.su.se

***Abstract*—Context-centric applications and services are premised on the ability to readily respond to changes in context. Centralized approaches to enabling this are undermined by their dependencies on DNS naming services while decentralized approaches using DHT variants have been centred on the provisioning of the underlying context information, creating information-centric rather than context-centric solutions. A dynamic Internet of Things mandates a new paradigm; approaches storing, discovering and associating context entities relevant to their context state. In this paper, we explore such a paradigm, and with the implementation of a prototype, show the advantages of moving towards the notion of context-state centricity on the Internet of Things.**

Keywords-context awareness; context; context models; Internet of things; context proximity; sensor information; p2p context

I. Introduction

Current trends in computing bring the paradigm of pervasive and ubiquitous computing into focus. Users are now even more connected; demanding a range of everything everywhere services. These services, including as social networking and media, benefit from the availability of context information seamlessly gathered and shared; providing customized and user-centric experiences.

Dey[1] contributed significantly to the understanding of this context centric paradigm and its central role in advancing ubiquitous and pervasive computing research. The two definitions introduced in [1], remain concrete definitions; pillars of modern context aware computing research. Defining context as:

> *"any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves"*

and context awareness as:

> *"a system is context aware if it uses context to provide relevant information and or services to a user, where relevancy depends on the user's task."*

Research towards this realization of context awareness has largely been focused on the ability to construct an accurate and timely representations of a user's state from the information gleaned from an intricately woven infrastructure of sensor and actuators.

Such an interconnected things infrastructure is expected to have an installed device base in the range of several billion [2] and will be capable of supporting a diverse set of experiences ranging from personalized and seamless media access, to intelligent commuting or environmental monitoring services. Dubbed the *Internet of Things*, it will incorporate devices such as electronics, vehicles, mobile telephones and even municipal infrastructures and the people themselves, merging towards the paradigm of everywhere computing [3].

This underpinning Internet of Things (IoT) is a key enabling factor in the creation and deployment of applications and services in response to the situation of a user and his current relationship with his environment, applications and services. This, according to Dey [1] constitutes the working definition of context and cements its central role in the explosion of pervasive and ubiquitous applications and services.

Approaches towards the realisation of such an Internet of Things have largely been focused on the design and implementation of systems that are capable of provisioning context information with acceptable degrees of availability, accuracy and reliability. However, our ability to discover related entities are limited by the arbitrary storage mechanisms such as the DHT approaches used by Kanter et. al. in [4] and Baloch et. al. in [5]. This resulted in bottom up approaches to finding related context entities through the use of costly searches over their constituent context information. Additionally, indexing and caching approaches such as [6] cannot offer guarantees in freshness of information as is required to drive real-time applications.

This in turn mandates newer approaches to the storage and retrieval of context entities in order provide support for the wide array of context centric applications and services that will occupy an Internet of Things. One such approach is the storage and discovery of entities as a factor of their overall context relationships. In this approach, we seek to persist and discover related context entities solely over their context relations and verify that we are capable of retrieving these entities with the same level of accuracy while negating the need for applications and services to compose multiple queries over composite context information.

Further, we explore the advantages of context-relational queries with respects to publish - subscribed based approaches. We demonstrate that relational discovery permits us to reduce the subscription related communication and computational overheads by changing the dynamics used to create and maintain subscriptions.

While we show that solutions can be created for simplifying the discovery of related entities within a region of interest or a degree of relationship, defining the rules for quantifying this relationship is a non-trivial problem. This requires further investigation into approaches for deriving and representing degrees of relationships or context proximity amongst entities in order to create the underlying relational context networks for driving future applications and services.

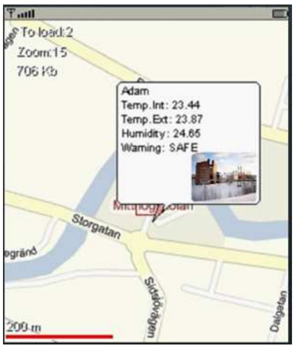
The remainder of the paper is structured as follows: Section II looks at the background work and motivation, Section IV outlines the approach. Section V presents our verification and results while Section VI summarises the conclusions and future work.

II. Background and Motivation

The early context provisioning architectures that offer support for an Internet of Things varied significantly with respects to both implementation and approach. They however converge on fulfilling a set of fundamental requirements capable of enabling access to the information required for driving the exploration of ubiquitous applications and services.

DNS Independence was explored in MediaSense [4] and SCOPE [5] through the use of Distributed Hash Table (DHT) type overlays such as [7]. This moved context provisioning architectures away from centralized approaches such as SenseWeb [8] and IP Multimedia Subsytem (IMS) [9], which are dependent on Domain Name Service (DNS) as a means of locating service portals, users and applications.

Real Time Provisioning using Distributed Hash Table (DHT) based architectures such as [4] and [5] enabled the resolution of context information within times that were comparable with User Datagram Protocol (UDP) and deemed adequate enough to support real-time context dependent services.



To load:2
Zoom:15
706 Kb

Adam
Temp.Int: 23.44
Temp.Ext: 23.87
Humidity: 24.65
Warning: SAFE

Figure 1. A Simple Mobile Context Awareness Application

Self-Organization of constituent context information was explored in by Walters . et. al in [10] verifying that we are capable creating solutions that retain the advantages of DHTs while introducing more dynamic self-organization qualities.

The problem, however, is that architectures that seek to represent the context interactions supporting the real world must be able to represent context entities over their higher level context states and the degrees or relational affinity among these states. This would serve to extend early solutions such as those employing DHTs to effectively create context-state centric networks.

Early solutions enabled the provisioning of context information concerning users and their environments. As such, a user finds it relatively simply to understand his current state. Such a user will readily comprehend: *It is five degrees Celsius and I am walking 2 km/h on the High Street.* Additionally, this information is made available to applications and services interested in the user's state of being.

Such approaches permitted the creation of simpler applications such as [11], shown in Figure 1, which permits users to locate other users based on their context.

However, context centric applications and services are becoming more collaborative requiring approaches to consider not only localized user context but rather a user's relationship within a dynamic context centric network.

Schilit [12] regarded context as being composed of three key aspects used to define an entity's situation: *"where you are, who you are with[near], and what resources are nearby"*. With further evolution into *Presentities* [13], this can be refined as: *a presentity, its related presentities and the degree of their relationship.*

Here, the complex context networks that underpin the pervasive computing paradigm exists as more general collections of *presentity-relationship-presentity* triples shown in Figure 2, where the relationship is the degree of affinity between the current context states of the presentities. Enabling us to create context networks where organization is achieved at a higher level. We used the terms presentity and context entity interchangeably in this paper.

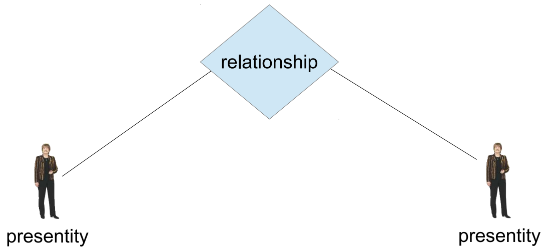


Figure 2. The Presentity Triple

While semantic approaches to triples provide a means of characterizing the relationships between entities, it obtains limited expressiveness with respect to a measurement of affinity. Here, models that support a metric over these relationships are, according to Schmohl and Baumgarten [14], complementary in characterizing the types of relationship illustrated in Figure 2.

III. Space Filling Curves

Peano first proposed the idea of a space filling curve as a finite curve which begins at the origin and traverse every point in an n-dimensional hypercube [15]. Several versions of space-filling curves have been derived with varying properties. Our approach regards each context state as corresponding to a point in n-dimensional space traversed by a space-filling curve. Thus, of particular interest are space filling curves having superior locality preservation. That is, the order in which the curve visits a point highly correlates with their observable proximity within the n-dimensional space, and subsequently their distance of separation on the curve.

Additionally, the heterogeneity of context information required a variant that was capable of handling dimensions measured on different scales without the need for padding. The Hilbert Space Filling Curve obtains superior locality preservation [16] and for our implementation we selected an extension, the Compact Hilbert Space Filling Curve [17], which is capable of generating Hilbert Indices for n-dimensional hypercubes where sides are of different scales.

IV. The Approach

Our approach is not focused on the creation of a new context model, but rather the organization of entities represented by existing context models. To this end, we adopt the context models described in [18] and [19] as outlined below. In response to the shortfalls discussed in Section II, we are required to organize and represent presentities and their context at a higher order, permitting us to reason over the state relations of presentities rather than their underlying context information.

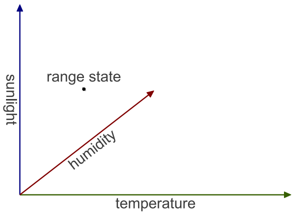


Figure 3. Weather Range

We define Context Networks as collections of *presentity triples*, where a presentity through its context information, possesses a specific state with a degree of relationship to other states within the context network. To enable this organization, each context state occupied by a presentity is indexed using a space-filling curve, distributing these values on a modified DHT overlay to preserve order and degrees of relationship among presentities. While this approach is partially explored in [20], its focus on geographical information does not provide a sufficient response to the requirements of Schmidt et. al. in [21]. We further implement a range query algorithm for querying presentities over current context states as opposed to raw underlying context information.

Our key point of departure from existing context aware implementations is therefore that we store, retrieve and query presentities relative to their state as opposed to their underlying context information

A. Presentity

We regard a presentity [13] to be all entities which, at any point in time, possess presence and context information that describes its current state. A presentity may contain any combination of context information and be continuously updated and retrieved. The distance or similarity between the states of presentities determine their suitability for a context-aware application or service.

B. Context Range

We adopted the concept of*context ranges* as described by Schmohl and Baumgarten [18]. Here, a context range is a group of individually observable context information that contribute to describing a specific category of context. For example, the range *weather* could be defined over temperature, humidity and sunshine intensity. We model each range as an n-dimensional hypercube as shown in Figure 3, where each face of the cube corresponds to a dimension of context information.

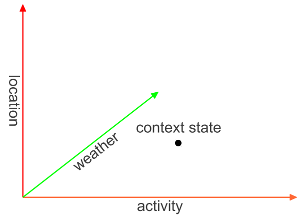


Figure 4. Application Space

C. Context State

The context state was defined by Padovitz [22] as the current situation of a context entity as defined by its current set of context information. A context state is therefore a point in an n-dimensional space whose position is defined by the values for each context dimension. A context state only exists for a specific time, t. We deviate from the Padovitz's [22] original definition and create two states; namely the *range state* as shown in Figure 3 and the *context state* as shown in Figure 4. A range state is the current or state of an entity within a particular range, e.g., the current state of the weather. Each state is a point in a hypercube where each face corresponds to an actual piece of context information.

The context state is the current state of an entity over a subset of its ranges, e.g. weather and location and time. This corresponds with the application space defined by Padovitz [22]; the n-dimensional space within which a state is valid with respects to the fulfilment of an application or service. As is currently attained, an application in this scenario would require a specific weather in a particular location at a given time. An application domain D could therefore be any combination of ranges R such that:

$$D \leftarrow \{R_1, R_2, R_3, ...R_i\}$$

And a context state, C, within this domain being expressed over all the range states, r of a presentity such that:

$$C \leftarrow \{r_1, r_2, r_3, ...r_k\}$$

Application spaces are, therefore, hypercubes with the difference being that each dimension being a range, see Figure 4

D. General Persistence

In order to support our approach, we created two DHT overlays namely, a General Persistence and a State Persistence. The general persistence overlay is an extension of our previous work towards a context provisioning architecture and uses a DHT overlay for persistence. When a presentity is introduced into our solution, its constituent context information is decomposed and persisted in the general persistence overlay, given an identifier, a URI of the format:

$$dcxp://user@domain/contextdimension$$

as described in by Kanter, et.al. [4]. For this, we used a Pastry DHT with SHA-1 as the underlying hashing algorithm, making no significant modifications. We selected a URI of this format in keeping with our previous work, however any chosen unique identifier would provide sufficient support.

E. State Persistence

In an effort to improve the performance for finding related presentities, we created a State Persistence component. This is a modified DHT used for persisting the current context state of a presentity derived from a calculation of its Hilbert Index [17]. The Hilbert Index we derived for each context state is of datatype *long*. We therefore modified the DHT overlay to created a distributed *long* overlay, effectively replacing the SHA-1 [23] hashing function in the DHT and modifying the overlay construction functions to build an ordered overlay which stores long values in a 96-bit distributed space. The resulting being that the space is organized in a clockwise order-preserving manner.

F. Indexing Presentities

The main aim of the solution is to index presentities over their current context state and permit applications and services to locate these presentities with respects to the affinity between current states of context. In achieving this, we first index all presentities within the general persistence component including all composite context information and the presentity identifier. Secondly, each context state is indexed.

In order to achieve this, we model each range described in Section IV-B as an n-dimensional context space containing a Compact Hilbert Space filling curve. Figure 5 shows a such a Hilbert Curve traversing a 2-dimensional range. The resulting is an order on all the possible states within a range as points on a line, where the distance between the points are indicative of the similarity between the states.

Indexing the context state of an entity is achieved using the same method, with the exception that each presentity is indexed over its constituent range indices, i.e., an index of indices.

The locality preserving property of the Hilbert Curve ensures that presentities with similar context values are assigned similar Hilbert Indices and their states subsequently are indexed within close proximity.

G. Range Search

This results in two types of index values for each presentity, namely range indices and state (application)

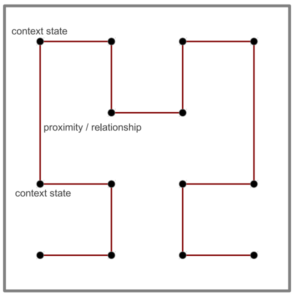


Figure 5. Context State Indexing

indices. We persist these indices in the state persistence layer by extending them to add additional information about the type of index being persisted. The first 16-bits describe the type of index, i.e., context state of range state. The consecutive 16-bit represented the range being index while the final 64-bits contains the index value for the range or context state.

The consequence of this being that index information is successively partitioned by type, range and by value. Replication is employed on the state persistence overlay in order to negate any issues that may arise with this partitioning. Additionally, the distance a presentity is stored from a another presentity is a factor of the context similarity or distance they between them; presentities with a similar expression of context are stored close together on a DHT.

Searching for presentities within the solution is achieved from a context state perspective. Applications typically search for presentities that, to a certain degree, possess similar context states. By utilizing a space filling curve, we created an order on the presentities as a factor of their states. Consequently, the distance between any two points on this curve is a factor of the distances between their collective underlying context information.

We exploit this property in creating range queries for locating groups of presentities that are within a specified area on the curve and therefore within a specified area of the n-dimensional context spaces and subsequently with similar context information. An application wishing to locate an entity, first decides on the maximum and minimum values for each dimension and with these values, calculate the maximum and the minimum Hilbert Indices, indicating the maximum distances to be traversed on the space filling curve. We then construct a single query which is sent to the nodes responsible for upper bound, the lower bound and median value in the range. Each node, on receiving the query forwards it in either a clockwise or an anti-clockwise direction around the DHT depending on its position in the range. The query is not forwarded beyond the nodes located at either ends of the query range, with the results returned to the querying node. With this approach, we can query entities based on their context states and the degrees of affinity amongst these states, with limited knowledge of the vast and heterogeneous array of underlying context information.

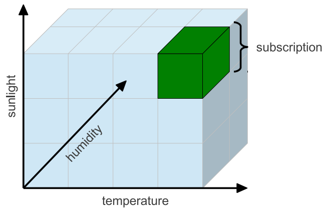


Figure 6. Range Subscription

H. Range Subscription

The search functionality described in the previous section is extended to support subscriptions. The subscription functionality moves away from previous approaches in context scenarios where a subscription was made to an entity. Here, a subscription is made to a region of interest as show in Figure 6, which is defined by the upper and lower bounds of the range query. The algorithm functions almost identically to the searching, with the exception that along with retuning the presentities matching the search query, a subscription is constructed where where any entity whose changes in context state effects a change in its position within this region of interest is sent to the subscribing/querying node.

V. Results

We simulated our approach using the Free Pastry variant of the Pastry DHT [24]. The Java-based API contains a simulation environment permitting us to create the numbers of nodes required to test our implementation. The lack of sufficient quantities of context information was addressed by using the constituent values of the RGB and CMYK colour scales as context information with each scale being considered a *context range.*

For the verification of the range query, the RGB scale was used as a range with each colour representing a range state. We created, indexed and persisted a number presentities with these states. A random colour was chosen and range query performed between an upper and lower colour. A range query was executed and by querying for range of colours from a given shades of colours from a given point regions of the RGB scale we compared expected result sets. The results are shown in Table I.

We verified the advantages of organizing context entities over their states as well as the ordered persistence of these states advantages of context state by observing the number of messages required to resolve a query. As

# Presentity States	# Queries	Accuracy
2000	250	98%
2000	2000	99%
100	250	100%

Table I
Range Query Verification

queries are not randomly flooded but rather propagated around the DHTs ring like structure, the number of messages required should increase as as a function of the range increase. For this, we created ranges as described earlier and executed searches gradually increasing the range with each iteration.

We observed an increase in the number of messages required to resolve a query as the range of the query increased. As queries are propagated exponentially with respects to range along the search path, the resulting graph shown in Figure 7 verifies the ordering of the context entities as a factor of the affinities of their underlying context states. A random distribution would have produce random numbers of query messages.

