AFIN 2021

The Thirteenth International Conference on Advances in Future Internet

ISBN: 978-1-61208-920-1

November 14 - 18, 2021

Athens, Greece

AFIN 2021 Editors

Pascal Lorenz, Université de Haute-Alsace, France

AFIN 2021

Forward

The Thirteeth International Conference on Advances in Future Internet (AFIN 2021), held on November 14-18, 2021, continued a series of events dealing with advances on future Internet mechanisms and services.

We are in the early stage of a revolution on what we call Internet now. Most of the design principles and deployments, as well as originally intended services, reached some technical limits and we can see a tremendous effort to correct this. Routing must be more intelligent, with quality of service consideration and 'on-demand' flavor, while the access control schemes should allow multiple technologies yet guarantying the privacy and integrity of the data. In a heavily distributed network resources, handling asset and resource for distributing computing (autonomic, cloud, on-demand) and addressing management in the next IPv6/IPv4 mixed networks require special effort for designers, equipment vendors, developers, and service providers.

The diversity of the Internet-based offered services requires a fair handling of transactions for financial applications, scalability for smart homes and ehealth/telemedicine, openness for web-based services, and protection of the private life. Different services have been developed and are going to grow based on future Internet mechanisms. Identifying the key issues and major challenges, as well as the potential solutions and the current results paves the way for future research.

We take here the opportunity to warmly thank all the members of the AFIN 2021 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to AFIN 2021.

We also thank the members of the AFIN 2021 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that AFIN 2021 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the field of future Internet.

AFIN 2021 Chairs

AFIN 2021 Steering Committee
Patrik Österberg Mid Sweden University, Sweden
Martin Zelm, INTEROP - Virtual Lab, Brussels, Belgium

AFIN 2021 Publicity Chair
Mar Parra Boronat, Universitat Politecnica de Valencia, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain

AFIN 2021

Committee

AFIN 2021 Steering Committee
Patrik Österberg Mid Sweden University, Sweden
Martin Zelm, INTEROP - Virtual Lab, Brussels, Belgium

AFIN 2021 Publicity Chair
Mar Parra Boronat, Universitat Politecnica de Valencia, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain

AFIN 2021 Technical Program Committee

Ahmed M. Abdelmoniem, King Abdullah University of Science and Technology (KAUST), Saudi Arabia
Ahmed Abid, Easy Global Market | Sophia Antipolis, France
Lavanya Addepalli, Universitat Politecnica de Valencia, Spain
Pouyan Ahmadi, George Mason University, USA
S M Nahian Al Sunny, University of Arkansas, USA
Abdullah S. Al-Alaj, Virginia Wesleyan University, USA
Cristina Alcaraz, University of Malaga, Spain
Azzam Alsudais, King Saud University, Saudi Arabia
Rachida Aoudjit, University of Mouloud Mammeri, Tizi Ouzou, Algeria
Niloofar Bahadori, North Carolina A&T State University, USA
Marcin Bajer, ABB Corporate Research, Kraków, Poland
Michail J. Beliatis, Aarhus University, Denmark
Peter Bloodsworth, University of Oxford, UK
Stefan Bosse, University of Koblenz-Landau, Germany
Christos J. Bouras, University of Patras, Greece
Manuel José Cabral dos Santos Reis, IEETA / University of Trás-os-Montes e Alto Douro, Portugal
Lianjie Cao, Hewlett Packard Labs, Palo Alto, USA
Stephan Cejka, Siemens AG Österreich, Vienna, Austria
Felipe S. Dantas Silva, LaTARC Research Lab | Federal Institute of Technology of Rio Grande do Norte, Brazil
Joe Dong, UNSW Sydney, Australia
Jun Duan, IBM T.J. Watson Research Center, USA
Ivan Ganchev, University of Limerick, Ireland / "Paisii Hilendarski" University of Plovdiv, Bulgaria
Jordi Garcia, UPC BarcelonaTECH, Spain
Apostolos Gkamas, University Ecclesiastical Academy of Vella of Ioannina, Greece
Tenshi Hara, Saxon University of Cooperative Education | State Study Academy Dresden, Germany
Ahmad Ibrahim, University of Birmingham (Dubai Campus), UAE
Sergio Ilarri, University of Zaragoza, Spain
Paridhika Kayal, University of Toronto, Canada
Ahmed Khaled, Northeastern Illinois University, USA
Zaheer Khan, University of the West of England, Bristol, UK
Aminollah Khormali, University of Central Florida, USA

Copyright Information

Table of Contents

CoAP Message Transport with Packet Wash

Lijun Dong, Richard Li
Futurewei Technologies Inc.
2330 Central Expressway
Santa Clara, CA, U.S.A
Email: {lijun.dong, richard.li}@futurewei.com

***Abstract*—Network congestion in the current Internet usually causes packet loss and prolonged packet delivery latency due to retransmission. Qualitative Communication as one of the major features in New IP reduces the network layer operation from the packet level to much smaller granularity, namely packet wash. This paper illustrates such possibility by using Constrained Application Protocol (CoAP) as an example, which would be of wide usage owing to the thriving of Internet of Things (IoT) applications. The paper proposes two aspects of packet wash actions to CoAP packet: (1) header compression facilitated by cross-layer design; (2) partial packet dropping empowered by payload chunkification and New IP metadata. The paper elaborates on the detailed mechanisms, which provide many benefits with very slim overhead.**

Keywords- In-network; New IP; CoAP; block-wise transfer; packet wash; Qualitative Communication; UDP; IP; partial dropping; caching.

I. INTRODUCTION

Recently, New IP [1][2][3] has been proposed as an advanced network protocol specification to modernize the network layer without changing the fundamental Internet architecture. The New IP framework ensures the backward compatibility with the current Internet protocols and requires minimum effort to upgrade. It brings the intelligence of user and application awareness, continuity of services into the network. The concise New IP packet format as shown in Fig. 1 is designed to include three major elements:

1) New IP allows for hybrid formats of source and destination addresses according to the functionality and network interfaces of the communicating parties. The New IP addressing allows different types of addresses to be integrated and communicated in a flexible way.
2) New IP contract enables new types of modern courier-like network services at the finest packet-level granularity, e.g., High Precision Communication [4][5][6], Qualitative Communication[7][8][9]. The network and routers fulfill the contract. The Contract could include contract clause(s) and associated metadata. A contract clause specifies how the routers process the packet as it is forwarded in the network based on the configured triggering event and condition. The "Metadata" contains data about the packet, as well as the contextual information about the user/application, etc. And it can also allow New IP packet to collect the customized statistics about the flow on intermediate hops.
3) The New IP Payload can be the same as that in the existing IP, but can also be a sequence of sub-payloads, which share the same sources and destinations, but may be chuckified from currently intact payload. By reducing the granularity of operation on packet payload, the transport paradigm switches to the new Qualitative Communication [7][8][9]. Packet wash action in Qualitative Communication is designed to be taken when facing network congestion. A packet wash truncates a packet and only drops some portions of the packet to mitigate entire data loss. This has the effect of reducing the packet size until that packet is small enough to be stored and/or transmitted by the network node even though there is network congestion.

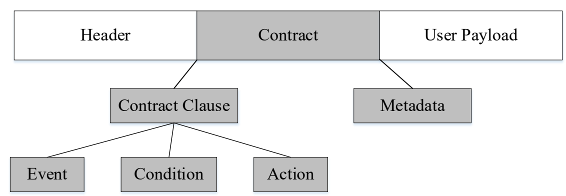

Figure 1. New IP: unified framework for future IP packet

Ideally, packets are transferred in their entirety without losing any part in the packet. However, under certain circumstances the controlled dropping of certain portions or fields in the packet as a last resort may be preferable over losing packet in their entirety, in particular when this means that extra delay due to the need for retransmission can be avoided. In our previous works on Qualitative Communication [8][9], the major effort has been focused on how to design the packetization methods to facilitate the chuckification of packet payload. Packet Wash is mainly applied to packet payload. However, Packet Wash could also exhibit certain commonalities with performing on-demand lossful compression.