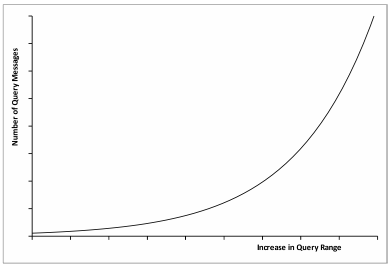


Figure 7. Range / Message Increases

The range subscription removed the need to subscribe to entities directly or issues relating to new entities arising after a search has been completed. Context provisioning approaches such as the one described by Kanter et. al. [4] maintain information on all subscriber - publisher relationships. Maintaining subscriptions for an area of interest N, with presentities P, each with S number of subscribers would maintain

$$\sum_{P \in N} (P \cdot S_P)$$

subscriptions. However, in our implementation, we are only interested in the number of subscribers to the area resulting in the number of subscriptions being equal to the number of subscribers, S.

VI. Conclusion & Discussion

Ubiquitous applications require access to relevant context entities in order to accurately provision the services demanded by end users. Approaches to provisioning context information have not satisfactorily addressed the need to model context entities with respect to their relations, enabling a more natural way to resource discovery.

In this paper, we presented an approach for indexing context entities relative to their current context states and exploiting this indexing information for searching and subscribing. Context information is categorised into context ranges and each presentity is indexed both within the context range and within an application space; a collection of context ranges. Indexing is achieved by modelling each range as a *n-dimensional* hypercube and calculating the Compact Hilbert Index for each state. This is persisted on an order preserving overlay, in order to maintain the locality achieved by the space filling curve. With this, we are able to query a context network relative to the affinity observed between any two entities using a range query algorithm.

We have shown that this approach is capable of finding relevant entities based on their context higher level context relationships and with respects to the requirements of an application or service. We implemented an approach that permits subscription to an interest area relative to a context state and reduce the need to discover and subscribe to all entities within a context network or to perform continuous searches.

Our solution however, is dependent on the Hilbert Curve as the means of realising our context relational model. While this permits us to motivate research into this area, applications and services will require more intuitive relational search and discovery. Future work therefore mandates more dynamic and user driven definitions of context relations, measures of proximity or affinity and efficient algorithms for constructing dynamic context-relational networks.

References

[1] A. K. Dey, “Understanding and Using Context,” *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7, Feb. 2001.

[2] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, “Vision and Challenges for Realising the Internet of Things,” in *Cluster of European Research Projects on the Internet of Things (CERP-IoT)*, no. March, 2010, pp. 43–67.

[3] J. Lee, J. Song, H. Kim, J. Choi, and M. Yun, “A User-Centered Approach for Ubiquitous Service Evaluation: An Evaluation Metrics Focused on Human-System Interaction Capability,” in *Computer-Human Interaction*, ser. Lecture Notes in Computer Science, S. Lee, H. Choo, S. Ha, and I. C. Shin, Eds., vol. 5068. Berlin, Heidelberg: Springer, 2008, pp. 21–29.

[4] T. Kanter, S. Pettersson, S. Forsstrom, V. Kardeby, R. Norling, J. Walters, and P. Osterberg, “Distributed context support for ubiquitous mobile awareness services,” in *2009 Fourth International Conference on Communications and Networking in China*. IEEE, Aug. 2009, pp. 1–5.

[5] R. Baloch and N. Crespi, “Addressing context dependency using profile context in overlay networks,” in *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*. IEEE, Jan. 2010, pp. 1–5.

[6] Y. Zhu, S. Ye, and X. Li, "Distributed PageRank computation based on iterative aggregation-disaggregation methods," in *Proceedings of the 14th ACM international conference on Information and knowledge management*. New York, New York, USA: ACM, 2005, pp. 578–585.

[7] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 1, pp. 41–53, Jan. 2004.

[8] A. Kansal, S. Nath, J. Liu, and F. Zhao, "Senseweb: An infrastructure for shared sensing," *IEEE MultiMedia*, vol. 14, no. 4, pp. 8–13, Oct. 2007.

[9] C. Gonzalo, "3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular World," p. 381, 2005.

[10] J. Walters, T. Kanter, and E. Savioli, "A Distributed Framework for Organizing an Internet of Things," in *The 3rd International ICST Conference on Mobile Lightweight Wireless Systems*, Bilbao, 2011, pp. 1–17.

[11] T. Kanter, S. Pettersson, S. Forsström, V. Kardeby, and P. \\"Osterberg, "Ubiquitous mobile awareness from sensor networks," in *Mobile Wireless Middleware, Operating Systems, and Applications-Workshops*. Springer, 2009, pp. 147–150.

[12] B. Schilit and N. Adams, "Context-aware computing applications," *Systems and Applications, 1994.*, pp. 85–90, 1994.

[13] H. Christein, *Distributed Communities on the Web*, ser. Lecture Notes in Computer Science, J. Plaice, P. G. Kropf, P. Schulthess, and J. Slonim, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, Dec. 2002, vol. 2468.

[14] R. Schmohl, "The Contextual Map," *deposit.ddb.de*, 2010.

[15] G. Peano, "Sur une courbe, qui remplit toute une aire plane," *Mathematische Annalen*, vol. 36, no. 1, pp. 157–160, Mar. 1890.

[16] J. Alber and R. Niedermeier, "On Multidimensional Curves with Hilbert Property," *Theory of Computing Systems*, vol. 33, no. 4, pp. 295–312, Jun. 2000.

[17] C. H. Hamilton and A. Rau-Chaplin, "Compact Hilbert indices: Space-filling curves for domains with unequal side lengths," *Information Processing Letters*, vol. 105, no. 5, pp. 155–163, Feb. 2008.

[18] R. Schmohl and U. Baumgarten, "The Contextual Map-A Context Model for Detecting Affinity between Contexts," *MobileWireless Middleware, Operating Systems, and Applications*, pp. 171–184, 2009.

[19] A. Padovitz, S. Loke, and A. Zaslavsky, "Towards a theory of context spaces," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, no. March. IEEE, 2004, pp. 38–42.

[20] M. Knoll, "A P2P-Framework for Context-based Information," *1st International Workshop on Requirements and*, 2006.

[21] A. Schmidt, M. Beigl, and H. Gellersen, "There is more to context than location," *Computers & Graphics*, vol. 23, no. 6, pp. 893–901, 1999.

[22] A. Padovitz, "Context Management and Reasoning about Situations in Pervasive Computing," Doctoral Thesis, Monash University, 2006.

[23] D. Eastlake 3rd and P. Jones, "US Secure Hash Algorithm 1 (SHA1)," 2001.

[24] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," *Design*, 2001.

A Quasi-random Multirate Loss Model supporting Elastic and Adaptive Traffic

Ioannis D. Moscholios
Dept. of Telecommunications
Science and Technology,
University of Peloponnese,
22 100, Tripoli, Greece
idm@uop.gr

John S. Vardakas, Michael D. Logothetis,
Michael N. Koukias
WCL, Dept. of Electrical & Computer Engineering,
University of Patras,
26 504, Patras, Greece
jvardakas@wcl.ee.upatras.gr
m-logo@wcl.ee.upatras.gr
koukias@wcl.ee.upatras.gr

***Abstract*— Nowadays, the telecom network traffic environment is composed mostly of emerging multirate services whose calls can tolerate bandwidth compression either by extending their service-time (elastic services) or not (adaptive services). The co-existence of elastic and adaptive services makes the call-level performance analysis and evaluation of modern networks much more complicated. To contribute, in this paper, we present a new multirate loss model for elastic and adaptive services with finite traffic-source population. Thus, the call arrival process is a quasi-random process which is, in many cases, a more realistic consideration than a random (Poisson) process. The proposed model does not have a product form solution, and therefore, we propose approximate but recursive formulas for the efficient calculation of the call-level performance metrics, such as time and call congestion probabilities and link utilization. The consistency and the accuracy of the new model are verified through simulation and found to be quite satisfactory.**

Keywords - quasi-random process; elastic-adaptive traffic; recursive formula; time-call congestion; Markov chain.

I. INTRODUCTION

In modern communication networks, the increase of elastic and adaptive multirate traffic necessitates the QoS network assessment through proper teletraffic loss models [1]. Based on them, we can accommodate various services in the network according to their offered traffic-load, and avoid costly over-dimensioning of the links. By the term elastic traffic (e.g., file transfer) we refer to calls that can compress their bandwidth, while simultaneously expanding their service time. On the other hand, adaptive traffic refers to calls that can tolerate bandwidth compression, but their service time does not alter (e.g., adaptive video) [2].

Call-level multirate loss models of a single link of fixed capacity, where calls of both elastic and adaptive service-classes are accommodated, have been proposed in [2]-[4]. In all cases, calls can tolerate bandwidth compression down to a minimum value. In [2], calls arrive in the link according to a Poisson process (a random process) and use their peak bandwidth requirement when the occupied link bandwidth does not exceed the capacity of the link. Otherwise, the link accepts a call by compressing its peak-bandwidth, as well as the bandwidth of all in-service calls, at the same time. Call blocking occurs when, after the maximum possible bandwidth compression, the minimum bandwidth requirement of a new call is still higher than the available link bandwidth. The minimum bandwidth requirement of a call is a proportion of its peak-bandwidth; this proportion is common to all service-classes. When an in-service call, whose bandwidth has been compressed, departs from the system, then the remaining in-service calls expand their bandwidth. Our analysis through Markov chains shows that this bandwidth compression/ expansion destroys the Markov chain reversibility, and therefore no Product Form Solution (PFS) exists. However, based on the method proposed in [2], we find an approximate but reversible Markov chain, which is solved and leads to a recursive formula for the determination of link occupancy distribution and, consequently, call blocking probabilities and link utilization. This formula resembles the well-known Kaufman-Roberts formula used in the Erlang Multirate Loss Model (EMLM), where Poisson arriving calls of different service-classes have fixed bandwidth requirements (stream traffic) [6],[7]; thus, we name the model of [2], Extended EMLM (E-EMLM). In [3], the E-EMLM is extended to include retrials, i.e., blocked calls may retry one or more times to be serviced with reduced bandwidth. In [4], new calls, upon their arrival, may reduce their bandwidth according to the occupied link bandwidth. In [5], the E-EMLM is further extended to include the Batched Poisson call arrival process which is used to approximate arrival processes that are more “peaked” and “bursty” than the Poisson process. Recently, a multirate loss model that includes stream, elastic and adaptive traffic has been proposed in [8]; the presence of stream traffic prohibits the recursive calculation of link occupancy distribution.

In this paper, we extend [2] by assuming that calls of each service-class (elastic or adaptive) come from finite sources. This arrival process is known as quasi-random and is smoother than the Poisson process [9]. The proposed model does not have a PFS. However, we propose an approximate recursive formula for the efficient calculation of the link occupancy distribution. This formula simplifies the determination of: a) Time Congestion (TC) probabilities, b) Call Congestion (CC) probabilities and c) link utilization. Applications of the proposed model are in the area of wireless networks, where calls come from finite sources and their bandwidth is compressed (e.g., [10]-[12]).

The remainder of this paper is as follows: In Section II, we: a) present the basic assumptions of the proposed model and the call admission control, b) prove the recursive formula for the link occupancy distribution and c) provide formulas for the various performance measures. In Section III, we provide numerical results whereby the new model is compared to the E-EMLM and evaluated through simulation results. We conclude in Section IV.

II. The Proposed Model

A. Basic assumptions and description of call admission

Consider a link of certain capacity C bandwidth units (b.u.) that accommodates elastic and adaptive calls of K different service-classes. Let K_e and K_a be the set of elastic and adaptive service-classes ($K_e + K_a = K$), respectively. Calls of service-class k (k=1,…,K) come from a finite source population N_k and request b_k b.u. (peak-bandwidth requirement). The mean arrival rate of service-class k idle sources is $\lambda_k = (N_k - n_k)v_k$ where n_k is the number of in-service calls and v_k is the arrival rate per idle source. This call arrival process is a quasi-random process [9]. If $N_k \to \infty$ for k=1,…,K then a Poisson process results. To introduce bandwidth compression, the occupied link bandwidth j may exceed C up to T b.u.

To describe call admission, consider the arrival of a service-class k call while the system is in state j. Then:

i) If $j + b_k \leq C$, the call is accepted in the system with its peak-bandwidth requirement for an exponentially distributed service time with mean μ_k^{-1}.

ii) If $j + b_k > T$, the call is blocked and lost.

iii) If $T \geq j + b_k > C$, the call is accepted in the system by compressing its peak-bandwidth requirement, as well as the assigned bandwidth of all in-service calls. All calls share the C b.u. in proportion to their peak-bandwidth requirement, while the link operates at its full capacity C. This is the processor sharing discipline [13].

When $T \geq j + b_k > C$, the compressed bandwidth b_k^{comp} of the newly accepted call, is given by:

$$b_k^{comp} = rb_k = \frac{C}{j'}b_k \tag{1}$$

where $r = C/j'$ denotes the compression factor, $j' = j + b_k$.

Since $j = \sum_{k=1}^{K} n_k b_k = \boldsymbol{nb}$, $\boldsymbol{n} = (n_1, n_2, ..., n_K)$ and $\boldsymbol{b} = (b_1, b_2, …, b_K)$, the values of r are given by $r \equiv r(\boldsymbol{n}) = C/(\boldsymbol{nb} + b_k)$. The bandwidth of all in-service calls is also compressed by the same factor $r(\boldsymbol{n})$ and becomes equal to $b_i^{comp} = \frac{C}{j'}b_i$ for i = 1,…,K. After bandwidth compression, we have $j = C$ and all adaptive calls do not alter their service time. On the other hand, all elastic calls increase their service time so that the product 'service time' by 'bandwidth' remains constant. The minimum bandwidth that a call of service-class k (k =1,…,K) tolerates, is:

$$b_{k,\min}^{comp} = r_{\min} b_k = \frac{C}{T} b_k \tag{2}$$

where $r_{\min} = C/T$ is the minimum proportion of the required peak-bandwidth and is common to all calls.

When an in-service call of service-class k, with bandwidth b_k^{comp}, departs from the system, then the remaining in-service calls of each service-class i (i=1,…,K), expand their bandwidth to b_i^{expan}, in proportion to their peak-bandwidth b_i:

$$b_i^{\exp an} = \min\left(b_i, b_i^{comp} + \frac{b_i}{\sum_{k=1}^{K} n_k b_k} b_k^{comp} \right) \tag{3}$$

B. Determination of link occupancy distribution and various performance metrics

Let $\boldsymbol{\Omega}$ be the system's state space $\boldsymbol{\Omega} = \{ \boldsymbol{n}: 0 \leq \boldsymbol{nb} \leq T \}$. Due to the bandwidth compression/expansion mechanism, we cannot describe the system by a reversible Markov chain (i.e., local balance does not exist between adjacent states of $\boldsymbol{\Omega}$). Therefore, the steady-state distribution $P(\boldsymbol{n})$ does not have a PFS. To derive an approximate but recursive formula for the efficient calculation of the link occupancy distribution, $G(j)$, j=0,1,…,T, we construct a reversible Markov chain that approximates the system by using state multipliers for all states $\boldsymbol{n} \in \boldsymbol{\Omega}$. The local balance equations between the adjacent states $\boldsymbol{n}_k^{-1}$ = (n_1, n_2,…, n_k– 1,…, n_K) and $\boldsymbol{n}$=(n_1, n_2, …, n_k, …,n_K) have the form:

$$P(\boldsymbol{n}_k^{-1})(N_k - n_k + 1)v_k = P(\boldsymbol{n})\phi_k(\boldsymbol{n})\mu_k n_k \,,\; k \in K_e \tag{4}$$

$$P(\boldsymbol{n}_k^{-1})(N_k - n_k + 1)v_k = P(\boldsymbol{n})\phi_k(\boldsymbol{n})\mu_k n_k \,,\; k \in K_a \tag{5}$$

where $\phi_k(\boldsymbol{n})$ is a state multiplier and is defined as:

$$\phi_k(\boldsymbol{n}) = \begin{cases} 1 & , when\ \boldsymbol{nb} \leq C\ and\ \boldsymbol{n} \in \boldsymbol{\Omega} \\ \dfrac{x(\boldsymbol{n}_k^{-1})}{x(\boldsymbol{n})}, & when\, C < \boldsymbol{nb} \leq T\ and\ \boldsymbol{n} \in \boldsymbol{\Omega} \\ 0 & , otherwise \end{cases} \tag{6}$$

and

$$x(\boldsymbol{n}) = \begin{cases} 1 & , when\ \ \boldsymbol{nb} \leq C,\ \boldsymbol{n} \in \boldsymbol{\Omega} \\ \dfrac{1}{C}\left(\sum_{k \in K_e} n_k b_k x(\boldsymbol{n}_k^{-1}) + r(\boldsymbol{n}) \sum_{k \in K_a} n_k b_k x(\boldsymbol{n}_k^{-1}) \right) & \\ & , when\ \ C < \boldsymbol{nb} \leq T,\ \boldsymbol{n} \in \boldsymbol{\Omega} \\ 0 & , otherwise \end{cases} \tag{7}$$

where $r(\boldsymbol{n}) = C/\boldsymbol{nb}$.