In this paper, we use Constrained Application Protocol (CoAP) [10] as an example to show that the packets transported on the Internet may be subject to packet wash operation when network congestion presents. CoAP is specifically designed to satisfy the urge for connecting and integrating the low-power and constrained devices such as sensors and actuators at a global scale, which has pushed towards the Internet of Things (IoT) vision. The CoAP message format and its major features are summarized in Section II. The paper proposes a cross-layer design of Internet protocol stack to expose the upper layer headers to the network layer, some fields that are not absolutely necessary may be partially removed to reduce the headers' size. The paper presents such methods of further compressing CoAP header, correspondingly User Datagram Protocol (UDP) header, and traditional IPv6 header. On the other hand, with the adoption of New IP framework, the application layer contexts may be made aware to the network layer, which facilitates appropriate actions on the payload, including washing (partial dropping), caching, proactive re-delivery, etc., which will be described in detail in Section III. In Section IV, the overhead and benefits that are brought by the proposed methods are discussed. Section V concludes the paper.

II. CoAP Message

Internet Engineering Task Force (IETF) Constrained RESTful Environments (CoRE) Working Group (WG) [12] defines CoAP as a customized web transfer protocol for use in IoT nodes.

The CoAP messaging model is based on the exchange of messages running over UDP between endpoints. CoAP uses a short fixed-length binary header of four bytes that might be succeeded by a variable-length Token value between 0 to 8 bytes, compact binary options in TLV (Type-Length-Value) format and a payload filling the rest of the datagram. The CoAP message format is shown in TABLE I.

- Type (T): 2-bit unsigned integer, which indicates the type of the message, whether it is a Confirmable (0), Non-confirmable (1), Acknowledgement (2), or Reset (3) type.
- Token Length (TKL): 4-bit unsigned integer. The Token is used to match a response with a request.
- Code: 8-bit unsigned integer, which is split into a 3-bit class (most significant bits) and a 5-bit detail (least significant bits). There are 4 types of method codes with the most three significant bits set to 0: 0.01 GET, 0.02 POST, 0.03 PUT, 0.04 DELETE.
- Message ID: 16-bit unsigned integer, which is used to detect message duplication and to match messages of type Acknowledgement/Reset to messages of type Confirmable/Non-Confirmable.

CoAP defines less options than Hypertext Transfer Protocol (HTTP), i.e. If-Match, Uri-Host, ETag, If-None-Match, Observe, Uri-Port, Location-Path, Uri-Path, Content-Format, Max-Age, Uri-Query, Accept, Location-Query, Proxy-Uri, Proxy-scheme and Size1. The detailed definition and usage of those options could be found in the specification [10], or in this survey [11] for a short reading.

When included in a GET request, the Observe Option [13] requests the server to notify the client if changes happen to the target resource. Otherwise, the request falls back to a normal GET request. When included in a response, the Observe Option identifies the message as a notification. The present of Observe Option can differentiate a notification from a normal response.

CoAP also defines the block wise transfer [14] to limit the size of datagrams in constrained networks: by the maximum datagram size (~ 64 KB for UDP), or to avoid IP fragmentation (MTU of 1280 bytes for IPv6), or to avoid adaptation-layer fragmentation (60-80 bytes for 6LoWPAN). Two options (Block1, Block2) are used: when Block1 is present in a request or Block2 in a response, it indicates a block-wise transfer and describes which part of the entire payload this specific block-wise occupies.

TABLE I. CoAP Message format

Ver	T	TKL	Code	Message ID
Token (if any)				
Options (if any)				
1 1 1 1 1 1 1 1			Payload (if any)…	

III. System Architecture and Functionalities

In this section, we will present the cross-layer design for Internet protocol stack and in-network packet wash mechanisms for CoAP packets.

A. Cross-Layer Design for Internet Protocol Stack

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the US-letter paper size.

When two parties initiate a communication between them, the application layer header and application transport layer (CoAP, HTTP etc.) header are set up properly. The transport layer adds a Transmission Control Protocol (TCP) or UDP header in front of the application transport layer header and treats the application transport layer header and payload as opaque. The network layer adds an IP header on top of the TCP or UDP header, the application transport layer header and payload, and treats them as opaque. In principle, one layer is not able to access the fields in the header of another layer, either it is upper layer or lower layer. The methodology of layered protocol design possesses some advantages from the protocol transparency perspective. For example, protocols in one layer can be designed, improved, or even substituted without imposing any influence on other protocol layers. However, it is likely that the information from one layer may be useful to another layer. As a result, the performance optimization between different protocol layers becomes impossible under such methodology of layered protocol design, which can significantly degrade the network performance. This unavoidably leads to the cross-layer design. The concept of cross layer design is about sharing of information among different protocol layers for adaptation purposes and to increase the inter-layer interactions.

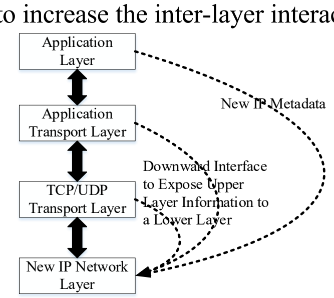

Figure 2. Cross-layer design for information sharing across layers

To avoid above restriction, Figure 2. shows that alternatively the application layer may a maintain downward interfaces to expose upper layer information to the network layer, such that the network layer is able to utilize the information from the application layer such as the type of payload data, characteristics of payload data, user's requirement on performances such as latency and quality of the data, etc., as well as its tolerance level on the disparity from the exact requirement. If the underlying network layer supports New IP framework, the application layer information can be naturally passed to the network layer through New IP Metadata. The application transport layer, and TCP/UDP transport layer protocol layer instead may maintain downward interfaces to expose their headers to IP protocol layer to enable the possible packet wash mechanisms proposed in this paper.

The paper mainly focuses on the protocol stack with CoAP protocol as the application transport protocol, correspondingly UDP as the transport layer protocol and proposes the in-network packet wash mechanisms for CoAP messages.

B. Details of In-Network Packet Wash Mechanisms

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

The in-network packet wash proposed in this paper mainly aims to design the optimal mechanisms to compress the headers, drop/cache the least important information and transport the most necessary information to the destination with the shortest latency when the network condition is not desirable. As shown in Figure 3. , between the two CoAP end nodes, there exists the network nodes (e.g., base station, gateway, routers) that would route and forward the CoAP messages. The in-network packet wash enables each of the intermediate network nodes to have the capability of differentiating multiple blocks in the packet and dispose them discriminately. Note, for the purpose of evolving deployment of such capability, some of the network nodes may remain with legacy implementation, which will simply either forward or drop the packet completely. For those network nodes (i.e., Router 2, 3, 5 and Base Station 6) which are enabled with the in-network packet wash capability for CoAP messages, the packet is processed with certain level of compression and discard when facing network congestion, which we will discuss in the later section.

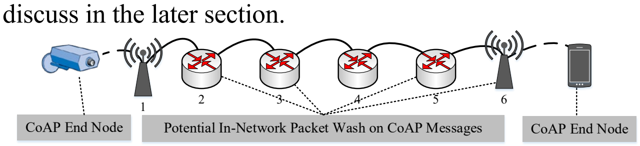

Figure 3. Illustrative example

Due to the simplicity and RESTful nature of CoAP protocols, the most common CoAP messages that are sent between two end nodes are the combinations of: [Confirmable, Non-Confirmable, Acknowledgement, Reset]+[GET, PUT, POST, DELETE]+[Request, Response]. Since the only difference between Confirmable and Non-Confirmable messages is that whether the message is acknowledged by the recipient upon arrival, we don't distinguish them later. Among those messages, the GET response, POST request, PUT request may contain data in the payload, which the paper will focus on. Given the GET response, POST request, PUT request messages have similar syntax, depending on whether it is initiated by the client or server, in the paper, we use the GET response message as the example to illustrate how a CoAP message may be washed by the network nodes. Other types of CoAP messages can be treated similarly.

The GET response message is sent by a CoAP end node upon receiving a GET request message or triggered by a notification due to an existing subscription. The only difference between those two cases is that whether the Observe Option is included in the message. In order to facilitate the in-network packet wash on the GET response message, we propose that in the corresponding GET request message, the requesting end node's application layer will optionally pass the following parameters to the New IP Metadata:

- Type of requested data that corresponds to the application (e.g., surveillance video, temperature)
- Latency budget for returned data
- Required quality of returned data (e.g., for video data, view angle, resolution etc.)
- Tolerance degree on the distortion of the data quality

When packet wash capable network node (e.g., Router 5 in Figure 3.) receives a GET response packet, it may make decisions on processing the packet, depending on the current network condition. The actions it may take include:

- Compress the CoAP header by accessing the CoAP header fields through the downward interface from the application transport layer to New IP network layer.
- Compress the UDP header by accessing the CoAP header fields through the downward interface from the application layer to New IP network layer.
- Compress the IP header.
- Cache the payload data and remove the payload from the packet.
- Truncate the payload partially or completely.

1) New IP Metadata Design

Status (6 bits): It indicates whether the packet has been washed by the previous network nodes which have forwarded the packet. The very first bit indicates whether the packet is original or not (e.g., 000000 denotes that the packet is original). The second bit indicates whether the packet's CoAP header is compressed or not (e.g., 110000 denotes that the packet's CoAP header has been compressed). The third bit indicates whether the packet's UDP header is compressed or not (e.g., 101000 denotes that the packet's UDP header has been compressed). The fourth bit indicates whether the packet's IP header is compressed or not (e.g., 100100 denotes that the packet's IP header has been compressed). The fifth bit indicates whether the packet's payload is partially dropped or not (e.g., 100010 denotes that the packet's payload has been partially dropped). The sixth bit indicates whether the packet's payload is completely dropped (e.g., 100001 denotes that the packet's payload has been completely dropped). Any combination of the 5 types of revisions to the original packet

Copyright (c) IARIA, 2021. ISBN: 978-1-61208-920-1

could exist by setting the corresponding bit to 1 and results in the first bit to be 1.

CoAPWash (2 bits)*:* Each bit indicates whether a field is removed/dropped from the original CoAP header. The first bit indicates whether that the *TKL*, *MessagID* and *Tokens* fields are removed. The second bit indicates whether the *content-format*, *max-age*, and *ETag* options are removed and cached.

UDPWash (4 bits): The first bit indicates whether the source port is compressed to 4 bits. The second bit indicates whether the destination port is compressed to 4 bits. The third bit indicates whether the *Length* field is removed. The fourth bit indicates whether the *Checksum* field is removed.

IPWash (6 bits): The first bit indicates whether the *Traffic Class* field is removed. The second bit indicates whether the *Flow Label* is removed. The third field indicates that the *Payload Length* field is removed. The fourth bit indicates whether the *Next Header* is removed and cached. The fifth bit indicates whether the *Hop Limit* field is removed and cached. The six bit indicates whether the *Source Address* and *Destination Address* fields are removed.

IPExtCache (4 bits): Each bit indicates whether the extension header is removed and cached, following the sequence as shown in TABLE II. The CoAP options that could be included in a GET response message are: *Content-Format*, *Max-Age*, *Etag* and *Observe*.

Multi (1 bit): It indicates whether there are simultaneous requests between the requesting end node and responding end node. If the bit is set, the *Tokens* field must be included in the CoAP header and cannot be removed. If the bit is not set, a network node may remove the *TKL* and *Tokens* field in the CoAP header. On the other hand, if there is only one request between the two end nodes, the *Message ID* field can also be removed since the requesting end node can match the response to the request by the responding end node's address.

TABLE II. OPTIONS IN COAP GET MESSAGE

Name	**Length (bytes)**
Etag	1-8
Content-Format	0-2
Max-Age	0-4
Observe	0-3

TagCache (1-8 bytes): The identifier of the cached copy of payload and other information stored in an intermediate network node, which can be used by the requesting node to retrieve the payload later when the network condition becomes satisfactory. Such identifier could be very short, as far as it is unique among all the cached content stored in the intermediate network nodes between source and destination. The field may contain multiple identifiers of cached portions of the current CoAP message.

Significance (1-8 bits): If the message is one block in the block-wise transfer, this field can be used to indicate the significance of the block in recovering or interpreting the original data. A network node can use this field to decide whether the payload in the message may be dropped.

Selects (length varies): This field is used to include any possible requirements from the requesting node (client), as well as properties related to the actual data being returned by the responding node (server). The *Type-of-Data* lets the network nodes understand the type of data included in the payload, which in turn may decide on the priority of the packet. For a multimedia type of data, the examples of the application layer parameters are proposed and shown in TABLE III. The *Tolerance-Degree* option following the previous option is to indicate the requesting node could have some level of tolerance if the data is not exactly matching its requirement. This *Tolerance-Degree* could give the flexibility to the network node to selectively drop some parts of the packet payload to fit the current network condition, with some sacrifice to the multimedia data' quality in resolution or view-angle, or performance in latency.

TABLE III. SELECTS IN METADATA SEGMENT

No.	Name	Format	Length
1	Type-of-Data	string	0-1
2	Latency-Budget	unit	0-1
3	Tolerance-Degree	unit	0-1
4	View-Angle	unit	0-2
5	Tolerance-Degree	unit	0-1
6	Resolution	unit	1
7	Tolerance-Degree	unit	0-1

2) Actions on Payload

A network node can take the following actions on the payload, the overall procedure is shown in Figure 4. .

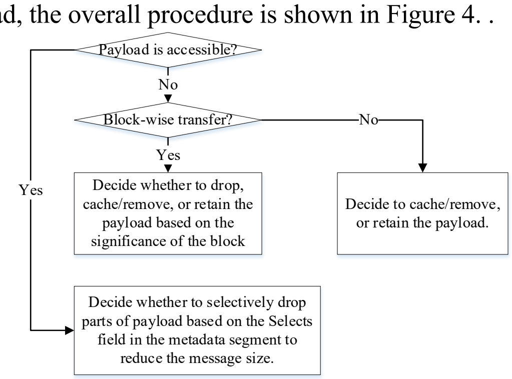

Figure 4. Actions on the payload

If the payload is encrypted, the data itself is invisible to the network node and cannot be processed in any method. However, the network node can decide to whether to cache the payload and other associating information in the CoAP header, as well as remove it completely from the original message. If the network node caches the payload and other associating information in the CoAP header (e.g., *Content-Format*, *Max-Age*, *ETag*) and removes the payload from the original message, the network node will set the very first bit in the *Status* field in the metadata segment to 1 to indicate such action. On the other hand, the network node can set the *TagCache* field in the metadata segment to include its local identifier of the cached content for the requesting node to retrieve the data when the network condition becomes better. If the *TagCache* is not setup after the network node drops the payload from the original message, it indicates that the network node is in charge of sending the data contained in the payload to the requesting node after it sees the network condition becomes better. The requesting node only needs to expect and wait for the data to be delivered later.