When $C < \boldsymbol{nb} \le T$ and $\boldsymbol{n} \in \boldsymbol{\Omega}$ the values of bandwidth of all in-service calls are compressed by a factor $\phi_k(\boldsymbol{n})$ so that:

$$\sum_{k\in K_e} n_k b_k^{comp} + \sum_{k\in K_a} n_k b_k^{comp} = C \qquad (8)$$

To derive (7), we keep the product 'service time' by 'bandwidth' of service-class k calls (elastic or adaptive) in state $\boldsymbol{n}$ of the irreversible Markov chain equal to the corresponding product in the same state $\boldsymbol{n}$ of the reversible Markov chain. This means that:

$$\frac{b_k r(\boldsymbol{n})}{\mu_k r(\boldsymbol{n})} = \frac{b_k^{comp}}{\mu_k \phi_k(\boldsymbol{n})} \Rightarrow b_k^{comp} = b_k \phi_k(\boldsymbol{n}),\ k \in K_e \qquad (9)$$

and

$$\frac{b_k r(\boldsymbol{n})}{\mu_k} = \frac{b_k^{comp}}{\mu_k \phi_k(\boldsymbol{n})} \Rightarrow b_k^{comp} = b_k \phi_k(\boldsymbol{n}) r(\boldsymbol{n}),\ k \in K_a \qquad (10)$$

Equation (7) results by substituting (9), (10) and (6), into (8).

In order to prove a recursive formula for the calculation of *G*(*j*)'s, we consider two cases: i) states where $0 \le j \le C$ and ii) states where $C < j \le T$.

When $0 \le j \le C$, then $\phi_k(\boldsymbol{n})$=1 and based on (4) and (5), it is proved that [14]:

$$G(j) = \begin{cases} 1 & \text{for } j = 0 \\ \dfrac{1}{min(j,C)} \displaystyle\sum_{k\in K} (N_k - n_k + 1)\alpha_k b_k G(j - b_k) & \\ & \text{for } j = 1,...,T \\ 0 & \text{otherwise} \end{cases} \qquad (11)$$

where: $\alpha_k = v_k/\mu_k$ is the offered traffic-load (in erl) per idle source of service-class *k*.

When $C < j \le T$, we multiply both sides of (4) by b_k^{comp} and sum over *k*=1,…,*K*e to have:

$$\sum_{k\in K_e} (N_k - n_k + 1) a_k b_k^{comp} P(\boldsymbol{n}_k^{-1}) = P(\boldsymbol{n}) \sum_{k\in K_e} n_k b_k^{comp} \phi_k(\boldsymbol{n}) \qquad (12)$$

Based on (6) and (9), (12) is written as:

$$x(\boldsymbol{n}) \sum_{k\in K_e} (N_k - n_k + 1) a_k b_k P(\boldsymbol{n}_k^{-1}) = P(\boldsymbol{n}) \sum_{k\in K_e} x(\boldsymbol{n}_k^{-1}) n_k b_k \qquad (13)$$

We continue by multiplying both sides of (5) by b_k^{comp} and sum over *k*=1,…,K_a to have:

$$\sum_{k\in K_a} (N_k - n_k + 1) a_k b_k^{comp} P(\boldsymbol{n}_k^{-1}) = P(\boldsymbol{n}) \sum_{k\in K_a} n_k b_k^{comp} \phi_k(\boldsymbol{n}) \qquad (14)$$

Based on (6) and (10) and since *r*(***n***) = *C*/*j*, (14) can be written as:

$$x(\boldsymbol{n}) \frac{C}{j} \sum_{k\in K_a} (N_k - n_k + 1) a_k b_k P(\boldsymbol{n}_k^{-1}) = P(\boldsymbol{n}) \frac{C}{j} \sum_{k\in K_a} x(\boldsymbol{n}_k^{-1}) n_k b_k \qquad (15)$$

Adding (13) and (15) we have:

$$x(\boldsymbol{n}) \left(\sum_{k\in K_e} (N_k - n_k + 1) a_k b_k P(\boldsymbol{n}_k^{-1}) + \frac{C}{j} \sum_{k\in K_a} (N_k - n_k + 1) a_k b_k P(\boldsymbol{n}_k^{-1}) \right)$$
$$= P(\boldsymbol{n}) \left(\sum_{k\in K_e} x(\boldsymbol{n}_k^{-1}) n_k b_k + \frac{C}{j} \sum_{k\in K_a} x(\boldsymbol{n}_k^{-1}) n_k b_k \right) \qquad (16)$$

Due to (7), (16) can be written as:

$$\sum_{k\in K_e} (N_k - n_k + 1) a_k b_k P(\boldsymbol{n}_k^{-1}) + \frac{C}{j} \sum_{k\in K_a} (N_k - n_k + 1) a_k b_k P(\boldsymbol{n}_k^{-1}) = CP(\boldsymbol{n}) \qquad (17)$$

To introduce the link occupancy distribution *G*(*j*) in (17), let $\boldsymbol{\Omega}_j = \{\boldsymbol{n} \in \boldsymbol{\Omega} : \boldsymbol{nb} = j\}$ be the state space where exactly *j* b.u. are occupied. Then, since $\sum_{\boldsymbol{n}\in\boldsymbol{\Omega}_j} P(\boldsymbol{n}) = G(j)$, summing both sides of (17) over $\boldsymbol{\Omega}_j$ we obtain:

$$\begin{aligned} &\sum_{\boldsymbol{n}\in\boldsymbol{\Omega}_j} \sum_{k\in K_e} (N_k - n_k + 1) a_k b_k P(\boldsymbol{n}_k^{-1}) + \\ &\frac{C}{j} \sum_{\boldsymbol{n}\in\boldsymbol{\Omega}_j} \sum_{k\in K_a} (N_k - n_k + 1) a_k b_k P(\boldsymbol{n}_k^{-1}) = CG(j) \end{aligned} \qquad (18)$$

Interchanging the order of summations in (18) and assuming that each state has a unique occupancy *j* we have:

$$\begin{aligned} &\sum_{k\in K_e} (N_k - n_k + 1) a_k b_k \sum_{\boldsymbol{n}\in\boldsymbol{\Omega}_j} P(\boldsymbol{n}_k^{-1}) + \\ &\frac{C}{j} \sum_{k\in K_a} (N_k - n_k + 1) a_k b_k \sum_{\boldsymbol{n}\in\boldsymbol{\Omega}_j} P(\boldsymbol{n}_k^{-1}) = CG(j) \end{aligned} \qquad (19)$$

or

$$\begin{aligned} &\sum_{k\in K_e} (N_k - n_k + 1) a_k b_k G(j - b_k) + \\ &\frac{C}{j} \sum_{k\in K_a} (N_k - n_k + 1) a_k b_k G(j - b_k) = CG(j) \end{aligned} \qquad (20)$$

The combination of (11) and (20) gives the approximate recursive formula of *G*(*j*)'s, when $1 \le j \le T$:

$$G(j)=\begin{cases}1 & \text{for } \ j=0\\ \dfrac{1}{\min(j,C)}\sum\limits_{k\in K_e}(N_k-n_k+1)\alpha_k b_k G(j-b_k) & \\ +\dfrac{1}{j}\sum\limits_{k\in K_a}(N_k-n_k+1)\alpha_k b_k G(j-b_k) & \text{for } j=1,\dots,T\\ 0 & \text{for } j<0\end{cases} \tag{21}$$

When $N_k \rightarrow \infty$ for k=1,…,K then the call arrival process is Poisson and the formula of $G(j)$'s is [2]:

$$G(j)=\begin{cases}1 & \text{for } \ j=0\\ \dfrac{1}{\min(j,C)}\sum\limits_{k\in K_e}\alpha_k b_k G(j-b_k)+ & \\ \dfrac{1}{j}\sum\limits_{k\in K_a}\alpha_k b_k G(j-b_k) \ \text{for} & j=1,\dots,T\\ 0 & \text{for } j<0\end{cases} \tag{22}$$

where α_k=λ_k/μ_k (in erl) and λ_k is the arrival rate of calls of service-class k.

The determination of $G(j)$'s in (21) requires the value of n_k which is unknown. In other finite multirate loss models (e.g., [14], [15]) there exist calculation methods for the determination of n_k in each state j through the use of an equivalent stochastic system, with the same traffic description parameters and exactly the same set of states. However, the state space determination of the equivalent system is complex, especially for large capacity systems that serve many service-classes. Thus, we avoid such methods and approximate n_k in state j, $n_k(j)$, as the mean number of service-class k calls in state j, $y_k(j)$, when Poisson arrivals are considered, i.e., $n_k(j)\approx y_k(j)$. Such approximations are common in the literature and induce little error (e.g., [16],[17]). The values of $y_k(j)$ are given by (23), (24) in the case of elastic and adaptive service-classes, respectively [2]:

$$\begin{aligned} y_k(j)G(j) &= \frac{1}{\min(C,j)}a_k b_k G(j-b_k)\big(y_k(j-b_k)+1\big)\\ &+\frac{1}{\min(C,j)}\sum_{\substack{i=1\\ i\neq k}}^{K_e} a_i b_i G(j-b_i)y_k(j-b_i)\\ &+\frac{1}{j}\sum_{i=1}^{K_a} a_i b_i G(j-b_i)y_k(j-b_i)\end{aligned} \tag{23}$$

$$\begin{aligned} y_k(j)G(j) &= \frac{1}{j}a_k b_k G(j-b_k)\big(y_k(j-b_k)+1\big)\\ &+\frac{1}{j}\sum_{\substack{i=1\\ i\neq k}}^{K_a} a_i b_i G(j-b_i)y_k(j-b_i)\\ &+\frac{1}{\min(C,j)}\sum_{i=1}^{K_e} a_i b_i G(j-b_i)y_k(j-b_i)\end{aligned} \tag{24}$$

where the values of $G(j)$'s are determined by (22).

Having determined $G(j)$'s according to (21), we calculate the following performance measures:

1) The TC probabilities of service-class k, denoted as P_{b_k}, which is the probability that at least T-b_k+1 b.u are occupied:

$$P_{b_k} = \sum_{j=T-b_k+1}^{T} G^{-1}G(j) \tag{25}$$

where: $G=\sum\limits_{j=0}^{T}G(j)$ is a normalization constant.

TC probabilities are determined by the proportion of time the system is congested.

2) The CC probabilities of service-class k, denoted as C_{b_k}, which is the probability that a new service-class k call is blocked and lost:

$$C_{b_k} = \sum_{j=T-b_k+1}^{T} G^{-1}G(j) \tag{26}$$

where $G(j)$'s are determined for a system with N_k - 1 traffic sources.

CC probabilities are determined by the proportion of arriving calls that find the system congested.

3) The link utilization, denoted as U:

$$U=\sum_{j=1}^{C} jG^{-1}G(j)+\sum_{j=C+1}^{T} CG^{-1}G(j) \tag{27}$$

III. Evaluation

In this section, we present an application example and compare the analytical results of the TC probabilities, CC probabilities and link utilization obtained by the proposed model and the E-EMLM. The corresponding simulation results, presented for the proposed model only, are mean values of 6 runs. Simulation is based on Simscript II.5 [18].

We consider a single link of capacity C = 90 b.u. that accommodates calls of three service-classes. The first two service-classes are elastic, while the third service-class is adaptive. The traffic characteristics of each service-class are:

1st service-class: N_1=200, v_1 = 0.10, b_1 = 1 b.u.
2nd service-class: N_2=200, v_2 = 0.04, b_2 = 4 b.u.
3rd service-class: N_3=200, v_3 = 0.01, b_3 = 6 b.u.

In the case of the E-EMLM the corresponding Poisson traffic loads are: α_1= 20 erl, α_2= 8 erl and α_3= 2 erl.

We also consider two values of T: a) T = 90 b.u. , where no bandwidth compression takes place and b) T = 100 b.u., where bandwidth compression takes place and $r_{\min}$ = C/T=0.9. In the x-axis of all figures, v_1 and v_2 increase in steps of 0.01 and 0.005 erl, respectively while v_3 remains constant. So in Point 1 we have (v_1, v_2, v_3) = (0.10, 0.04, 0.01), while in Point 6 (v_1, v_2, v_3) = (0.15, 0.065, 0.01). In the case of the E-EMLM the corresponding Poisson traffic loads in Point 1 and Point 8 are (α_1, α_2, α_3)= (20, 8, 2) and (α_1, α_2, α_3)= (30, 13, 2), respectively.

In Figs. 1-3, we present the analytical and the simulation TC probabilities of the three service-classes while in Figs. 4-6, we present the corresponding analytical and simulation CC probabilities. In all cases, both T=90 and T=100 b.u. are considered. Note that the term N=inf. in all figures refers to the E-EMLM where the number of traffic sources is infinite for each service-class. All figures show that: i) analytical and simulation results for both TC and CC probabilities are very close, ii) the application of the compression/expansion mechanism reduces congestion probabilities compared to those obtained when C=T=90 b.u. and iii) the results obtained by the E-EMLM fail to approximate the corresponding results obtained by the proposed model (quasi-random traffic model). Finally, in Fig. 7, we present the analytical and simulation results of the link utilization (in b.u.). It is clear, that the application of the bandwidth compression/expansion mechanism increases link utilization since it decreases CC probabilities.

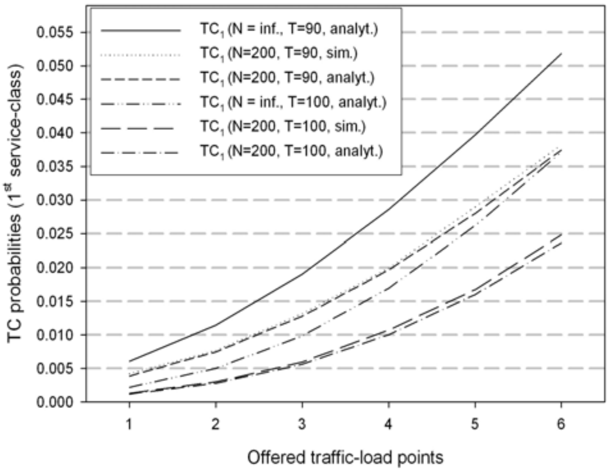


Figure 1. TC probabilities of the 1st service-class.

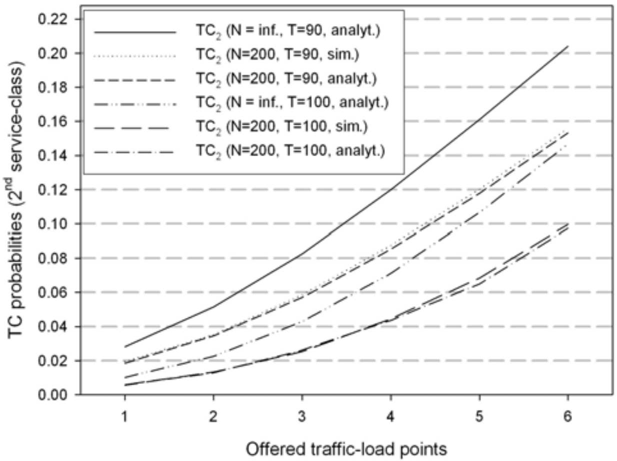


Figure 2. TC probabilities of the 2nd service-class.

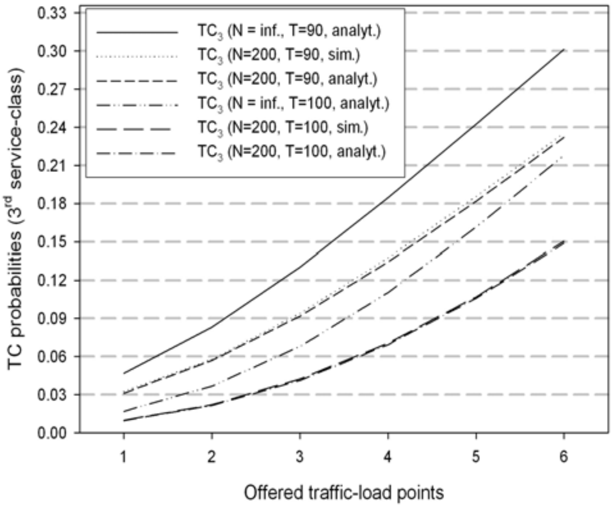


Figure 3. TC probabilities of the 3rd service-class.

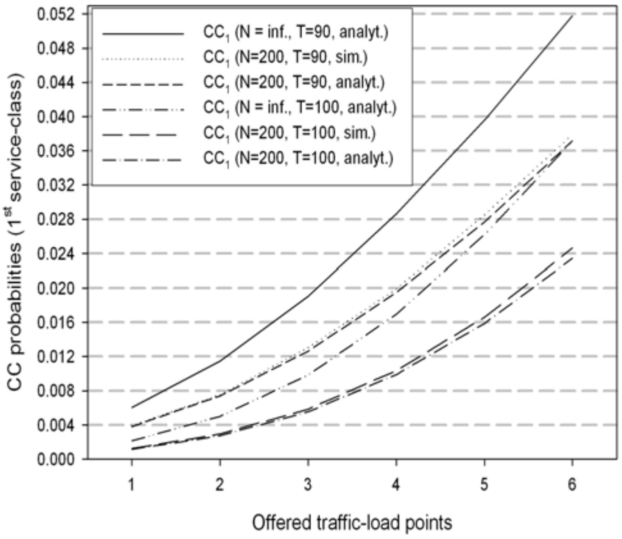


Figure 4. CC probabilities of the 1st service-class.