If the message is one block in the block-wise transfer, we propose that the responding node may organize the blocks such that each of the blocks may contain different parts of the requested data with different importance levels. In the *Significance* field of the metadata segment in the message, it will indicate the relative importance of the current block compared to the other blocks. The larger the *Significance* value is, the more relatively important the block is to recover the information contained in the data. The network node could decide whether to drop the block given that the network condition is not satisfactory to transport the block to the next hop. If the block is relatively important, the network node needs to try its best to retain the payload and send to the requesting node. Otherwise, the payload could be dropped, removed/cached for later retrieval or delivery.

A network node may need to decide on which CoAP messages to perform packet wash operation on, if there are multiple candidates from different flows waiting in the outgoing queue. The relative significance of a CoAP message is determined as shown in (1).

$$sig = \begin{cases} 1, & \text{if non} - \text{blockwise} \\ Significance, & \text{if blockwise} \end{cases} \quad (1)$$

$$Significance = \alpha * \frac{sig_{block}}{num} \quad (2)$$

If a CoAP message does not have block-wise transfer options (i.e., block1 and block2 options), then the significance level of the message is regarded as 1. Otherwise, the relative significance of the CoAP message among all other blocks from the same flow is indicated in the Significance metadata field, which is basically assigned by the sender as the significance level of the block divided by the number of blocks in the flow. The factor α would make sure that the significance level of the blocks which the sender considers as the most important and are preferred to be delivered without re-transmission is larger than 1, as shown in (2).

If the payload is not encrypted or different chunks within the payload are encrypted independently, the network nodes can selectively drop parts of the payload based on the network condition, user's requirement and application layer parameters included in the metadata segment of the message. The *Selects* field is used to specify those proposed requirements, which could be set up by the requesting node when the CoAP request message is sent. The responding node prepares the data according to the requesting node's requirements specified in the *Selects* field. When the response message is sent, the actual properties of the data (e.g., *Type-of-Data*, *Resolution*, *View-Angle*) are setup accordingly. The related *Tolerance-Degree* is copied from the request message. On the other hand, the *Latency-Budget* is modified for the response message to reach the requesting node by deducting the used time from the original latency budget. Based on the *Tolerance-Degree*, the message size may be reduced. For example, the resolution may be adapted by selectively dropping some parts of payload, such that the resolution could be lower than the current value, but higher than the requesting node's tolerable value. Consequently, the packet size is reduced, and the packet can be avoided being completely dropped.

3) CoAP Header Compression

The CoAP header of a GET response packet could be possibly compressed to reduce message size, under the condition that some of the fields in the header may be removed without influencing the processing or understanding of the CoAP header at the receiver side. If the CoAP header is not compressed by any previous network node on the path from the sender to the receiver, based on the status field in the New IP metadata (the first and second bit in *Status* field), the currently received message might be eligible for CoAP header compression. Otherwise, the network node moves to the UDP header, which will be discussed in Section 4). The proposed algorithm of compressing CoAP header of GET response message is shown in Figure 5.

Algorithm 1 CoAP Header Compression

```
If the first two bits of Status is "11", then
    The CoAP header has been compressed and no further
    actions needed on the CoAP header.
else if the second bit of Status is "0", then
    if multi = 0, then
        remove TKL, MessageID and Tokens field if they exist.
        Set the first bit of CoAPWash to "1".
    end
    if payload is removed from the message and cached, then
        Remove content-format, max-age, and ETag options.
        Cache content-format, max-age and ETag.
        Set the second bit of CoAPWash to "1".
    else
        Retain content-format and max-age options
        Remove ETag option.
    end
        Set the first and second bit in the Status field is set to
        "11".
end
```

Figure 5. Algorithm 1

In the CoAP header, the *TKL*, *MessagID* and *Tokens* field can be removed if there are no simultaneous requests between the requesting end node and responding end node, which is indicated in the *Mutli* field in the metadata by the responding end node.

The CoAP options that could be included in a GET response message are: *Observe*, *Content-Format*, *Max-Age* and *ETag*. The *Observe* Option indicates that this is a notification for the subscription. This option should not be removed. The *Content-Format* Option indicates that the representation format of the message payload. The representation format is given as a numeric content format identifier that is defined in the "CoAP Content-Formats" registry [15]. It should not be removed if the payload stays in the message. The *Max-Age* Option indicates that the maximum time a response may be cached before it is considered not fresh. It should not be removed if the payload stays in the message. The *ETag* Option is generated by the responding node and used as a resource-local identifier for differentiating between representations of the same resource that vary over time. The *ETag* Option in a response provides the current value of the entity-tag for the requested resource representation in the payload. The *ETag* Option can be removed, resulting in that the requesting node is not aware of the entity-tag of the received data, which is not crucial information in interpreting the representation.

After the CoAP header is processed and compressed according to the algorithm proposed above, the first two bits in the *CoAPWash* field in metadata segment of the message is set up accordingly and the first and second bit in the *Status* field is set to "11". For the future network nodes, after they detect the second bit in the *Status* field is set up, they would not act on the CoAP header anymore.

4) UDP Header Compression

UDP (both source and destination) ports may be compressed to 4 bits, if the requesting and responding nodes agree to only use 16 number of specified ports for different applications which are using CoAP as the application layer protocol. Other than the source port and destination port, there are two other fields in the UDP header:

- Length: It indicates the length in bytes of the UDP header and the encapsulated data. The minimum value for this field is 8. This field can be removed without influencing packet interpretation.
- Checksum: This is computed as the 16-bit one's complement sum of a pseudo header of information from the IP header, the UDP header, and the data, padded as needed with zero bytes at the end to make a multiple of two bytes. If the checksum is set to zero, then checksuming is disabled. If the computed checksum is zero, then this field must be set to 0xFFFF. Since a network node may partially drop the IP header, UDP header and the payload based on the mechanisms proposed in the paper, the Checksum field needs to be disabled if any modifications happen to the original packet, thus is removed (in this paper, removing Checksum field means disabling the checksuming).

After a network node processes the UDP header, the *UDPWash* field in the metadata segment is setup correspondingly.

5) IP Header Compression

The standard IPv6 header is composed of the fields as shown in Figure 6. .

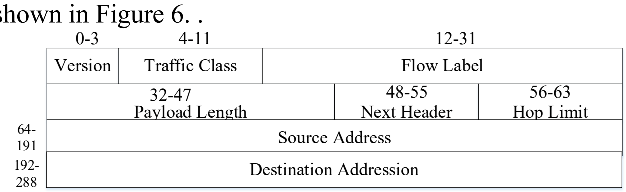

Figure 6. IPv6 header

- Version (4-bits): It represents the version of Internet protocol, i.e., 0110, which can be fixed, thus can be removed.
- Traffic Class (8-bits): These 8 bits are divided into two parts. The most significant 6 bits are used for Type of Service to let the routers know what services should be provided to this packet. The least significant 2 bits are used for Explicit Congestion Notification (ECN). For a CoAP message, if this field is set up by the source, it should be not removed, and the routers need to take actions according to the Type of Service and Explicit Congestion Notification to forward the message. Otherwise, this field can be removed.
- Flow Label (20-bits): This label is used to maintain the sequential flow of the packets belonging to a communication. The source labels the sequence to help the router identify that a specific packet belonging to a particular flow of information. This field helps avoid re-ordering of data packets. Since the data can be carried in one CoAP message or multiple ones with block-wise transfer, the Flow Label is not necessary for the requesting node to recover the data accurately. The Block2 option contains the order information of the blocks. Thus, the Flow Label field can be removed.