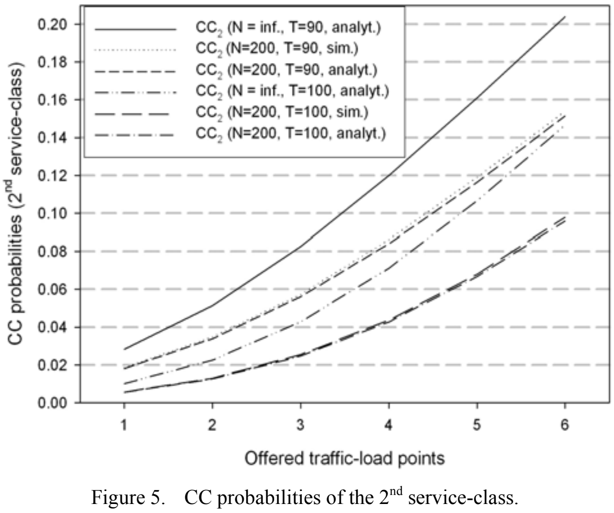


Figure 5. CC probabilities of the 2nd service-class.

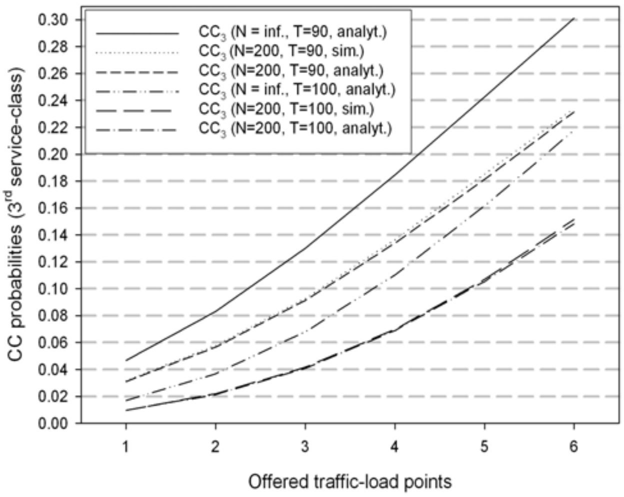


Figure 6. CC probabilities of the 3rd service-class.

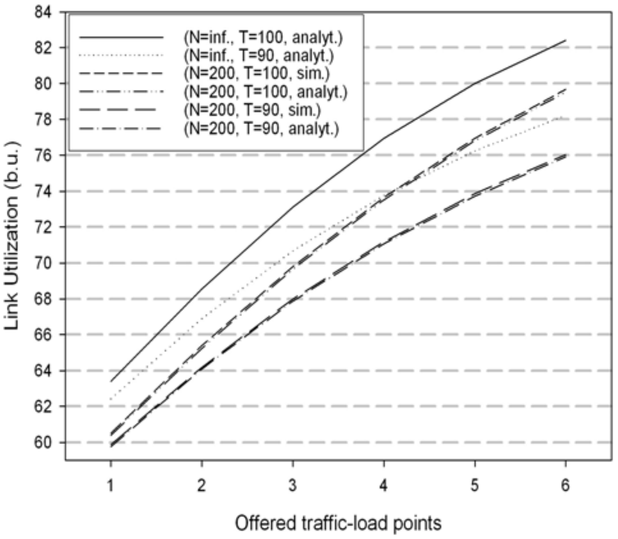


Figure 7. Link utilization (in b.u.)

IV. CONCLUSION

We propose an analytical model for the call-level performance assessment of telecom networks, when elastic and/or adaptive calls of different service-classes come from finite traffic-sources and compete for the available bandwidth of a single link with certain capacity. Due to the existence of the bandwidth compression/expansion mechanism, the proposed model does not have a product form solution. Therefore, we propose approximate but recursive formulas for the calculation of the most important performance measures, namely TC and CC probabilities and link utilization. Simulation results verify the analytical results and prove the accuracy and the consistency of the proposed model. Furthermore, the comparison of the results obtained by the proposed model and the E-EMLM shows the necessity of the proposed model, since the E-EMLM fails to approximate the case of quasi-random traffic. Potential applications of the proposed model are in the environment of wireless networks that support elastic and adaptive traffic.

ACKNOWLEDGEMENT

The authors would like to thanks Dr. V. Vassilakis (PostDoc., University of Essex, U.K.), for his contribution.

REFERENCES

[1] K. Ross, Multiservice loss models for broadband telecommunication networks, Springer, 1995, ISBN 3-540-19918-7.

[2] S. Racz, B. Gero, and G. Fodor, “Flow level performance analysis of a multi-service system supporting elastic and adaptive services”, Performance Evaluation, vol. 49, Sept. 2002, pp. 451-469.

[3] I. Moscholios, V. Vassilakis, J. Vardakas and M. Logothetis, “Call Blocking Probabilities of Elastic and Adaptive Traffic with Retrials”, Proc. of 8th Advanced Int. Conf. on Telecommunications, AICT 2012, Stuttgart, Germany, 27 May-1 June 2012, pp. 92-97.

[4] V. Vassilakis, I. Moscholios and M. Logothetis, “The extended connection-dependent threshold model for call-level performance analysis of multi-rate loss systems under the bandwidth reservation policy”, Int. J. Commun. Syst., Wiley, vol. 25, issue 7, pp. 849-873, July 2012.

[5] I. Moscholios, J. Vardakas, M. Logothetis and A. Boucouvalas, “QoS Guarantee in a Batched Poisson Multirate Loss Model Supporting Elastic and Adaptive Traffic”, Proc. of IEEE ICC 2012, Ottawa, Canada, 10-15 June 2012, pp. 1296-1301.

[6] J. Kaufman, “Blocking in a shared resource environment”, IEEE Trans. Commun. vol. 29, Oct. 1981, pp. 1474-1481.

[7] J. Roberts, “A service system with heterogeneous user requirements”, in: G. Pujolle (Ed.), Performance of Data Communications systems and their applications, North Holland, Amsterdam, 1981, pp.423-431.

[8] B. P. Gerő, P. L. Pályi and S. Rácz, “Flow-level performance analysis of a multi-rate system supporting stream and elastic services”, Int. J. Commun. Syst., Wiley, doi: 10.1002/dac.1383, 2012.

[9] H. Akimaru and K. Kawashima, Teletraffic – Theory and Applications, 2nd edition, Springer-Verlag, Berlin, 1999.

[10] G. Kallos, V. Vassilakis, I. Moscholios, and M. Logothetis, “Performance Modelling of W-CDMA Networks Supporting Elastic and Adaptive Traffic”, in Proc. 4th Int. Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks, Ilkley, U.K., 11-13 Sept. 2006, pp. 09/1-09/10.

[11] G. Stamatelos and V. Koukoulidis, “Reservation – Based Bandwidth Allocation in a Radio ATM Network”, IEEE/ACM Trans. Networking, vol. 5, June 1997, pp. 420-428.

[12] G. Fodor and M. Telek, “Bounding the Blocking Probabilities in Multirate CDMA Networks Supporting Elastic Services”, IEEE/ACM Trans. on Networking, vol. 15, Aug. 2007, pp. 944-956.

[13] V. Iversen, “Teletraffic Theory and Network Planning”, Technical University of Denmark, 2011, available at (retrieved: July 2012): http://oldwww.com.dtu.dk/education/34340/material/telenook2011pdf.pdf

[14] G. Stamatelos and J. Hayes, “Admission control techniques with application to broadband networks”, Comput. Commun., vol. 17, no. 9, pp. 663-673, 1994.

[15] I. Moscholios, M. Logothetis and P. Nikolaropoulos, “Engset Multi-Rate State-Dependent Loss Models”, Performance Evaluation, Vol. 59, Issues 2-3, pp. 247-277, February 2005.

[16] M. Glabowski and M. Stasiak, “An approximate model of the full-availability group with multi-rate traffic and a finite source population”, in Proc. of 12th MMB&PGTS, Dresden, Germany, pp. 195-204, Sept. 2004.

[17] M. Glabowski, K. Kubasik, and M. Stasiak, “Modelling of Systems with Overflow Multi-rate Traffic and Finite Number of Traffic Sources,” in Proc. CNSDSP 2008, pp. 196–199, July 2008.

[18] Simscript II.5, http://www.simscript.com (retrieved: July 2012).

A New Architecture for Trustworthy Autonomic Systems

Thaddeus O. Eze, Richard J. Anthony, Chris Walshaw and Alan Soper
Autonomic Computing Research Group
School of Computing & Mathematical Sciences (CMS)
University of Greenwich, London, United Kingdom
{T.O.Eze, R.J.Anthony, C.Walshaw and A.J.Soper}@gre.ac.uk

***Abstract* — This paper presents work towards a new architecture for trustworthy autonomic systems (different from the traditional autonomic computing architecture) that includes mechanisms and instrumentation to explicitly support run-time self-validation and trustworthiness. The state of practice does not lend itself robustly enough to support trustworthiness and system dependability. For example, despite validating system's decisions within a logical boundary set for the system, there's the possibility of overall erratic behaviour or inconsistency in the system. So a more thorough and holistic approach, with a higher level of check, is required to convincingly address the dependability and trustworthy concerns. Validation alone does not always guarantee trustworthiness as each individual decision could be correct (validated) but overall system may not be consistent or dependable. A new approach is required in which, validation and trustworthiness are designed in and integral at the architectural level, and not treated as add-ons as they cannot be reliably retro-fitted to systems. In this paper we analyse current state of practice in autonomic architecture and propose a different architectural approach for trustworthy autonomic systems. To demonstrate the feasibility and practicability of our approach, a case example scenario is examined. The example is a deployment of the architecture to an envisioned Autonomic Marketing System that has many dimensions of freedom and which is sensitive to a number of contextual volatility.**

Keywords - trustworthy architecture; trustability; validation; autonomic marketing; autonomic system; dependability

I. INTRODUCTION

The autonomic architecture as originally presented in the autonomic computing blueprint [1] has been widely accepted and deployed across an ever-widening spectrum of autonomic system (AS) design and implementations. Research results in the autonomic research community are based, predominantly, on the architecture's basic MAPE (monitor-analyse-plan-execute) control loop, e.g., [13][14][15][16]. Although several implementation variations of this control loop have been promoted, alternative approaches (e.g., [17]) have also been proposed. In [17], Shuaib *et al.* presented an 'alternative' autonomic architecture based on Intelligent Machine Design (IMD), which draws from the human autonomic nervous system. However, research [11] shows that most approaches are MAPE [2] based. Despite progress made, the traditional autonomic architecture and its variations is not sophisticated enough to produce trustworthy ASs. A new approach with inbuilt mechanisms and instrumentation to support trustworthiness is required.

At the core of system trustworthiness is validation and this has to satisfy run-time requirements. In large systems with very wide behavioural space and many dimensions of freedom, it is close to impossible to comprehensively predict possible outcomes at design time. So it becomes highly complex to make sure or determine whether the autonomic manager's (AM's) decision(s) are in the overall interest and good of the system. There is a vital need, then, to dynamically validate the run-time decisions of the AM to avoid the system 'shooting itself on the foot' through control brevity. The traditional autonomic architecture does not explicitly and integrally support run-time self-validation; a common practice is to treat validation and reliability as add-ons. Identifying such challenges, the traditional architecture has been extended (e.g., in [3]) to accommodate validation. Diniz *et al.* [3] extended the MAPE control loop to include a new function called *test*. By this it defines a new control loop comprising Monitor, Analyse, Decision, Test and Execute –MADTE activities. The main point here is that a *self-test* activity is integrated into the autonomic architecture to provide a run-time validation of AM decision-making processes. But the question is can validation alone guarantee trustworthiness.

The peculiarity of context dynamism in autonomic computing places unique and complex challenges on trustworthy ASs that validation alone cannot sufficiently address. Take for instance; if a manager (AM) erratically changes its mind, it ends up introducing noise to the system rather than smoothening the system. In that instance, a typical validation check will pass each correct decision (following a particular logic or rule) but this could lead to oscillation in the system resulting in instability and inconsistent output. A typical example could be an AM that follows a set of rules to decide when to move a server to or from a pool of servers. As long as the conditions of the rules are met, the AM will move servers around not minding the frequency of changes in the conditions. An erratic change of mind (high rate of moving servers around) will cause undesirable oscillations that ultimately detriment the system. What is required is a kind of intuition that enables the manager to carry out a change only when it is safe and efficient to do so – within a particular safety margin. A higher level of self-monitoring to achieve, e.g., stability over longer time frames, is absent in the MAPE-oriented architectures. This is why ASs need a different approach. The ultimate goal is not just to achieve self-management but to achieve consistency and reliability of results through self-management. These are the core values of the proposed architecture.

We look at the current state of practice in the work towards AS trustworthy architecture in Section II. We propose an AS trustworthy architecture in Section III and present a case example in Section IV. Section V concludes the paper.

II. CURRENT STATE OF PRACTICE TOWARDS TRUSTWORTHY ARCHITECTURE

In this section, we look at the current state of practice and efforts directed towards AS trustworthiness. We analyse few proposed trustworthy architectures and some isolated bits of work that could contribute to trustworthy autonomic computing. Trustworthiness requires a holistic approach, i.e., a long-term focus as against the near-term needs that merely address methods for building trust into existing systems. This means that trustworthiness needs to be designed into systems as integral properties.

A trustworthy autonomic grid computing architecture is presented in [4]. This is to be enabled through a proposed fifth self-* functionality, *self-regulation*. Self-regulating capability is able to derive policies from high-level policies and requirements at run-time to regulate self-managing behaviours. One concern here is that proposing a fifth autonomic functionality to regulate the other functionalities as a solution to AS trustworthiness assumes that trustworthiness can be achieved when all four functionalities perform 'optimally'. The four self-* functionalities alone do not ensure trustworthiness in ASs. For example, the self-* functionalities do not address *validation* which is a key factor in AS trustworthiness. Amongst effort focused on validation include [3][5][6]. As explained earlier, Diniz *et al.* [3] has extended the MAPE-based autonomic architecture to incorporate a self-test activity to guarantee run-time validation of AM decisions. This is a huge step towards AS trustworthiness. The approach in [5][6] is another extension of the MAPE-based structure to include self-testing as an integral and implicit part of the AS. The same model for AS management using autonomic managers (AMs) is replicated for the self-testing. In the self-test structure, test managers (TMs) (which extend the concept of AMs to testing activities) implement closed control loops on AMs (such as AMs implement on managed resources) to validate change requests generated by AMs. Although not a 'trustworthy' solution in itself, King *et al.* [5] introduces an important concept (nested control looping) useful for the proposed trustworthy architecture as explained in Section III.

Another idea is that trustworthiness is achieved when a system is able to provide accounts of its behaviour to the extent that the user can understand and trust. But these accounts must, amongst other things, satisfy three requirements: provide a representation of the policy guiding the accounting, some mechanism for validation and accounting for system's behaviour in response to user demands [7]. The system's actions are transparent to the user and also allows the user (if required) the privilege of authorising or not authorising a particular process. This is a positive step (at least it provides the user a level of confidence and trust) but also important is a mechanism that ensures that any 'authorised' process does not lead to oscillation and/or instability in the system resulting in misleading or unreliable results. One powerful way of addressing this challenge is by implementing a *dead-zone* (DZ) logic presented in [8]. A DZ, which is a simple mechanism to prevent unnecessary, inefficient and ineffective control brevity when the system is sufficiently close to its target value, is implemented in [8] using Tolerance-Range-Check (TRC) object. The TRC object encapsulates DZ logic and a three-way decision fork that flags which action (left, null or right) to take depending on the rules specified. The size of the DZ can be dynamically adjusted to suit changes in environmental volatility. A key use of dead-zones is to reduce oscillation and ensure stability despite high extent of adaptability. A mechanism to automatically monitor the stability of an autonomic component, in terms of the rate the component changes its decision (for example when close to a threshold tipping point), was presented in [12]. The *DecisionChangeInterval* property is implemented in the AGILE policy language [12] on decision making objects such as rules and utility functions. This allows the system to monitor itself and take action if it detects instability at a higher level than the actual decision making activity.

A. *Trustworthy architecture life-cycles representing current practice*

We argue that trustworthiness cannot be reliably retrofitted into systems but must be designed into system architectures. We track autonomic architecture (leading to trustworthiness) pictorially in a number of progressive stages addressing it in an increasing level of detail and sophistication. Figure 1 provides a key to the symbols used.

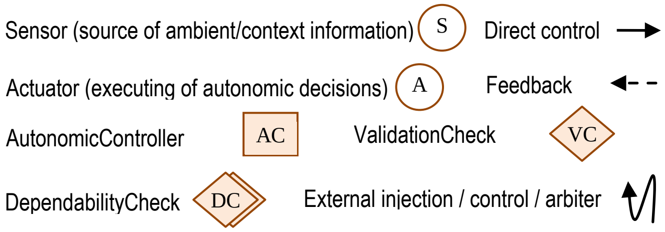

Sensor (source of ambient/context information)
S
Direct control
Actuator (executing of autonomic decisions)
A
Feedback
AutonomicController
AC
ValidationCheck
VC
DependabilityCheck
DC
External injection / control / arbiter

Figure 1. Pictographic key used for the architecture

Figure 2 illustrates the progression, in sophistication, of autonomic architectures and how close they have come to achieving trustworthiness. Although this may not be exhaustive as several variations and hybrids of the combinations may exist, it represents a series of discrete progressions in current approaches. Two distinct levels of sophistication are found: 1. The traditional autonomic architecture (a and b) basically concerned with direct self-management of controlled/monitored system following some basic *sense-manage-actuate* logic defined in AC. For the prevailing context, AC is just a container of autonomic logic which, could be based on MAPE or any other control logic. To add a degree of trust and safeguard, an external interface for user control input is introduced in (b). This chronicles such approaches that provide a console for external administrative interactions (e.g., real-time monitoring, tweaking, feedback, knowledgebase source, trust input, etc.) with the autonomic process. 2. On the horizon (c and d) are efforts towards addressing run-time validation. Systems are able to check the conformity of management decisions and where this check fails; VC sends feedback to AC with notification of failure (e.g., policy violation) and new decision is generated. An additional layer of sophistication is introduced (d) with external touch-point for higher level of manageability control. This can be in the form of an outer control loop monitoring over a longer time frame an inner (shorter time frame) control loop (e.g., as presented in [5]).

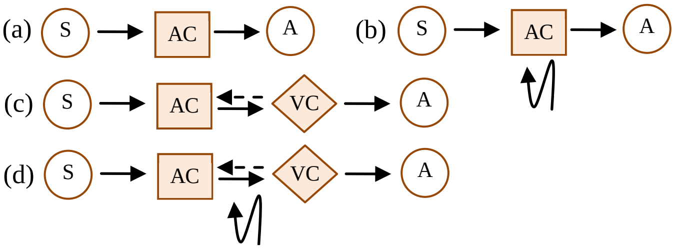


Figure 2. Pictorial representation of trustworthy autonomic architecture life-cycles.

At the level of current sophistication (state-of-the-art), there are techniques to provide run-time validation check (for behavioural and structural conformity), additional console for higher level (external) control, etc. Emerging and needed capabilities include techniques for managing oscillatory behaviour in ASs. These are mainly implemented in isolation. What is required is a holistic framework that collates all these capabilities into a single autonomic *unit*. Policy autonomics is one of the most used autonomic solutions. Autonomic managers (AMs) follow rules to decide on actions. As long as policies are validated against set rules the AM adapts its behaviour accordingly. This may mean changing between states. And when the change becomes rapid (despite meeting validation requirements) it is capable of introducing oscillation, vibration and erratic behaviour (all in form of noise) into the system. This is more noticeable in highly sensitive systems. So a trustworthy autonomic architecture (TAA) needs to provide a way of addressing these issues.

III. TRUSTWORTHY AUTONOMIC ARCHITECTURE

In this section we introduce our proposed TAA. We start with a general view of the architecture and then move on to explain its components. Figure 3 explains a trustworthy autonomic architecture that embodies self-validation and dependability. The architecture builds on the traditional autonomic computing solution (denoted as the *AutonomicController* component). Other components include *ValidationCheck* (which is integrated with the decision-making object to validate all *AutonomicController* decisions) and *DependabilityCheck* component which, guarantees stability and reliability after validation.

The *AutonomicController* component (based on e.g., MAPE logic, Intelligent Machine Design framework, etc.) monitors the managed sub-system for context information and takes decision for action based on this information. The decided action is validated against the system's goal (described as policies) by the *ValidationCheck* component before execution. If validation fails (e.g., policy violation), it reports back to the *AutonomicController* otherwise the *DependabilityCheck* is called to ensure that outcome does not lead to, e.g., instability in the system.

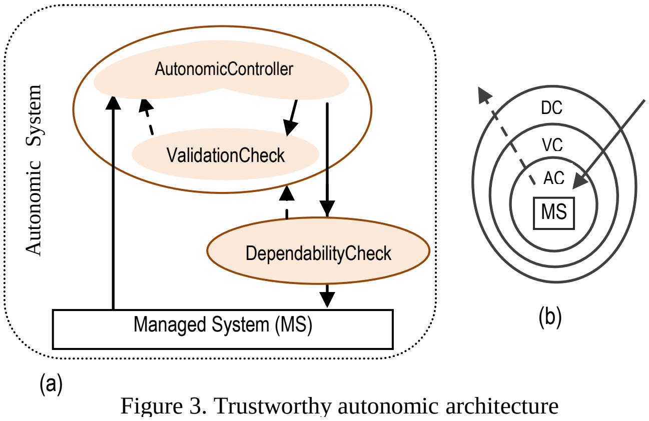


Figure 3. Trustworthy autonomic architecture

The *DependabilityCheck* component has a sub-component (*Predictive* sub-component) that allows it to predict the outcome of the system based on the validated decision. It either prevents execution and sends feedback in form of some input parameters to the *AutonomicController* or calls the actuator.

A. Overview of the proposed architecture components

We present the architecture in a number of progressive stages addressing it in an increasing level of detail. First, we define the self-management process as a ***Sense–Manage–Actuate*** loop where *Sense* and *Actuate* define Touchpoints (the AM's interface with a managed system) and '*Manage'* the embodiment of the autonomic management. Figure 4 is a detailed representation of the architecture.

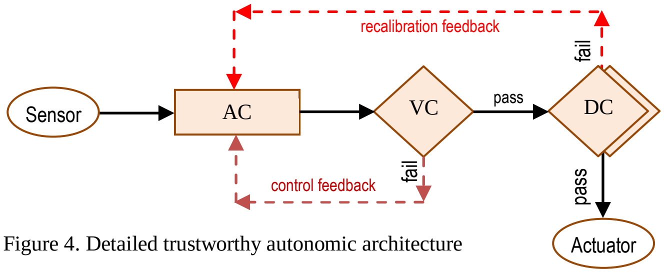


Figure 4. Detailed trustworthy autonomic architecture

Traditionally, the AutonomicController (AC) senses context information, decides (following some rules) on what action to take and then executes the action. This is the basic routine of an AM and is at the core of most of the autonomic architectures in use today (Figure 2). At this level the autonomic *unit* matters but the *content* of the *unit* does not matter much, i.e., it does not matter what autonomic logic (e.g., MAPE, IMD, etc.) that is employed as far as it provides the desired autonomic functionalities. So, the AC component provides designers the platform to express rules that govern target goal and policies that drive decisions on context information for system adaptation to achieve the target goal.

But, the nature of ASs raises one significant concern; input variables (context info) are dynamic and (most times) not predictable. Although rules and policies are carefully and robustly constructed, sensors sometimes do inject *rogue* variables that are capable of thwarting process and policy deliberations. In addition, the operating environment itself can have varying volatility –causing a controller to become unstable in some circumstances. Thus a mechanism is needed to mitigate behavioural (e.g., contradiction between two policies, goal distortion, etc.) and structural (e.g., illegal structure not conforming to requirement, division by zero, etc.) anomalies. This is where the ValidationCheck (VC) component comes in. It should be noted that AC will always decide on action(s) no matter what the input variable is. Once the AC reaches a decision, it passes control to the VC which then validates the decision and calls the Actuator (Figure 2c) or the DependabilityCheck (DC) (Figure 4) otherwise it sends feedback to AC if the check fails (while retaining previous *passed* decision). The VC is a higher level mechanism that oversees the AM to keep the system's goal on track. The ultimate concern here is to maintain system goal adhering to defined rules, i.e., adding a level of trust by ensuring that target goal is reached only within the boundaries of specified rules. It is then left for designers to define what constitute *validation pass* and *validation fail*. Actual component logic are application specific but some examples in literature include fuzzy logic [18], reinforcement learning [19], etc.

It is also important to consider situations above this level where, despite the AM taking legitimate decisions within the boundaries of specified rules, there's the possibility of overall inconsistency in the behaviour of the system. I.e., each individual decision could be correct (by logic) but the overall behaviour is wrong. A situation where the AM erratically (though legally) changes its mind, thereby injecting oscillation into the system, is a major concern especially in large scale and sensitive systems. Therefore it is necessary to find a way of enabling the AM to avoid unnecessary and inefficient change of decision that could lead to oscillation. This task is handled by the DC component. It allows the AM change its decision (i.e., adapt) only when it is necessary and safe to do so. Consider a simple example of a room temperature controller in which, it is necessary to track a dynamic goal –a target room temperature. The AM is configured to maintain the target temperature by automatically switching heating *ON* or *OFF*. A VC would allow any decision or action that complies with the basic logic 'IF *RoomTemperature* < *TargetTemperature* THEN *ONHeating* ELSE IF *RoomTemperature* > *TargetTemperature* THEN *OFFHeating*'. With the lag in adjusting the temperature the system may decide to switch *ON* or *OFF* heating at every slight tick of the gauge below or above target (when room temperature is sufficiently close to the target temperature). This may in turn cause oscillation which, can lead to undesirable effects. The effects are more pronounced in more sensitive and critical systems where such changes come at some cost. For example, a data centre management system that erratically switches servers between pools at every slight fluctuation in demand load is cost ineffective. One simple way of configuring a DC to mitigate this problem is by using dead-zone logic. In this case, a system has to exceed a boundary by a minimum amount before action is taken. Small deviations into the dead-zone do not result in actuations. The DC component may also implement other sub-components like *Prediction*, *Learning*, etc. This enables it to predict (based on knowledge, trend analysis, etc.) the outcome of the system and to decide whether it is safe to allow a particular decision or not. So after validation phase, the DC is called to check (based on specified rules) for dependability. DC avoids unnecessary and inefficient control inputs to maintain stability. If the check passes, control is passed to the Actuator otherwise feedback is sent to AC. DC is capable of tweaking input to the controller as feedback from its prediction. A particular aspect of concern is that for dynamic systems the boundary definition of DC may itself be context dependent (e.g., in some circumstances it may be appropriate to allow some level of changes which under different circumstances may be considered destabilizing).

Consider the whole architecture as a nested control loop (Figure 3b) with AC the core control loop while VC and DC are intermediate and outer control loops respectively. In summary, a system, no matter the context of deployment, is truly trustworthy when its actions are continuously validated (i.e., at run time) to satisfy set requirements (system goal) and results produced are dependable and not misleading.

IV. CASE EXAMPLE

This example is used to illustrate how powerful our proposed architecture is (in terms of cost savings, improved reliability and trustability) when compared to traditional architectures. We compare three autonomic managers that are based on AC (Figure 2a), AC+VC (Figure 2c) and AC+VC+DC (Figure 4). We use rule-based (policy autonomics) approach in this example.

The case example used deploys one of the current technology innovations –Autonomic Marketing. Autonomic Marketing employs the fundamentals of autonomic computing to monitor the market ambience and uses current (real-time) information to formulate appropriate marketing strategies for dynamic, adaptive and effective target marketing. The term is used to describe a step-change in the sophistication of automated marketing systems, in which the marketing activity itself is dynamically configured and contextualised to suit the current market conditions [9]. This has been proposed by the Autonomic Marketing Interest Group (AMIG) and they have in [9] defined some initial concepts and promise of the technology. An autonomic marketing system tracks current market state (which can be from several sources and is subject to influences such as market conditions, customer demographics, significant world events, trends from social media analysis, weather, seasonal information, etc.) and makes marketing decisions based on the analysis of the information gathered. This is representative of many real-world systems of high complexity and sensitive to several sources of environmental volatility.

In this example, we implement a particular aspect of Autonomic Marketing, that of targeted television advertising during a live sports competition airing. A company is interested in running an adaptable marketing campaign on television with different adverts (of different products appealing to audiences of different demographics) to be aired at different times during a live match between two teams. There are three adverts (Ad1, Ad2 and Ad3) to be run and the choice of an ad will be influenced by, amongst other things, viewer demographics, time of ad (local time, time in game, e.g., half time, TV peak/off-peak time, etc.), length of ad (time constraint), cost of ad, who is winning in the game, etc. This is a typical example of a system with many dimensions of freedom and very wide behaviour space. For brevity, we divide the behaviour space into four different zones and express them along two dimentions of freedom (*Mood* and *CostImplication*) as shown in Figure 5.

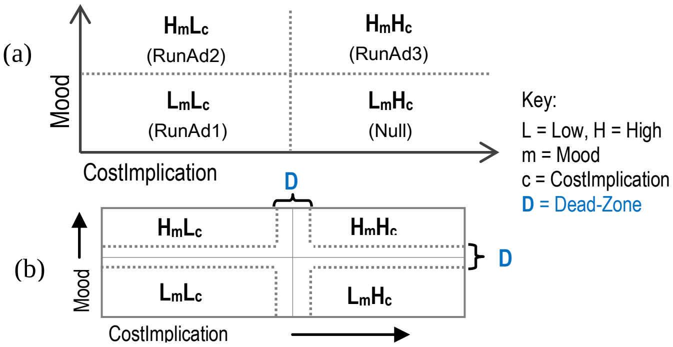


Figure 5. System behaviour space in two dimensions of freedom

The two dimensions of freedom represent a collation of all possible decision influencers into two key external variables –*Mood* and *CostImplication*. *Mood* is defined by two variables (*MatchScore* i.e., info about who's winning and *WeatherInfo*) while *CostImplication* is defined by another two variables (*TimeOfAd* and *LengthOfAd*). An action (in this case, RunAd 1, 2 or 3 or Null) is defined for each zone. Each action (ad) is thus activated only in its allocated zone following specified policy (excerpt shown in Figure 6). Internal variables (e.g., L_BenchMarkMatchScore and U_BenchMarkTimeOfAd), design-time specified, are used to define decision benchmarks.

```
If MatchScore < L_BenchMarkMatchScore And WeatherInfo < L_BenchMarkWeatherInfo Then
      Mood = "LOWMood"
   ElseIf MatchScore > U_BenchMarkMatchScore And WeatherInfo > U_BenchMarkWeatherInfo
            Then
       Mood = "HIGHMood"
   Else : Mood = "Null"
End If
If TimeOfAd < L_BenchMarkTimeOfAd And LengthOfAd < L_BenchMarkLengthOfAd Or
            TimeOfAd > U_BenchMarkTimeOfAd And LengthOfAd > U_BenchMarkLengthOfAd
            Then
      CostImplication = "LOWCostImplication"
   ElseIf TimeOfAd > U_BenchMarkTimeOfAd And LengthOfAd > M_BenchMarkLengthOfAd Then
      CostImplication = "HIGHCostImplication"
   Else :CostImplication = "Null"
 End If
Select Case DecisionParameter
    Case "LOWMoodLOWCostImplication"
         CurrentAction(CurrentActionCounter) = "RunAd1"
    Case "HIGHMoodLOWCostImplication"
         CurrentAction(CurrentActionCounter) = "RunAd2"
    Case "HIGHMoodHIGHCostImplication"
         CurrentAction(CurrentActionCounter) = "RunAd3"
    Case Else
         CurrentAction(CurrentActionCounter) = "NullAction"
 End Select
```

Figure 6. Excerpt of decision policy used.

The system goal is defined by a set of rules (Figure 7) that the AM must adhere to in making decisions. Basically, AC is concerned with making decisions within the boundaries of the rules while VC validates decisions for conformity with the rules. DC verifies that the measure of success is achieved. DC also improves reliability by instilling stability in the system. One way of achieving this is by introducing dead-zone boundaries (Figure 5b) within which, no action is taken (avoiding erratic and unnecessary changes) –in this case, a running ad is not changed. The size of the boundaries, which, though can be dynamically adjusted to suit real-time changes, is initially design-time specified.

1. Extract external variables (decision parameters) at defined time interval and decide on action
2. Send trap msg and change action if (*condition omitted*) otherwise retain previous action
3. …
4. If current action is same as previous action, do not send trap and do not change action

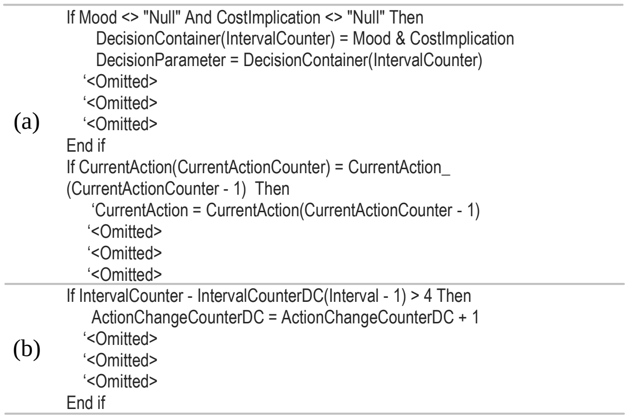
   ================Measure of Success================

5. Cost of action change (total ad run) must fall within budget
6. Rate of change should be considerably reasonable
7. …
8. Turnover should justify cost

Figure 7. Excerpt of rules defining system goal.

AC will, at every sample collection, decide (by running the policy in Figure 6) which action (ad) to run. Because it is wired to make fresh decision at every policy run, it is bound to send trap (notice of change of ad). But before that decision is implemented, VC validates it for *pass/fail*. It is important to define what *pass/fail* means in this context: if decided action is same as previous action (running ad), VC returns *fail* (then no trap is sent and no change is made) and passes control to AC while retaining previous action. VC also returns *fail* if policy is violated in decision making, i.e., decision must be within the boundaries of specified benchmarks (e.g., a "Null" return should not influence action change). Control is passed to DC each time VC returns a *pass*. DC is concerned with the measure of success aspect of the rule. In this case, a TRC (Tolerance-Range-Check) is implemented: DC returns *fail* if ActionChange is more than one within the first five sample collections and subsequently if action changes at every sample instance. So DC maintains action change at maximum of one within the first five sample collections and subsequently maximum of two in any three sample instances. This will help calm any erratic behaviour that could arise. Take for instance, there could be a 360 degrees change in 'Mood' within a short space of time (e.g., a team's status in a game can change from *winning→losing→winning* within a very short space of time) which is capable of adversely affecting the choice of an ad. Figure 8 (a) and (b) are excerpts of managers of VC and DC, respectively.

The need for a new and different approach is reinforced by the capabilities exhibited in DC. It addresses situations where it's possible for overall system to fail despite process (in terms of structural, legal, syntactical, etc.) correctness.