TABLE IV. IPV6 EXTENSION HEADER TYPES AND SEQUENCE IN THE MESSAGE

IPv6 header
Hop-by-Hop Options header
Destination Options header
Routing header
Fragment header
Authentication header
Encapsulating Security Payload header
Destination Options header
Upper-layer header

- Payload Length (16-bits): This field is used to tell the routers how much information a particular packet contains in its payload. Payload is composed of Extension Headers and Upper Layer data. The field needs to be changed if the upper layer header compression and partial payload dropping happen.
- Next Header (8-bits): This field is used to indicate whether the type of Extension Header is present. The types and sequence of the Extension Headers are shown in TABLE IV. If we assume that the blocks in the block-wise transfer share the same values for the Extension Headers if they are included in the IPv6 header, then the Extension Headers only need to be transferred once in the first block in the block-wise transfer. The Extension Headers can be extracted from the cached copy stored in either the receiving node (in the scenario of successful delivery of the first block) or the intermediate network node (in the scenario of the Extension Headers of the first block are washed and cached during the transmission). This proposal also applies to the Hop Limit, Source Address and Destination Address.
- Hop Limit (8-bits): This field is used to prevent a packet to loop in the network endlessly. The value of Hop Limit field is decremented by 1 as it passes a hop. When the field reaches 0 the packet is discarded. This field cannot be removed.
- Source Address (128-bits): This field indicates the address of originator of the packet.
- Destination Address (128-bits): This field provides the address of intended recipient of the packet.
- After a network node processes the IP header, the IPWash and IPExtCache fields in the metadata segment are setup correspondingly.

IV. Discussions on Overhead and Benefits

The proposed methods rely on the New IP metadata inserted in the packet to instruct the intermediate network nodes to take actions when network congestion appears, which inevitably increases the packet size. However, we can affirm that such size overhead is extremely light and compatible with the existing packet size configurations, which will be explained below.

For a CoAP message, it usually fits within a single IP packet to avoid IP fragmentation (MTU of 1280 bytes for IPv6). The good upper bounds are 1152 bytes for the message size and 1024 bytes for the payload size. If the data is larger, the block-wise transfer would be implemented. The CoAP header size has a fixed value of 4 bytes. Followed by the fixed header fields, there could exist token and options. The token size can be as large as 8 bytes, while the options could be of varying sizes and the total size of options in TABLE II. could reach 17 bytes. By abstracting 1152 from 1280 bytes, the capacity that could be used by the New IP metadata and lower layer headers is 128 bytes. The UDP header size is 2 bytes. The IPv6 header size could be 80 bytes, including the extension headers. The total size of the New IP metadata proposed in Section III.B.1) can be as high as 20 bytes (less than 46=128-2-80). Thus, the New IP metadata size is small enough to be carried in the CoAP message.

CoAP is bound to unreliable transports such as UDP, it implements a lightweight reliability mechanism, without trying to re-create the full feature set of transport like TCP. It adopts a simple stop-and-wait retransmission reliability with exponential back-off for Confirmable messages. Qualitative Communication does not entirely eliminate the need for re-transmission since it cannot mitigate against irrecoverable loss of critical elements of the packet. However, the amount of information needing retransmission is dramatically reduced, since the critical information that is contained in CoAP blocks has higher priority/significance and is less prone to discards than before. As proposed in (2), the most important block(s) in a block-wise transfer would have higher significance level than the other concurrent CoAP messages queued in a congested network node, which ensures that they are more likely to be retained as much as possible and reach the receiver. Then the receiver is able to obtain the most crucial information without any extended delay caused by packet loss and retransmission.

V. Conclusions

Qualitative Communication has been proposed to mitigate the performance degradation in throughput, latency due to entire packet dropping caused by network congestion. Instead of dropping packets completely, packet wash proposed for Qualitative Communication is an action triggered by network congestion to partially remove some parts from the packet such that the packet can be retained in the outgoing queue and survive the deteriorating network condition. In this paper, we propose that packet wash could be applied to both packet headers and payload. Through the enablement of cross-layer design, the upper layer headers are potentially visible and compressible by the network nodes. We use CoAP protocol as an example to show the details of header compression mechanisms, i.e., to the CoAP header, UDP and IPv6 header correspondingly. On the other hand, thanks to the unified New IP framework, CoAP message could be encapsulated in New IP packet with metadata, which can be leveraged to pass some application or user's context to the network node to make intelligent and more effective packet washing operation on the packet payload. The paper also discusses the overhead might be brought by the proposed mechanism, as well as its feasibility from the perspective of packet size limit. In the last but not least, the benefits are summarized. A proof-of-concept prototype is planned as future works.

References

[1] R. Li, A. Clemm, U. Chunduri and L. Dong , "New IP: Enabling the Next Wave of Networking Innovation," book chapter in "Design Innovation and Network Architecture for the Future Internet", pp. 1-42, IGI Global.

[2] R. Li, K. Makhijani and L. Dong, "New IP: A Data Packet Framework to Evolve the Internet," IEEE HPSR 2020.

[3] R. Li, A. Clemm, U. Chunduri, L. Dong, and K. Makhijani, "A New Framework And Protocol For Future Networking Applications," ACM Sigcomm Workshop on Networking for Emerging Applications and Technologies (NEAT 2018).

[4] L. Han, Y. Qu, L. Dong and R. Li, "A Framework to Realize the Guaranteed Service for Bandwidth and Latency for Future IP network," 2020 Infocom workshop on New IP: The Next Step.

[5] A. Clemm and T. Eckert, "High-Precision Latency Forwarding over Packet - Programmable Networks," IEEE/IFIP Network Operations and Management Symposium, 2020.

[6] L. Dong and R. Li, "Packet Level In-Time Guarantee: Algorithm and Theorems", IEEE Globecom 2020.

[7] R. Li, K. Makhijani, H. Yousefi, C. Westphal, L. Dong, T. Wauters, and F. De Turck, "A Framework For Qualitative Communications Using Big Packet Protocol," ACM Sigcomm Workshop on Networking for Emerging Applications and Technologies (NEAT 2019).

[8] L. Dong and R. Li, "In-Packet Network Coding for Effective Packet Wash and Packet Enrichment," 2019 IEEE Globecom Workshop on Future Internet Architecture, Technologies and Services for 2030 and Beyond.

[9] L. Dong, K. Makhijani and R. Li, "Qualitative Communication Via Network Coding and New IP," IEEE HPSR 2020.

[10] Z. Shelby, K. Hartke and C. Bormann, "The Constrained Application Protocol (CoAP)," IETF RFC 7252.

[11] L. Dong and R. Li, "A Survey on IETF Standardization for Connecting and Integrating the Low-Power and Constrained IoT Devices," 2020 International Conference on Computing, Networking and Communications (ICNC).

[12] Constrained RESTful Environments (CoRE), IETF, https://datatracker.ietf.org/wg/core/about/.

[13] K. Hartke, "Observing Resources in the Constrained Application Protocol (CoAP)," IETF RFC 7641.

[14] C. Bormann amd Z. Shelby, "Block-Wise Transfers in the Constrained Application Protocol (CoAP)," IETF RFC 7959.

[15] Z. Shelby, "Constrained RESTful Environments (CoRE) Link Format," IETF RFC 6690.

 ISBN: 978-1-61208-920-1

Borel Cayley Graph-based Topology Control for Power Efficient Operation in Ad Hoc Networks

Abdellatif Serhani
RAI2S, INPT
University of Mohammed V
Rabat, Morocco
abdollatiif@gmail.com

Najib Naja
RAI2S, INPT
University of Mohammed V
Rabat, Morocco
naja@inpt.ac.ma

Abdellah Jamali
RI2M, ENSA
University of Hassan I
Settat, Morocco
abdellah.jamali@uhp.ac.ma

***Abstract*—In this paper, we focus on the design of network topology control algorithm in wireless ad hoc networks. We approach topology control with the well known Borel Cayley Graphs (BCGs), a family of pseudo-random graphs. BCGs have been shown to be an efficient candidate topology in interconnection networks due to their small diameter, short path length and low degree. We consider the problem of adjusting the transmit powers of nodes before assigning IDs of Borel Cayley Graphs as logical topologies in wireless ad hoc networks. We compare the performance of our algorithm with other existing topology control algorithms. Our simulation indicates that the proposed ID assignment has a better performance compared with other assignment methods.**

***Index Terms*—Distributed algorithms, power management, Borel Cayley Graph, Topology Control**

I. Introduction

A multihop wireless networks, such as radio networks, ad hoc networks, and sensor networks, are networks where a packet may have to traverse multiple consecutive wireless links in order to reach its destination. Mobile Ad hoc NETwork (MANET) is a mobile, multihop wireless network which is capable of autonomous operation. It is characterized by energy-constrained nodes, bandwidth-constrained, variable-capacity wireless links and dynamic topology, leading to frequent and unpredictable connectivity changes.

The absence of a central infrastructure implies that an ad hoc network does not have an associated fixed topology. Indeed, topology control, defined by how to determine an appropriate topology over which high-level routing protocols are implemented, has emerged to be one of the most important issues in wireless multi-hop networks [1]. Topology control is important in wireless ad hoc networks for at least two reasons: (*i*) It affects networks spatial reuse and hence the traffic carrying capacity, and (*ii*) It impacts on contention for the medium and hence mitigates MAC-level interference.

The topology depends on ”uncontrollable” factors, such as node mobility, weather, interference, noise, as well as on ”controllable” parameters, such as transmit power and antenna direction [2]. Then, several topology control algorithms lie in the fact that each node can potentially change the network topology by changing the transmit powers to control its set of neighbors [2] [4] [5]. Whereas, other techniques react on routing mechanisms, based on predefined metrics, that efficiently deal with changes in the topology due to uncontrollable factors.

According to [3], topology control algorithms can be classified as location-based, direction-based, or neighbor-based approaches. In each of these approaches, a resulting topology is returned by the topology control algorithm in an ad hoc manner. However, since these approaches do not generate a predefined graph, there is no guarantee on the topology's overall properties, such as a bounded diameter and average path length.

In this paper, we approach topology control with predefined structured graphs. We react on adjusting transmission power to reduce interference and hence expanding the network's lifespan. Thus, we achieve higher throughput as compared to schemes that use fixed transmission power [6]. Structured graphs have been studied for a long time and are good candidates for interconnection networks [7]. Various graph-based interconnection networks have been studied and used as good physical topologies for wavelength routing lightwave networks [7]. In peer-to-peer overlay network schemes, various predefined structured graphs are used to form a P2P overlay. The proposed systems utilize the properties of the graph to form the overlay network and develop a routing algorithm. Examples of P2P overlay networks include Arrangement-Graph Overlay (AGO) to reduce system overhead and bind routing hops [8], k ring lattices with Chords [9], and PeerStar by utilizing the star interconnection network as a peer to peer topology [10]. Graph-based wireless sensor networks have also been explored by other researches [11] [12]. In general, a predefined graph topology with its deterministic connection rule facilitates performance analysis. In addition, some offer symmetry, hierarchy, and hamiltonicity, and can have a constant low degree and existing distributed routing protocol, all preferable properties for ad hoc networks.

This paper features three major contributions.

Fistly, we focus on Borel Cayley graphs (BCG), a member of Cayley graph family [13], and we are interested in the problem of imposing and overlying the physical topology, defined by the transmission range of each node, by the logical topology dictated by our BCGs. We focus on this theoretic structure graphs because of their relatively small diameter

 ISBN: 978-1-61208-920-1

and low constant degree. Furthermore, these graphs also have a small average hop count between nodes and a simple distributed routing algorithm [14]. Let ***V*** denote the collection of nodes and let ***G*** denote the physical graph on ***V*** in which there is an edge from node *u* to node *v* if and only if *u* can directly reach *v*. when running our topology control algorithm, each node in ***V*** can be assigned an ID of the entire BCGs, and in result returning an undirected subgraph T_{BCG}, such that (*i*) T_{BCG} consists of all the nodes in ***G*** but has fewer edges, (*ii*) if *u* and *v* are connected in ***G***, they are still connected in T_{BCG}, and (*iii*) a node *u* can transmit to all its neighbors in T_{BCG} using less power than is required to transmit to all its neighbors in ***G***. Secondly, in our topology control algorithm, each node *u* in the network constructs its neighbor set $N(u) = \{v|(u,v) \in \boldsymbol{T}\}$ in a distributed fashion. Finally, if T_{BCG} changes to T_{BCG}**'** due to node failures or mobility, our topology control algorithm reconstructs automatically a connected T_{BCG}**'** without global coordination.

The outline of this paper is as follows. Section 2 reviews the definition and properties of BCGs and gives preliminaries on outdoor radio propagation and the power consumption model. In Section 3, we present our distributed topology control algorithm. Section 4 describes network simulation results that show the effectiveness of the algorithm. Finally, Section 5 summarizes our contributions and conclusion.

II. Related work

Topology Control (TC) is defined as communication nodes having the ability to modify or select their neighbors or connections (i.e. active links) at their will. This allows each node to be part of the network, based on its own judgment of constraints and resources for optimum operation. A network topology is formed in an ad hoc manner based on nodes' location, direction, or some sort of neighbor ordering. Topology control also aims at achieving specific design goals such as energy efficiency and interference mitigation by selecting logical neighbors and adjusting the transmission power accordingly.

In [15], Blough et al provided a K-$Neigh$ topology control algorithm in which a node chooses k closest physical neighbors as logical neighbors. K-$Neigh$ is a basic neighbor-based topology control protocol based on the construction of k-neighbor graph as logical communication graph that guarantees an interference bounded topology connected with high probability, provided the maximum power topology (i.e., the graph obtained when all the nodes transmit at maximum power) is connected. K-$Neigh$ is a simple, fully distributed, asynchronous, and localized protocol. The overall number of messages exchanged by K-$Neigh$ is exactly two times of the number of nodes and the execution time is strictly bounded.

In [16], Wattenhofer et al. described the XTC topology control algorithm which is one of the few topology control protocols which are location-free. XTC is based on ranking information between nodes. Their algorithm try to remove long links while preserving network connectivity to force nodes to use multiple short hops, which saves energy and prolongs network lifetime. Their technique does not assume location information, neither does it require the network to be a Euclidean graph. The XTC algorithm consists of three steps. In the first step, each node broadcasts once at the maximum power and then ranks all its neighbors according to its link quality to them (from high to low). Each node transmits its ranking results to neighboring nodes during the second step. In the final step, each node locally examines all of its neighbors in the order of their ranking and decides which one needs to be directly linked. The XTC algorithm features the basic properties of topology control such as symmetry and connectivity while running faster than most previous algorithms.

Ning Li et al. [17] presented a Minimum Spanning Tree (MST) based topology control algorithm, called Local Minimum Spanning Tree ($LMST$). Nodes in $LMST$ topology control algorithm construct a local minimal spanning tree with a bounded logical node degree and establish bidirectional link with a guaranteed connectivity. Each node builds its local MST independently and only keeps on-tree one-hop nodes as its neighbors. Their approach tries to minimize the overhead to maintain a connected topology in a dynamic wireless ad hoc network which the degree of any node in the resulting topology can be bounded by 6. The bounded degree on each node is desirable because a small degree reduces the MAC level contention and interference.