(a)

```
If Mood <> "Null" And CostImplication <> "Null" Then
     DecisionContainer(IntervalCounter) = Mood & CostImplication
     DecisionParameter = DecisionContainer(IntervalCounter)
  '<Omitted>
  '<Omitted>
  '<Omitted>
End if
If CurrentAction(CurrentActionCounter) = CurrentAction_
(CurrentActionCounter - 1)  Then
    'CurrentAction = CurrentAction(CurrentActionCounter - 1)
   '<Omitted>
   '<Omitted>
   '<Omitted>
```

(b)

```
If IntervalCounter - IntervalCounterDC(Interval - 1) > 4 Then
    ActionChangeCounterDC = ActionChangeCounterDC + 1
  '<Omitted>
  '<Omitted>
  '<Omitted>
End if
```

Figure 8. Excerpt of VC and DC managers

In the experiment presented here, a computer program is written to simulate three autonomic managers (AC, AC+VC and AC+VC+DC). Four external variables, now referred to as context samples, (*MatchScore*, *WeatherInfo*, *TimeOfAd* and *LengthOfAd*) are fed into the managers at every sample collection instance. Sample collection instances are defined by a set time interval which can be fixed (design-time specific) or dynamically tuned. Based on policies (Figure 6), the managers decide how, when and which ad to change. The simulation was run for a total duration of 50 sample collection instances. During this duration, the managers are analysed for total number of ad changes and the distributions of those changes. For accurate analysis and comparison, the same sample at the same time instance and interval are fed into the managers concurrently. Samples may (most likely) change at every time instance and separately feeding these to the managers will lead to unbalanced judgment.

A. *Experimental Results*

Results presented are for a simulation of 50 sample collections. All three autonomic managers (AC, AC+VC and AC+VC+DC) are analysed based on number of ad changes and number of ad distributions.

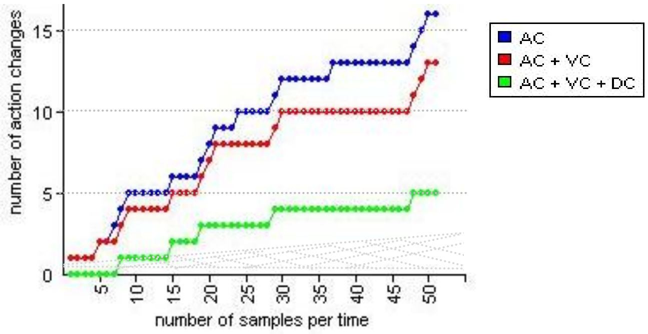


Figure 9. A sample of managers' behaviour in a 50 sample collection. (Note: If printed in black/white, the top graph is AC followed by AC+VC and then AC+VC+DC)

The optimisation of the proposed architecture in this autonomic marketing scenario is in terms of achieving balance between efficient just-in-time target-marketing decision and cost effectiveness (savings maximisation) while maintaining improved trustability and dependability in the process. Figure 9 shows the behaviour of the managers in 50 sample collections in a game duration in which the proposed architecture (AC+VC+DC) shows significant gain in stability and cost savings. It's clearly seen, for example, how (AC+VC+DC) smoothened the high fluctuation rate (high adaptability frequency) experienced between the 5th and 25th sample collections. In general, the average ad change ratio of about one change in three samples (1:3) is reduced to one change in ten samples (1:10), representing an overall gain of about 31.25% in terms of stability and cost efficiency.

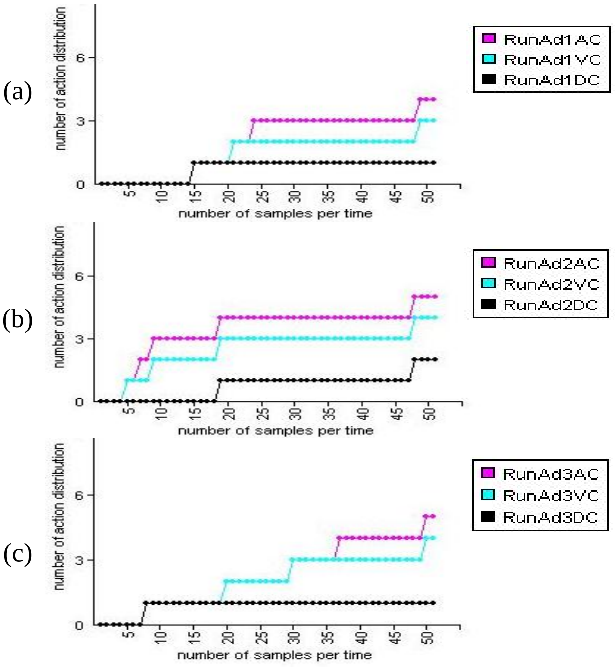


Figure 10. A distribution of the ads (Ad1, Ad2 and Ad3). (Note: If printed in black/white, the top graph is AC followed by VC and DC)

Figure 10 shows the distribution of ads across the 50 sample duration ("NullActions" i.e., 'run no ad' are not shown). This also corroborates the significant gain by the DC component. In (c), for example, only one Ad3 is run while two Ad2 are run in (b) by the (AC+VC+DC) AM. This directly translates to adaptive cost savings. Recall from Figure 5(a) that Ad2 is run when *Mood* is high and *Cost* is low (best value for money) while Ad3 is run when *Mood* and *Cost* are both high (when it costs more to run an Ad).

While it has been shown that the proposed approach is capable of maintaining reliability by reducing inefficient adaptation (cutting off unnecessary adaptations), it should be noted that reducing alone is not the answer. If the rate is very low it will not be right either. For example, if the behaviour of the manager falls within the shaded area of Figure 9, it shows that the manager is almost inactive (or not making decisions frequently enough). For every application, it is necessary to determine which rate is appropriate or cost effective in the long run. The proposed approach provides a way for tuning this (e.g., through adjusting the width of the TRC dead-zone).

There is a cost associated with bad or over frequent decisions and also a cost with not making frequent enough decisions. Success is measured by striking a balance between the two.

V. CONCLUSION

A new architecture for trustworthy autonomic systems has been presented. Different from the traditional autonomic solutions, the proposed architecture consists of mechanisms and instrumentation to support run-time self-validation and trustworthiness. At the core of the architecture are three components, the AutonomicController, ValidationCheck and DependabilityCheck, which allow developers specify controls and processes to improve system trustability. An analysis of the current state of practice in autonomic architecture shows that a new approach is required in which validation and trustworthiness are not treated as add-ons as they cannot be reliably retro-fitted to systems. Validation alone does not always guarantee trustworthiness as logical processes/actions could sometimes lead to overall system instability. There are situations where, for example, despite the autonomic manager's legitimate decisions within the logical boundaries of specified rules, there's the possibility of overall erratic behaviour or inconsistency in the behaviour of the system. This is why autonomic systems need a new approach.

To demonstrate the feasibility and practicability of our approach, a case example scenario has been presented. The case scenario demonstrates how the proposed architecture can maximise cost, improve trustability and efficient target marketing in a company-centric Autonomic Marketing System that has many dimensions of freedom and is sensitive to a number of contextual volatility. As this approach is new, future research work will focus on improving the robustness of the proposed architecture. This includes adding a predictive/learning sub-component to the DependabilityCheck component and verifying how results of this approach can vary in other contexts to see which factors could influence its adoption or not in practice.

ACKNOWLEDGEMENT: Graphs are plotted using csXGraph ActiveX / OCX control by Chestysoft [10].

REFERENCES

[1] IBM, *An architectural blueprint for autonomic computing*, IBM Whitepaper, 2004

[2] Kephart Kephart and David Chess, *The Vision of Autonomic Computing*. Computer, IEEE, Volume 36, Issue 1, January 2003, pp. 41-50

[3] Andrew Diniz, Viviane Torres, and Carlos José, *A Self-adaptive Process that Incorporates a Self-test Activity*, Monografias em Ciência da Computação, No. 32/09, Rio De Janeiro – Brasil, Nov. 2009.

[4] Xiaolin Li, Hui Kang, Patrick Harrington, and Johnson Thomas, *Autonomic and trusted computing paradigms*, In Proceedings of ATC'2006, pp. 143-152

[5] Tariq King, Djuradj Babich, Jonatan Alava, Peter Clarke, and Ronald Stevens, *Towards Self-Testing in Autonomic Computing Systems*, Proceedings of the Eighth International Symposium on Autonomous Decentralized Systems (ISADS'07), Arizona, USA, 2007

[6] Tariq King, Alain Ramirez, Peter Clarke, and Barbara Quinones-Morales, *A Reusable ObjectOriented Design to Support SelfTestable Autonomic Software*, Proceedings of the 2008 ACM symposium on Applied computing, Fortaleza, Ceara, Brazil, 2008, pp. 1664-1669

[7] Stuart Anderson, Mark Hartswood, Rob Procter, Mark Rouncefield, Roger Slack, James Soutter, and Alex Voss, *Making Autonomic Computing Systems Accountable*, Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA), 2003

[8] Richard Anthony, *Policy-based autonomic computing with integral support for self-stabilisation*, Int. Journal of Autonomic Computing, Vol. 1, No. 1, pp. 1–33. 2009

[9] Carl Adams, Richard Anthony, Wendy Powley, David Bell, Chris White, and Chun Wu, *Towards Autonomic Marketing*, The 8th International Conference on Autonomic and Autonomous Systems (ICAS), pp. 28-31, St. Maarten 2012.

[10] Chestysoft csXGraph:www.chestysoft.com/xgraph/instructions.pdf Last accessed date 29th June 2012.

[11] Thaddeus Eze, Richard Anthony, Chris Walshaw, and Alan Soper, *Autonomic Computing in the First Decade: Trends and Direction*, The 8th International Conference on Autonomic and Autonomous Systems (ICAS), pp. 80-85. St. Maarten 2012.

[12] Richard Anthony, *Policy-centric Integration and Dynamic Composition of Autonomic Computing Techniques*, The 4th International Conference on Autonomic Computing (ICAC), 2007, Florida, USA

[13] Markus Huebscher and Julie McCann, *A survey of autonomic computing—degrees, models, and applications*, ACM Computer Survey, 40, 3, Article 7 (August 2008)

[14] Christoph Reich, Kris Bubendorfer, and Rajkumar Buyya, *An autonomic peer-to-peer architecture for hosting stateful web services*, The 8th IEEE International Symposium on Cluster Computing and the Grid (CCGRID 08), pp. 250-257, 2008.

[15] Fang Mei, Yanheng Liu, Hui Kang, and Shuangshuang Zhang, *Policy-based autonomic mobile network resource management architecture*, The 2nd International Symposium on Networking and Network Security (ISNNS 10), pp. 144-148, April 2010.

[16] Joao Ferreira, Joao Leitao, and Luis Rodrigues, *A-osgi: A framework to support the construction of autonomic osgi-based applications*, Technical Report RT/33/2009, May 2009.

[17] Haffiz Shuaib, Richard Anthony, and Mariusz Pelc, *A Framework for Certifying Autonomic Computing Systems*, The 7th International Conference on Autonomic and Autonomous Systems (ICAS), pp. 122-127, 2011, Venice, Italy

[18] Ting-Jung Yu, Robert Lai, Menq-Wen Lin, and Bo-Rue Kao, *A Fuzzy Constraint-Directed Autonomous Learning to Support Agent Negotiation*, The 3rd International Conference on Autonomic and Autonomous Systems (ICAS), pp. 28, 2007, Athens, Greece

[19] Han Li and Srikumar Venugopal, *Using Reinforcement Learning for Controlling an Elastic Web Application Hosting Platform*, The 8th International Conference on Autonomic Computing (ICAC), pp. 205-208, 2011, Karlsruhe, Germany

Quality Assessment for Recognition Tasks (QART)

Mikołaj Leszczuk
AGH University of Science and Technology
Department of Telecommunications
Kraków, Poland
Email: leszczuk@agh.edu.pl

Joel Dumke
National Telecommunications and Information Administration
Institute for Telecommunication Sciences
Boulder CO, USA
Email: jdumke@its.bldrdoc.gov

***Abstract*—Users of video to perform tasks require sufficient video quality to recognize the information needed for their application. Therefore, the fundamental measure of video quality in these applications is the success rate of these recognition tasks, which is referred to as visual intelligibility or acuity. One of the major causes of reduction of visual intelligibility is loss of data through various forms of compression. Additionally, the characteristics of the scene being captured have a direct effect on visual intelligibility and on the performance of a compression operation-specifically, the size of the target of interest, the lighting conditions, and the temporal complexity of the scene. This paper presents a Work in Progress (WIP) Quality Assessment for Recognition Tasks (QART) project, which is performing a series of tests to study the effects and interactions of compression and scene characteristics. An additional goal is to test existing or develop new objective measurements that will predict the results of the subjective tests of visual intelligibility.**

***Index Terms*—Video; compression; MOS (Mean Opinion Score), WIP, QART**

I. Introduction

The transmission and analysis of video is often used for a variety of applications outside the entertainment sector, and generally this class of (task-based) video is used to perform a specific recognition task. Examples of these applications include security, public safety, remote command and control, tele-medicine, and sign language. The Quality of Experience (QoE) concept for video content used for entertainment differs materially from the QoE of video used for recognition tasks because in the latter case, the subjective satisfaction of the user depends upon achieving the given task, e.g., event detection or object recognition. Additionally, the quality of video used by a human observer is largely separate from the objective video quality useful in computer vision [1]. Therefore, it is crucial to measure and ultimately optimize task-based video quality. This is discussed in more detail in [2].

Enormous work, mainly driven by the Video Quality Experts Group (VQEG) [3], has been carried out for the past several years in the area of consumer video quality. The VQEG is a group of experts from various backgrounds and affiliations, including participants from several internationally recognized organizations, working in the field of video quality assessment. The group was formed in October of 1997 at a meeting of video quality experts. The majority of participants are active in the International Telecommunication Union (ITU) and VQEG combines the expertise and resources found in several ITU Study Groups to work towards a common goal [3]. Unfortunately, many of the VQEG and ITU methods and recommendations (like ITU's Absolute Category Rating – ACR – described in ITU-T P.800 [4]) are not appropriate for the type of testing and research that task-based video, including CCTV, requires.

This paper is organized as follows. Section II describes related work and motivation. In Section III, the QART Project and standardisation is discussed. Section IV concludes the paper and details the future work.

II. Related Work and Motivation

Some subjective recognition metrics, described below, have been proposed over the past decade. They usually combine aspects of Quality of Recognition (QoR) and QoE. These metrics have been not focused on practitioners as subjects, but rather on naïve participants. The metrics are not context specific, and they do not apply video surveillance-oriented standardized discrimination levels.

One of the metrics being definitively worth to mention is Ghinea's Quality of Perception (QoP) [5], [6]. However, the QoP metric does not entirely fit video surveillance needs. It targets mainly video deterioration caused by frame rate (fps), whereas fps does not necessarily affect the quality of Closed-Circuit Tele-Vision (CCTV) and the required bandwidth [7]. The metric has been established for rather low, legacy resolutions, and tested on rather small groups of subjects (10 instead of standardized 24 valid, correlating subjects). Furthermore, a video recognition quality metric for a clear objective of video surveillance context requires tests in fully controlled environment [8], with standardized discrimination levels (avoiding ambiguous questions) and with minimized impact of subliminal cues [9].

Another metric being worth to mention is QoP's offshoot, Strohmeier's Open Profiling of Quality (OPQ) [10]. This metric puts more stress on video quality than on recognition/discrimination levels. Its application context, being focused on 3D, is also different than video surveillance which requires rather 2D. Like the previous metric, this one also does not apply standardized discrimination levels, allowing subjects to use their own vocabulary. The approach is qualitative rather than quantitative, whereas the latter is preferred by public safety practitioners for, e.g., public procurement. The OPQ model is somewhat content/subject-oriented, while a more generalized metric framework is needed for video surveillance.

ISBN: 978-1-61208-239-4

OPQ partly utilizes free sorting, as used in [11], but also applied in the method called Interpretation Based Quality (IBQ) [12], [13], adapted from [14], [15]. Unfortunately, these approaches allow mapping relational, rather than absolute, quality.

Furthermore, there exists only a very limited set of quality standards for task-based video applications. Therefore, it is still necessary to define the requirements for such systems from the camera, to broadcast, to display. The nature of these requirements will depend on the task being performed.

European Norm №. 50132 [16] was created to ensure that European CCTV systems are realized under the same rules and requirements. The existence of a standard has opened an international market of CCTV devices and technologies. By selecting components that are consistent with the standard, a user can achieve a properly working CCTV system. This technical regulation deals with different parts of a CCTV system including acquisition, transmission, storage, and playback of surveillance video. The standard consists of such sections as lenses, cameras, local and main control units, monitors, recording and hard copy equipment, video transmission, video motion detection equipment, and ancillary equipment. This norm is hardware-oriented as it is intended to unify European law in this field; thus, it does not define the quality of video from the point of view of recognition tasks.

The Video Quality in Public Safety (VQiPS) Working Group, established in 2009 and supported by the U.S. Department of Homeland Securitys Office for Interoperability and Compatibility, has been developing a user guide for public safety video applications. The goal of the guide is to provide potential public safety video customers with links to research and specifications that best fit their particular application, as such research and specifications become available. The process of developing the guide will have the desired secondary effect of identifying areas in which adequate research has not yet been conducted, so that such gaps may be filled. A challenge for this particular work is ensuring that it is understandable to customers within public safety, who may have little knowledge of video technology.