Ramanathan et al. [2] proposed to adjust incrementally node transmit powers in response to topological changes so as to maintain a connected topology using minimum power. Using (bi)connectivity as their objective, They described optimal centralized algorithms and distributed heuristics for transmit power control. In a mobile ad hoc network, they proposed a two distributed heuristics for topology control, namely Local Information No Topology ($LINT$) and Local Information Link-State Topology ($LILT$). In the the former, each node checks periodically the number of active number of neighbors (degree) in its neighbor table (built by the routing mechanism) and adjust the transmit powers based on the formula 1 to reduce the power.

$$p_d = p_c - 5.\mathcal{E}.log(\frac{d_d}{d_c}) \quad (1)$$

where d_d, d_c, p_c and p_d denote, respectively, the desired degree, the current degree, the current transmit power of a node, and the targeted power. The propagation loss function varies as some $\mathcal{E}$ power of distance.

In these techniques and in many other TC techniques, such as Common power ($COMPOW$) protocol [19], cone based topology control ($CBTC$) [4], extended topology control (XTC) [16], K-$Neigh$ topology control and adaptive neighbor-based topology control ($ANTC$) [18], a network topology is formed in ah hoc manner to retain a minimum number of interconnections among the nodes that can communicate by expending very little energy. However, since these approaches do not generate a predefined graph, there is no guarantee on the topology's overall properties such as a bounded diameter and average path length. Furthermore,

 ISBN: 978-1-61208-920-1

producing a degree bounded network topology that preserves connectivity in the worst-case is a challenge [15]. In other words, the goals of preserving worst-case connectivity and having a nontrivial upper bound on the physical node degree inherently conflict with each other.

Motivated by this observation, we tackle the TC problem with the goal of generating a network topology by imposing a predefined well-known $BCGs$ graph topology with a constant low-degree d_{BCG}. More precisely, we produce a network topology in which the physical node degree as well as the logical node degree are upper bounded by exactly the same BCG degree d_{BCG} by adjusting the transmit power for each node and its d-neighborhoods.

III. Preliminaries

In this section, we provide a brief summary of the definition and important properties of Cayley graphs, and Borel Cayley graphs [13].

A. Cayley Graph

A Cayley graph is a special family of pseudo-random graphs constructed from a finite group of elements which correspond to the nodes of the graph [14]. Connections between nodes of Cayley graphs are defined by the group operation and a set of generators. The formal definition of Cayley graphs is as follows:

Definition 1 (*Cayley graph* [14]). A graph $C = \mathbb{C}(V, G)$ is a Cayley Graph with vertex set V if two nodes $v_1, v_2 \in V$ are adjacent $\Leftrightarrow v_1 = v_2 * g$ for some $g \in G$, where $(V, *)$ is a finite group and $G \subset V \backslash \{I\}$. G is called the generator set of the graph and I is the identity element of the finite group $(V, *)$.

The definition of Cayley graph requires vertices to be elements of a group but does not specify a particular group. Thus, a Cayley graph can be generated over an arbitrary finite group, and there are many varieties of Cayley graphs.

B. Borel Cayley Graph

The Borel Cayley graphs (BCGs) are Cayley graphs constructed from Borel subgroups. The BCGs are regular, vertex transitive, and pseudo-random graphs [14]. The definition of a Borel subgroup is given below.

Definition 2 (*Borel subgroup* [14]). Let V be a Borel subgroup, $BL_2(\mathbb{Z}_p)$, of the nonsingular upper triangular 2×2 matrices $GL_2(\mathbb{Z}_p)$ with a parameter a such that $a \in \mathbb{Z}_p \backslash \{0, 1\}$, then

$$V = \left\{ \begin{pmatrix} x & y \\ 0 & 1 \end{pmatrix} : x \equiv a^t (\textbf{mod}\, p),\, y \in \mathbb{Z}_p,\, t \in \mathbb{Z}_k \right\} \quad (2)$$

where p is prime, and k is the smallest positive integer such that $a^k \equiv 1 (\textbf{mod}\, p)$.

Definition 3 (*Borel Cayley graph (BCG)* [14]). Let V be a Borel subgroup, and let G be a generator set such that $G \subseteq V \backslash \{I\}$; then $B = \mathfrak{B}(V, G)$ is a Borel Cayley graph with vertices for the 2×2 matrix elements of V and with directed edge from v to u if $u = v * g$, where $u \neq v \in V, g \in G$, and $*$ is a modulo-p multiplication chosen as a group operation.

Note that $n =| V |= p \times k$, where k is a factor of $p - 1$.

C. Node representation

Generalized Chordal Rings (GCR) [20] and the more specialized Chordal Rings (CR) [21], on the other hand, are two existing topologies that are defined in the integer domain and have a systematic and regular structure.

Definition 4 (*Generalized chordal ring (GCR)* [20]). A graph is a generalized chordal ring if its nodes can be labeled with integers mod n, the number of nodes, and there is a divisor q of n such that node i is connected to node j iff node $i + q$ is connected to node $j + q \ (mod\, n)$.

According to this definition, nodes of GCR are classified into q classes, each class with n/q elements. The classification is based on modulo q arithmetic. Two node having the same residue $(mod\, q)$ are considered to be in the same class. Since i connects to j implies $i+q$ connects to $j+q \ (mod\, n)$, nodes in the same class have the same connection rules defined by the *connection constants* or *GCR constants*. When the GCR constants for the different classes are known, connections of the entire graph are defined.

A chordal ring (CR) is a special case of GCR, in which every node has $+1$ and -1 modulo n connections. In other words, all nodes on the circumference of the ring are connected to form a Hamiltonian cycle. A Hamiltonian cycle is a graph cycle through a graph that visit each node exactly once.

This GCR representation is useful for routing because nodes are defined in the integer domain and there is a systematic description of connections. An interesting proposition, given in [14], provided an explicit algorithm to transform any Cayley graph into GCR. We restate this proposition as follows:

Proposition 1 For any finite Cayley graph, $\mathbb{C}$, with vertex set V, and any $T \in V$ such that $T^m = I$, there exists a GCR representation of $\mathbb{C}$ with divisor $q = n/m$ where $n =| V |$.

The element T is refereed to as the transform element and it can be any element in the vertex set V. The elements of the group are then partitioned into q classes by premultiplying the transform element with the representing element of each class a_i.

Any vertex $v \in V$ is represented with T and a_i as follows [13]:

$$v = T^j * a_i \quad (3)$$

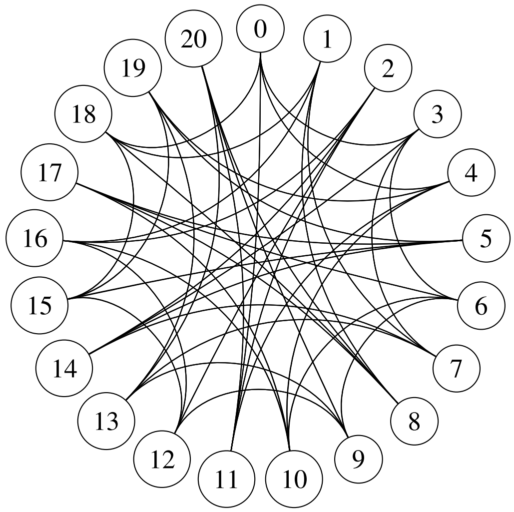

Fig. 1. Borel Cayley Graph with $p = 7, a = 2,$, and $k = 3$

Thus, there is a function that maps the Borel subgroup onto integer domain of GCR numbered between 0 and $n-1$, where n is the number of elements in the Borel subgroup:

$$\begin{array}{rccc} f: & V & \rightarrow & \mathbb{N} \\ & T^j * a_i & \rightarrow & q * j + i \end{array} \quad (4)$$