The approach taken by VQiPS is to remain application-agnostic. Instead of attempting to individually address each of the many public safety video applications, the guide is based on their common features. Most importantly, as mentioned above, each application consists of some type of recognition task. The ability to achieve a recognition task is influenced by many parameters, and five of them have been selected as being of particular importance. They are:

- **Usage time-frame.** Specifies whether the video will need to be analysed in real-time or recorded for later analysis.
- **Discrimination level.** Specifies the level of detail required from the video.
- **Target size.** Specifies whether the anticipated region of interest in the video occupies a relatively small or large percentage of the frame.
- **Lighting level.** Specifies the anticipated lighting level of the scene.
- **Level of motion.** Specifies the anticipated level of motion in the scene.

These parameters form what are referred to as Generalized Use Classes, or GUCs [17]. Fig. 1 is a representation of the GUC determination process.

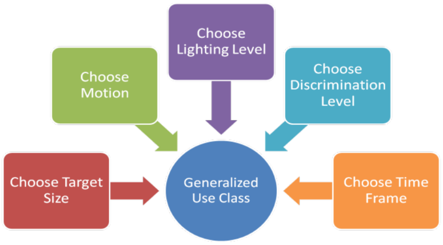


Fig. 1. Classification of video into GUC as proposed by VQiPS (source: [1]).

To develop accurate objective measurements and models for video quality assessment, subjective tests (psychophysical experiments) must be performed. The ITU has recommendations that address the methodology for performing subjective tests in a rigorous manner [8], [18]. These methods are targeted at the entertainment application of video and were developed to assess a person's perceptual opinion of quality. They are not entirely appropriate for task-based applications, in which video is used to recognize objects, people or events.

Assessment principles for the maximization of task-based video quality are a relatively new field. Problems of quality measurements for task-based video are partially addressed in a few preliminary standards and a Recommendation ITU-T P.912 [9], [19] that mainly introduce basic definitions, methods of testing and psycho-physical experiments. ITU-T P.912 describes multiple choice, single answer, and timed task subjective test methods, as well as the distinction between real-time and viewer-controlled viewing, and the concept of scenario groups to be used for these types of tests. Scenario groups are groups of very similar scenes with only small, controlled differences between them, which enable testing recognition ability while eliminating or greatly reducing the potential effect of scene memorization. While these concepts have been introduced specifically for task-based video applications in ITU-T P.912, more research is necessary to validate the methods and refine the data analysis.

III. The QART Project and Standardisation

Internationally, the number of people and organizations interested in this area continues to grow, and there is currently enough interest to motivate the creation of a task-based video project under VQEG. At one of the recent meetings of VQEG, a new project was formed for task-based video quality research. The Quality Assessment for Recognition Tasks (QART) project addresses precisely the problem of lack of quality standards for video monitoring [20]. The initiative is co-chaired by Public Safety Communications Research (PSCR) program, U.S.A., and AGH University of Science and Technology in Krakow, Poland. Other members include research teams from Belgium, France, Germany, and South Korea. The purpose of QART is exactly the same as the other VQEG projects – to advance the field of quality assessment for task-based video through collaboration in the development of test methods, performance specifications and standards for task-based video, as well as predictive models based on network and other relevant parameters [21].

There has been some QART work conducted so far. The research has answered the practical problem of a network link with a limited bandwidth and detection probability is an interesting parameter to find. QART have presented the results of the development of critical quality thresholds in licence plate recognition by human subjects, based on a video streamed in constrained networking conditions. Many video sequences originated from the database of the Consumer Digital Video Library (CDVL) [22]. QART have shown that, for a particular view, a model of detection probability based on bit rate can work well. Nevertheless, different views have very different effects on the results obtained. QART have also learned that for these kinds of psycho-physical experiments, licence plate characteristics (such as illumination) are of great importance, sometimes even prevailing over the distortions caused by bit-rate limitations and compression [23].

One important conclusion is that for a bit rate as low as 180 kbit/s the detection probability is over 80% even if the visual quality of the video is very low. Moreover, the detection probability depends strongly on the Source Reference Channel/Circuit. (SRC, over all detection probability varies from 0 (sic!) to over 90%) [23].

Furthermore, a study of the ability to recognize a moving or stationary object given several lighting and target size combinations, and a study of license plate recognition, both processed at a number of compression rates, have been completed. These are the first in a planned series of studies with the similar goal of studying the ability to recognize objects given various network conditions [1].

Recently, a subjective test has been completed, consisting of various levels of compression and resolution reduction following the methods suggested in ITU-T P.912 and the VQiPS GUCs [3]. The test method was the multiple choice method. Bit-rates from 64 kbit/s to 1536 kbit/s using H.264 encoding were studied, in combination with either VGA or CIF resolution. A total of 10 bit-rate/resolution combinations were tested. The recognition task for the viewer was the identification of an object within a simulated real-time environment (i.e., pausing or replaying the video was not allowed.) An example of the user interface is shown in Fig. 2.

The objects were either stationary or moving, and were filmed under three lighting conditions and at two distances from the camera. The test results thus can be categorized into several of the GUCs. Results were presented as recognition rates; in other words, the percentage of objects correctly identified (after normalization for guessing). Recognition rates of 90% and 50% were chosen as significant thresholds for which recommendations were suggested based on test results.

The accuracy of answers given by subjects was growing during the test. It suggests that subjects were aided by memory effects as the test progressed.

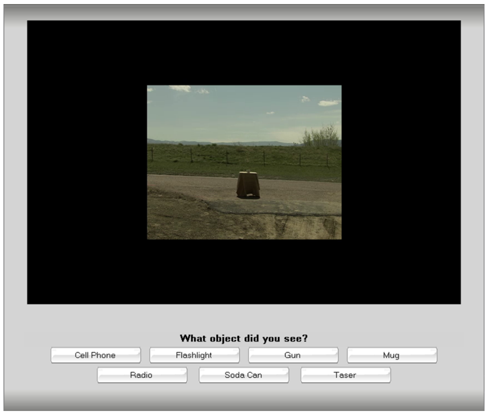


Fig. 2. User interface for subjective target recognition task test (source: [1]).

Finally, QART recently completed a test of subjects' ability to recognize car registration numbers in video material recorded using a CCTV camera and compressed with the H.264/AVC codec [2].

A subjective experiment was carried out in order to perform the analysis. A psycho-physical evaluation of the video sequences scaled in the compression or spatial domain at various bit-rates was performed. The aim of the subjective experiment was to gather the results of human recognition capabilities. Thirty non-expert testers rated video sequences influenced by different compression parameters. ITUs Single Stimulus (SS) described in ITU-R BT.500-11 [8], was selected as the subjective test methodology [2].

The recognition task was threefold: 1) type in the licence plate number, 2) select car colour, and 3) select car make. Testers were allowed to control playback and enter full screen mode. A more detailed description of the recognition task is available in [2].

The tests were conducted using a web-based interface connected to a database. In the database both information about the video samples and the answers received from the testers were gathered. An example of the user interface is shown in Fig. 3.

Video sequences used in the test were recorded in a car park using a CCTV camera. The H.264 codec with x264 implementation was selected as the reference as it is a modern, open, and widely used solution. Video compression parameters were adjusted in order to cover the recognition ability threshold. The compression was done with the bit-rate ranging from 40 kbit/s to 440 kbit/s [2].

The testers who participated in this study provided a total of 960 answers. Each answer could be interpreted as the number of per-character errors, i.e., zero errors meaning correct recognition. The average probability of a license plate being identified correctly was 54.8% with 526 recognitions out of 960, 64.1% recognitions had no more than one error, and 72% of all characters were recognized [2].

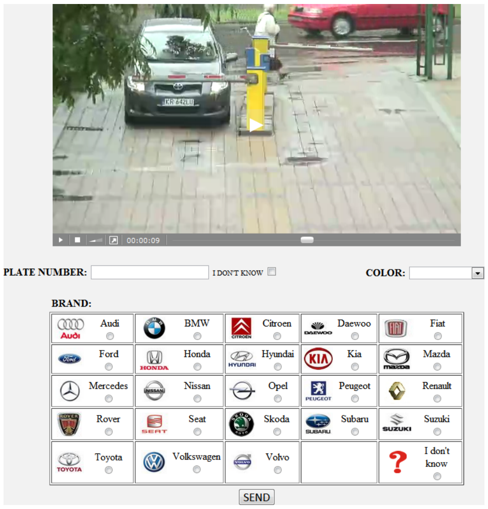


Fig. 3. User interface for subjective plate recognition task test (source: [23]).

IV. Conclusion and Future Work

In summary, QART introduced contributions to the field of task-based video quality assessment methodologies: from subjective psycho-physical experiments to objective quality models. The developed methodologies are just a single contribution to the overall framework of quality standards for task-based video. It is necessary to further define requirements starting from the camera, through the broadcast, and until after the presentation. These requirements will depend on the particular tasks users wish to perform [2]. Future work will include, e.g., quantification of GUC and extension of P.912.

Acknowledgments

For the research leading to these results, Mikołaj Leszczuk would like to thank the European Community's Seventh Framework Program (FP7/2007-2013) for received funding under grant agreement №. 218086 (INDECT) as well as to thank AGH University of Science and Technology for received funding under Department of Telecommunications statutory work. Joel Dumke would like to thank the U.S. Department of Commerce for received funding for the research under the Public Safety Communications Research project.

References

[1] M. Leszczuk, I. Stange, and C. Ford, "Determining image quality requirements for recognition tasks in generalized public safety video applications: Definitions, testing, standardization, and current trends," in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2011, pp. 1–5.

[2] M. Leszczuk, "Assessing task-based video quality   a journey from subjective psycho-physical experiments to objective quality models," in *Multimedia Communications, Services and Security*, ser. Communications in Computer and Information Science, A. Dziech and A. Czyżewski, Eds. Springer Berlin Heidelberg, 2011, vol. 149, pp. 91–99. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-21512-4_11

[3] *The Video Quality Experts Group*, VQEG, July 2012, http://www.vqeg.org/.

[4] *ITU-T P.800, Methods for subjective determination of transmission quality*, International Telecommunication Union Recommendation, 1996. [Online]. Available: http://www.itu.int/rec/T-REC-P.800-199608-I

[5] G. Ghinea and J. P. Thomas, "Qos impact on user perception and understanding of multimedia video clips," in *Proceedings of the sixth ACM international conference on Multimedia*, ser. MULTIMEDIA '98. New York, NY, USA: ACM, 1998, pp. 49–54. [Online]. Available: http://doi.acm.org/10.1145/290747.290754

[6] G. Ghinea and S. Y. Chen, "Measuring quality of perception in distributed multimedia: Verbalizers vs. imagers," *Computers in Human Behavior*, vol. 24, no. 4, pp. 1317–1329, 2008.

[7] L. Janowski and P. Romaniak, "Qoe as a function of frame rate and resolution changes," in *Future Multimedia Networking*, ser. Lecture Notes in Computer Science, S. Zeadally, E. Cerqueira, M. Curado, and M. Leszczuk, Eds. Springer Berlin / Heidelberg, 2010, vol. 6157, pp. 34–45. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13789-1_4

[8] *ITU-R BT.500-12, Methodology for the subjective assessment of the quality of television pictures*, International Telecommunication Union Recommendation, Rev. 12, 2009. [Online]. Available: http://www.itu.int/rec/R-REC-BT.500-12-200909-I

[9] *ITU-T P.912, Subjective video quality assessment methods for recognition tasks*, International Telecommunication Union Recommendation, 2008. [Online]. Available: http://www.itu.int/rec/T-REC-P.912-200808-I

[10] D. Strohmeier, S. Jumisko-Pyykkö, and K. Kunze, "Open profiling of quality: a mixed method approach to understanding multimodal quality perception," *Adv. MultiMedia*, vol. 2010, pp. 3:1–3:17, January 2010. [Online]. Available: http://dx.doi.org/10.1155/2010/658980

[11] M. Duplaga, M. Leszczuk, Z. Papir, and A. Przelaskowski, "Evaluation of quality retaining diagnostic credibility for surgery video recordings," in *Proceedings of the 10th international conference on Visual Information Systems: Web-Based Visual Information Search and Management*, ser. VISUAL '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 227–230.

[12] J. Radun, T. Leisti, J. Hakkinen, H. Ojanen, J. L. Olives, T. Vuori, and G. Nyman, "Content and quality: Interpretation-based estimation of image quality," *ACM Transactions on Applied Perception*, vol. 4, no. 4, pp. 2:1–2:15, 2008. [Online]. Available: http://doi.acm.org/10.1145/1278760.1278762

[13] G. Nyman, J. Radun, T. Leisti, J. Oja, H. Ojanen, J. L. Olives, T. Vuori, and J. Hakkinen, "What do users really perceive — probing the subjective image quality experience," in *Proceedings of the SPIE International Symposium on Electronic Imaging 2006: Imaging Quality and System Performance III, Vol. 6059*, 2006, pp. 1–7.

[14] P. Faye, D. Bremaud, M. D. Daubin, P. Courcoux, A. Giboreau, and H. Nicod, "Perceptive free sorting and verbalisation tasks with naive subjects: an alternative to descriptive mappings," *Food Quality and Preference*, vol. 15, no. 7-8, pp. 781 – 791, 2004, fifth Rose Marie Pangborn Sensory Science Symposium. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950329304000540

[15] D. Picard, C. Dacremont, D. Valentin, and A. Giboreau, "Perceptual dimensions of tactile textures," *Acta Psychologica*, vol. 114, no. 2, pp. 165–184, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0001691803000751

[16] *CENELEC EN 50132, Alarm systems. CCTV surveillance systems for use in security applications.*, European Committee for Electrotechnical Standardization European Norm, 2011.

[17] VQiPS, "Video quality tests for object recognition applications," U.S. Department of Homeland Security's Office for Interoperability and Compatibility, June 2011. [Online]. Available: http://www.safecomprogram.gov/SAFECOM/library/technology/1627_additionalstatement.htm

[18] *ITU-T P.910, Subjective video quality assessment methods for multimedia applications*, International Telecommunication Union Recommendation, 1999. [Online]. Available: http://www.itu.int/rec/T-REC-P.910-200804-I

[19] C. G. Ford, M. A. McFarland, and I. W. Stange, "Subjective video quality assessment methods for recognition tasks," *Human Vision and Electronic Imaging XIV*, vol. 7240, no. 1, p. 72400Z, 2009. [Online]. Available: http://link.aip.org/link/?PSI/7240/72400Z/1

[20] M. Leszczuk and J. Dumke, *The Quality Assessment for Recognition Tasks (QART)*, VQEG, July 2012, http://www.its.bldrdoc.gov/vqeg/project-pages/qart/qart.aspx.

[21] P. Szczuko, P. Romaniak, M. Leszczuk, R. Mirek, M. Pleva, S. Ondas, G. Szwoch, P. Korus, C. Kollmitzer, P. Dalka, J. Kotus, A. Ciarkowski, A. Dąbrowski, P. Pawłowski, T. Marciniak, R. Weychan, and F. Misiorek, "D1.2, report on ns and cs hardware construction," The INDECT Consortium: Intelligent Information System Supporting Observation,

Searching and Detection for Security of Citizens in Urban Environment, European Seventh Framework Programme FP7-218086-collaborative project, Europa, Tech. Rep., 2010, cop.

[22] *The Consumer Digital Video Library*, CDVL, July 2012, http://www.cdvl.org/.

[23] M. Leszczuk, L. Janowski, P. Romaniak, A. Głowacz, and R. Mirek, “Quality assessment for a license plate recognition task based on a video streamed in limited networking conditions,” in *4th International Conference on Multimedia Communications, Services and Security*, ser. Communications in Computer and Information Science, A. Dziech and A. Czyżewski, Eds., vol. 149. Krakow, Poland: Springer Berlin Heidelberg, 2-3 June 2011, pp. 10–18. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-21512-4_2

Development of Remote Support Service by Augmented Reality Videophone

Atsushi Fukayama, Shunsuke Takamiya, Junichi Nakagawa, Shinyo Muto, and Naoki Uchida
NTT Service Evolution Laboratories
Nippon Telegraph and Telephone Corporation
1-1, Hikari-no-oka, Yokosuka-shi, Kanagawa, Japan
{fukayama.atsushi, takamiya.shunsuke, nakagawa.junichi, muto.shinyo, uchida.naoki}@lab.ntt.co.jp

***Abstract*—This paper presents the development of a remote support service based on an existing videophone system enhanced with AR (Augmented Reality) technology. The service enables a support person to point to a real object in a remote site by overlaying a virtual object, which is named as “air stamp,” onto video image. Besides, the air stamp stays on the object it was attached to at first based on image-based AR technology, even when a camera of mobile device in the remote site is moved. These features make the communication between the support person and the onsite worker much more efficient. Video latency and effective frame rate of the videophone image were compared between the original videophone system and AR-enabled system, which indicated the addition of the AR function did not affect the performance significantly.**

Keywords-videophone; augmented reality; network value-added service; remote support; telepointing.

I. INTRODUCTION

The recent spread of fixed and mobile broadband networks and dissemination of smart telecommunication devices facilitates the people’s easy access to videophone services. These conditions could launch a rapid rise of videophone service use, finally after communication service providers’ long struggle for it. But, it seems to be a little further to come.

One possible way to encourage the videophone use is offering new use cases by extending basic videophone functionality. Ordinary videophone services just transfer live audio and video media back and forth as it is captured. But, recent improvement of media processing technologies [1][2] allows real-time and real-world media processing which is needed to enhance plain videophone service.

For instance, an acoustic speech recognition technology estimates “who speaks when and what” based on audio media in real time in natural multi-party meeting configuration [1]. A face recognition technology can continuously detects registered person’s face in live video image under various types of situation change such as face posture and illumination [2].

By combining with the wide variety of media processing technologies, a single videophone service can match wide variety of use cases. But, each use case is not as general as ones covered by traditional basic videophone services. This means that the expansion of existing videophone service must be realized in low cost.

Against these issues, we propose a service architecture to enhance existing videophone service with Augmented Reality (AR) technology [3] that detects real objects in videophone image and overlays related virtual objects on the real objects. This architecture facilitates the low-cost videophone service expansion by enabling reuse of their existing asset such as Multi-point Connecting Unit (MCU) and client software on terminal devices [4].

As an application of our AR videophone service architecture, we developed a remote support service by which an instructor can place a virtual object, which we call as “air stamp,” onto a real object being shot in video image. The air stamp can be used to convey what the instructor talked about to a remote onsite worker. Based on image recognition technology, the air stamp stays on the object it firstly attached to even when a camera of mobile device that captures the remote site scene is moved, which permits free movement of the onsite worker who is holding the device.

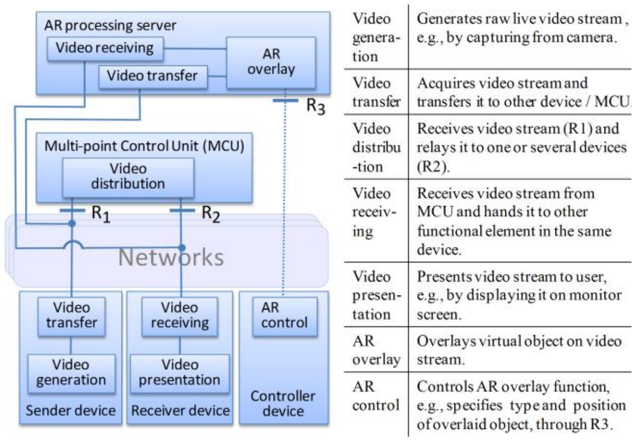
In this report, after introducing related works in Section II, design of the remote support service based on collected requirements is discussed in Section III. Performance evaluation result is shown in Section IV to confirm that AR function added on base videophone system does not degrade system performance.

II. RELATED WORK

An example of the efforts the telecommunication industry has been making to realize media-enhanced network value-added services was recently seen in the Rich Communications Services (RCS) program enacted by the GSM Association (GSMA). In the program, they are exploring enriched communication services other than mere voice communication and trying to standardize technical realization [5]. Proposed examples of network value-added services include enriched content sharing, in which pictures shared between RCS-enabled devices are converted in the IMS network, and enriched chat with text conversion services such as language translation. The Telecommunication Technology Committee (TTC) is discussing technical realization with respect to network value-added media services. It has proposed an architecture in which an Application Server (AS) for network value-added services intervenes the media flow between a device and other AS dedicated for existing services [6].

Intelligent media processing services combined with telecommunication services are about to appear from the telecom industry. For example, NTT DoCoMo announced and demonstrated an automatic live translation service available during mobile telephone conversation, which

ISBN: 978-1-61208-239-4



Video genera-tion	Generates raw live video stream , e.g., by capturing from camera.
Video transfer	Acquires video stream and transfers it to other device / MCU.
Video distribu -tion	Receives video stream (R1) and relays it to one or several devices (R2).
Video receiv-ing	Receives video stream from MCU and hands it to other functional element in the same device.
Video presen-tation	Presents video stream to user, e.g., by displaying it on monitor screen.
AR overlay	Overlays virtual object on video stream.
AR control	Controls AR overlay function, e.g., specifies type and position of overlaid object, through R3.

Figure 1. Basic architecture of AR videophone service.

performs speech recognition, translation, and synthesis on a network cloud [7]. This prototype service enhances the audio medium of voice telephone in a network. Our study proposes an enhancement of visual media in network cloud.

As one of many visual media processing technologies, AR has been the subject of a huge number of references in the literature, including some that applied AR to teleconferencing [8][9] and remote collaboration scenarios [10]. Historically, many AR-based conferencing and collaboration studies have employed specialized devices such as head mounted displays (HMDs). Against the background of recent trends in commoditized Internet video-chat and smart devices, an attempt was made to combine a client-based AR system with a video-chat system [9]. Defining our study from this viewpoint, we applied a server-based AR technology to existing managed videophone service.

In the Internet service domain, we can find some communication services that incorporate media processing technologies in the middle of communication channels. Google Talk enhancement with language translation functionality seems to match with our concept [11]. The translation functionality in the text chat system is implemented as translation “bot,” which is apparently same as human chat participants but actually an autonomous agent automatically responding to others. The translation bot listens to other participant’s words, translates the words, and utters the translated words. Addition of this translation functionality does not affect the base text-chat system at all. It can lower deployment cost for the service provider and entry barrier for users.

III. SYSTEM DESIGN

A. *Basic Architecture*

Figure 1 depicts the architecture of AR videophone service we proposed [4]. Based on this architecture, the AR processing server connects with other system components in the same way as normal terminal device. This can minimize system modification for inserting the AR functionality in the middle of video image transfer channel.

The architecture comprises three types of terminal devices, a Multi−point Control Unit (MCU), and an AR processing server. Although devices are categorized into sender, receiver, and controller in terms of device function, one implemented device can belong to more than one functional category. For instance, ordinary videophone device has both functions of sender and receiver.

B. *Service Concept*

Figure 2 shows core functionality of our remote support service based on the AR videophone architecture [4]. The target service of this system is remote support, where instructors in a support center give instructions to support their customers or onsite workers. At the remote site, a video image is taken using a mobile device, e.g., smart-phone and tablet, and shared with the instructor. The instructor can put virtual objects, i.e., air stamps, onto actual objects seen in the video so that the user in the remote site can understand what the instructor is talking about. The stamps must be stuck to the actual objects because the remote user often moves the mobile camera.

In the previous report [4], we already proposed the concept of this remote support service and presented its prototype. Development of a practical system following the prototype is introduced in this report.

C. *Requirements*

By using the prototype visualizing our service concept, we interviewed and derived requirements to the remote support service from people working in related domains such as Information Technology (IT) maintenance and manufacturing as well as customer support.

1) *Number of devices:* Many instructors should be involved in some cases such as failure analysis of complicated system. Therefore, several instructor devices should be able to join one session. Some installation works of networked systems require cooperation between onsite workers in different sites. To support this type of work, two workers should be able to join one same session and show situation in their working sites to instructors in turn.

2) *Media quality:* In most remote support use cases, high frame rate is not required because the objects to work on are static. Video image resolution that is enough for instructors in remote support sites to recognize printed

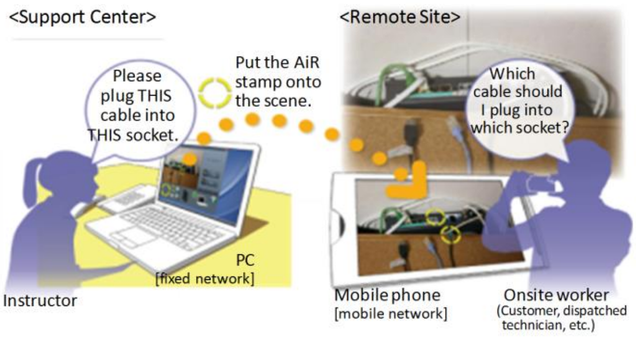

<Support Center>
<Remote Site>
Please plug THIS cable into THIS socket.
Put the AiR stamp onto the scene.
Which cable should I plug into which socket?
PC
[fixed network]
Instructor
Mobile phone
[mobile network]
Onsite worker
(Customer, dispatched technician, etc.)

Figure 2. Remote support service based on AR videophone.

ISBN: 978-1-61208-239-4

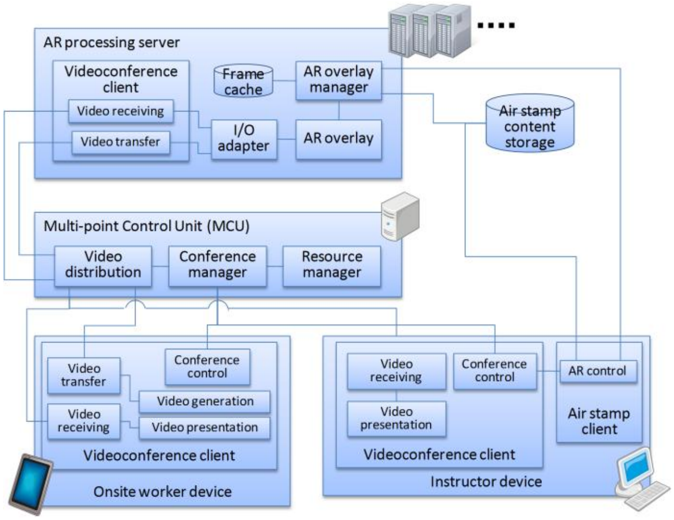


Figure 3. Developed remote support system diagram.

letters on working object is required. As it is assumed that onsite worker uses mobile device (Figure 2), it would be sufficient if the letters can be read when they are captured in close-up shot. Clear voice is required for efficient remote support communication. Though the air stamps can facilitate smooth communication, it would play a complementary role with verbal instruction.

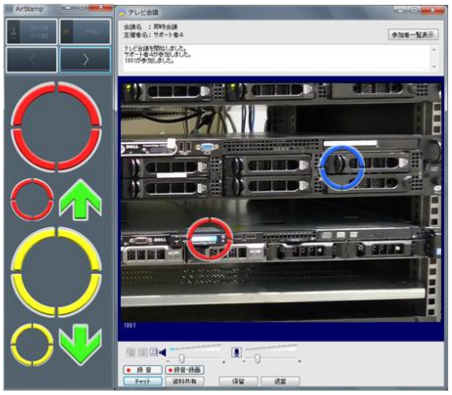
3) *Onsite network environment:* Wide range of network access for the onsite mobile device is required. For working sites in outside and third-party's premise, 3G / LTE mobile access is required. In one's own premise, WiFi is sufficient. Wired access is required when use of wireless communication is forbidden for security reason or to avoid possible harm of radio wave on electronic equipment.

4) *Additional functions:* Many additional functions supplementary with the core air-stamp functionality were found. Recording of audio / video stream, document sharing, and text messaging were common requirements.

D. Implementation

Figure 3 shows system diagram of developed remote support system.

We deployed a commercial videoconferencing system for MCU and video generation / transfer / receiving / presentation functions shown in Figure 1. Analyzing the requirements about number of devices and additional functions, we found that many high-end or middle-class videoconferencing systems already meet those requirements.



会議名　：即時会議
主催者名：サポート者4

テレビ会議を開始しました。
サポート者4が参加しました。
1001が参加しました。

Figure 4. User interface on instructor device.

Considering the media quality requirements and onsite network environment, we configured video resolution to VGA (640x480 pixels), which a preliminary test indicated was the bottom line to allow users to read letters in the video image. It was expected that VGA would spend too much bandwidth for applying to 3G mobile access with standard frame rate such as 15 or 30 frames per second (fps). In that case, we can adjust the frame rate to lower values.

The videoconference client on the onsite-worker and instructor devices is the standard client software of the deployed videoconferencing system. Figure 4 shows user interface on the instructor computer, which consists of the standard videoconference client on the right side and an air stamp client on the left side.

When a user sets up a normal videoconference, the user requests the conference manager function through the conference control function in the client software. The conference manager function controls the video distribution function to make a call to devices of conference participants including the requesting party.

In the case the user requests an AR-enabled videoconference, the conference manager inquires available AR processing server to the resource manager function and obtain its address. Then, a conference including the participants and the available AR server is established.

After a videoconference is set up, raw video stream is transferred to the AR processing server through the MCU. Raw video frames decoded by the client software in the server are input to the AR overlay function through the Input / Output (I/O) adapter function which is needed when I/O interface of deployed AR technology and videoconference client does not match.

The AR overlay function searches specified reference images in the input video image and overlay an air stamp on each discovered image. Because this processing is quite general among image-based AR technologies, various AR technologies can be used for the AR overlay function.

The AR overlay manager function receives information about instructor's operation to put an air stamp on a sub-region in the video image from the AR control function. The reference image is extracted from raw input video image based on the sub-region information, which is passed to the AR overlay function together with the air stamp image.

Video transfer delay prevents the AR overlay function from making the reference image from the right frame of the right timing. The isolation between the base videoconferencing system and AR functionality added on the system makes it difficult to use timestamp information to compensate the delay, though it brings the reusability of the base system. To solve this problem, the AR overlay function caches raw input frames during the user's air-stamp operation and the AR control sends the frame image when the air stamp is stamped, which enables the AR overlay function to retrieve the right input frame from the cache.

ISBN: 978-1-61208-239-4

IV. EVALUATION

To evaluate possible performance degradation caused by insertion of the AR processing server in the middle of video source and destination, we compared video delay and effective frame rate between AR-enabled conference and standard video-only conference. Evaluation was performed in local network environment consisting of WiFi for onsite worker device's access and wired gigabit Ethernet for other part. AR processing server, MCU, and instructor device were built on a desktop computer with Intel Core-i processor and Windows 7. A mobile tablet with 1.0GHz ARM Cortex-A8 processor and Android 2.3 was used for onsite worker device.

The results of video delay comparison are shown in Figure 5. The video delay measurement was almost in the same level between the video-only and AR-enabled conference. Number of air stamps affixed on the video image less than six did not increase the delay. Android-based mobile tablet device showed longer delay than Windows-based desktop computer because of less codec performance. Longer delay in the video-only conference caused by difference in the tablet's codec performance between front and rear camera employed in the video-only and AR-enabled conference.

Effective video frame rate was also compared. The base videoconferencing system showed 10 fps for VGA resolution, which was restricted by capability of the Android-based tablet. In an AR-enabled conference, effective frame rate was 8-10 fps.

These results denied any significant effect on video transfer performance in this configuration which the AR processing server intervention was expected to cause.

Current major determining factor of the delay and effective frame rate was performance of video codec especially on the mobile tablet device. In near future, if performance of mobile device improves faster than performance of the AR processing server, which is quite likely to happen, the result could change.

V. CONCLUSION AND FUTURE WORK

We developed a remote support service system based on the architecture we have proposed for videophone enhancement with Augmented Reality (AR) technology. We discussed requirements to develop a practical system and presented the system design determined based on the requirements. The system did not give significant modification on base videophone system, which suggests that the architecture can facilitate system reusability in enhancing existing videophone system. Evaluation indicated that the addition of AR functionality on top of existing videophone system did not cause performance degradation. The developed system will be tested in actual remote support scenarios to evaluate its effect on work performance and accuracy.

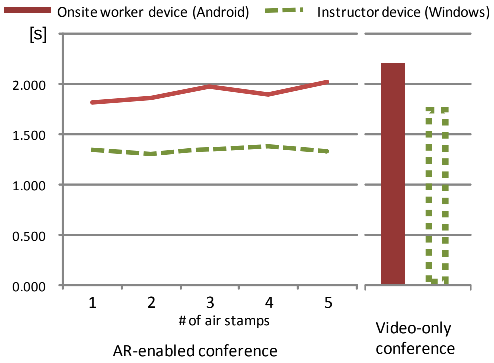


Figure 5. Round-trip video delay.

REFERENCES

[1] S. Araki, T. Hori, M. Fujimoto, S. Watanabe, T. Yoshioka, T. Nakatani, and A. Nakamura, "Online meeting recognizer with multichannel speaker diarization," *Conference Record of 44th Asilomar Conf. on Signals, Systems and Computers (ASILOMAR 2010)*, pp.1697-1701, 2010.

[2] H. Imaoka, Y. Morishita, and A. Hayasaka, "Real-time face recognition demonstration," *Proc. 2011 IEEE International Conf. on Automatic Face & Gesture Recognition and Workshops (FG 2011)*, pp.344, 2011

[3] G. Simon, A.W. Fitzgibbon, and A. Zisserman, "Markerless tracking using planar structures in the scene," *Proc. IEEE and ACM International Symposium on Augmented Reality, 2000. (ISAR 2000)*, pp.120-128, 2000

[4] A. Fukayama, S. Takamiya, J. Nakagawa, N. Arakawa, N. Kanamaru, and N. Uchida, "Architecture and prototype of augmented reality videophone service," *Proc. 15th International Conference on Intelligence in Next Generation Networks (ICIN2011)*, pp. 80-85, 2011.

[5] "Rich Communications Suite Release 4 - Service Definition -," Version 1.0, GSM Association, 2011.

[6] "RCES Phase 2 Stage 2 / 3 Specification Network Value Added Services," TS-1014, Version 1.1, Telecommunication Technology Committee, 2010.

[7] "Cloud-based Translator Phone," NTT docomo, http://www.nttdocomo.com/features/mobility36/, 2012. [retrieved: July, 2012]

[8] M. Billinghurst, A. Cheok, S. Prince, and H. Kato, "Real world teleconferencing," *IEEE J. Computer Graphics and Applications*, Vol. 22, Issue 6, pp. 11-13, 2002.

[9] I. Barakonyi, T. Fahmy, and D. Schmalstieg, "Remote collaboration using augmented reality videoconferencing," *Proc. Graphics Interface 2004, GI '04*, pp. 89-96, 2004.

[10] S. Bottecchia, J. Cieutat, and J. Jessel, "T.A.C: augmented reality system for collaborative tele-assistance in the field of maintenance through internet," *Proc. 1st Augmented Human International Conference, AH'10*, 2010.

[11] "Adding bots to your chat list," http://support.google.com/talk/bin/answer.py?hl=en&answer=172257-12-19-n41.html, 2007. [retrieved: July, 2012]