D. Example (BCG with $p = 7, a = 2,$, and $k = 3$)

Figure 1 shows an example of Cayley graphs over $BL_2(\mathbb{Z}_p)$ with $a = 2$ and $p = 7$. for degree-4 graphs, there are four generators: $\mathbf{A}$, $\mathbf{B}$, $\mathbf{A^{-1}}$ and $\mathbf{B^{-1}}$. Let $\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ and $\mathbf{B} = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$. Note that in this case, we have $k = 3$, $p = 7$, $n = p \times k = 21$ and the diameters is 3. We arbitrarily choose the transform element $\mathbf{T} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ with $\mathbf{T}^7 = \mathbf{I}\ (mod\ p)$ to produce a GCR representation. Furthermore, we choose the representing element of class i to be $\mathbf{a_i} = \begin{pmatrix} a^i & 0 \\ 0 & 1 \end{pmatrix}$, $i = 0, ..., q-1$. Since $m = 7$, the divisor $q = n/m = 3$. Let $\mathbf{V} = \{0, 1, ..., 20\}$. For any $i \in \mathbf{V}$, if $i\ mod\ 3 =$:
"0": i is connected to $i+3$, $i-3$, $i+4$, $i-10$ $(mod\ 21)$;
"1": i is connected to $i+6$, $i-6$, $i+7$, $i-4$ $(mod\ 21)$;
"2": i is connected to $i+9$, $i-9$, $i+10$, $i-7$ $(mod\ 21)$;

E. The power consumption model

Transmit power control has been recently proposed as a technique to increase network capacity and to reduce energy consumption in ad hoc network. Decreasing the nodes' transmission power with respect to the minimum level potentially has two positive effects: (*i*) increasing the nodes' energy consumption, and (*ii*) increasing the spatial reuse, with a positive overall effect on network capacity.

Although, the current literature on topology control has focused attention on energy consumption, trying to adjusting the transmit powers of nodes or to minimizing the energy cost in order to create a desired topology [2] [5]. In these works, the general idea is that increasing spatial reuse by providing upper bounds on the node degree in the final network topology. The rationale for considering node degree is that, if a node has relatively small degree, it will experience relatively low contention when accessing the wireless channel. Thus, the spatial reuse is increased, as well as network capacity. However, since these approaches do not generate a predefined graph, there is no guarantee on the final topologys bounded degree. Furthermore, producing a degree bounded network topology that preserves connectivity in the worst-case is impossible [15]. In other words, the goals of preserving worst-case connectivity and having a nontrivial upper bound on the physical node degree inherently conflict with each other.

Ramanathan and Rosales-Hain [2] propose an ingenious distributed topology control algorithm for transmit power control. For mobile networks, they present two distributed heuristics that adaptively adjust node transmit powers in response to topological changes and attempt to maintain a connected topology using minimum power. The idea in [2] inspires us to let each node adjusting their transmit powers before running our TC algorithm, and thus assuring a minimum network lifetime.

As in [2], each node is configured with two parameters, the 'desired" node degree that is the BCG' degree d_{BCG}, and the current degree d_c that is the number of active neighbors in its neighbor table (built by the routing mechanism). The node in the network attempts to adjust its operational power periodically. If the current degree is greater than d_{BCG}, the node reduces its operational power. If the degree is less than d_{BCG}, the node increases its operational power. If neither is true, no action is taken.

Based on the well-known generic model for propagation [22], by which the propagation loss function varies as some $\mathcal{E}$ power of distance, Ramanathan et al. [2] derive the formula to reduce the power. Specifically, if d_d, d_c, p_c and p_d denote, respectively, the desired degree, the current degree, the current transmit power of a node, and the targeted power. Equation 1 computes transmit power p_d so that the node has the desired degree d_d.

In our system, we use $\mathcal{E} = 4$ and $d_d = d_{BCG}$. Equation 1 can thus be used to calculate the new power periodically. We note that the formula applies for both power increase and decrease to bring the degree close to d_{BCG}.

IV. DISTRIBUTED NETWORK PROTOCOL

Mobility is a prominent characteristic of MANETs, and is the main factor affecting topology changes and route invalidation. Thus, node ID assignment becomes an important issue. An effective assignment allows most connections to be imposed and hence having most of the communication links of the original theoretic graph while preserving a small diameter and average path lengths.

A. Terminologies

We denote the logical node ID of node u in BCG by u_{id} and define the following.

1) $\mathcal{N}(u)$: Set of physical neighbors of node u.
2) $\mathcal{D}(u_{id})$: Set of logical neighbors of node u_{id} in BCG graph.
3) $\mathcal{T}(u_{id})$: Set of logical neighbors of node u_{id} in BCG graph that aren't yet assigned to u's physical neighbors.
4) $\mathcal{L}(u)$: Set of logical IDs of node u's physical neighbors.
5) $\mathcal{F}(v, u_{id}) = |\mathcal{D}(u_{id}) \cap \mathcal{L}(v)|$: the number of logical neighbors of u_{id} that are also logical IDs of node v's physical neighbors.

We define three different packet types.

1) **Token Assignment** packet: forwarded by nodes and used to initiate the distributed node ID assignment algorithm. the node that starts the operation sets an assignment counter A_{Cnt} to N. As the Token Assignment reach node v, A_{Cnt} decreased by one before forwarding the token to the next node, and assignment algorithm ends once A_{Cnt} reaches 0.
2) **Request Info** packet: Used by the node u holding the Token Assignment to broadcast $\mathcal{T}(u_{id})$ to its physical neighbors so as to identify those who do not yet have their ID.
3) **Reply Info** packet: if a physical neighbor node v, without any ID assigned yet, receiving a Request Info packet from node u, it sends a Reply Info packet to node u. The Request Info packet contains a list of $\mathcal{F}(v, x_{id})$ for each x_{id} proposed by u.

B. Algorithm

The distributed node ID assignment involves the following algorithm:

(*i*) Periodically, each node checks the number of active neighbors (degree) in its neighbor table built by routing mechanism. Equation 1 can thus be used to adjust transmit power to bring the degree close to d_{BCG}.

(*ii*) Assignment operation after node u received a Token Assignment packet.

Step 1: Node u received a Token Assignment packet.

1) If any ID u_{id} not yet assigned to u it takes randomly an u_{id} ranging from 0 to $N-1$.
2) Send a Request Info packet to all its physical neighbors so as to identify those who do not yet have their IDs.

Step 2: Node u collects the Reply Info packets.

1) If there is no Reply Info packet received, forward the Token packet randomly to one of its physical neighbors.
2) Having received a list of $\mathcal{F}(v, x_{id})$, $v \in \mathcal{N}(u)$ and $x_{id} \in \mathcal{T}(u_{id})$, from their physical neighbors. Node u grouped them for each proposed $x_{id} \in \mathcal{T}(u_{id})$ and sort each group in descending order. and assign to node v the ID x_{id} that have a larger number of $\mathcal{F}(v, x_{id})$ among all $\mathcal{F}(w, x_{id})$, $v \neq w \in \mathcal{N}(u)$.
3) Node u send to those neighbors their respective new $x_{id} \in \mathcal{T}(u_{id})$ values.

Step 3: Node u sends Token Assignment packet randomly to one of the node newly assigned.

(*iii*) The operation after node v receives a Request Info packet.

Step 1: A node v receives a Request Info packet.

Step 2: Node v calculates $\mathcal{F}(v, x_{id})$, $x_{id} \in \mathcal{T}(u_{id})$ that represent, for each proposed x_{id}, the number of logical neighbors of x_{id} that are also IDs of node v's physical neighbors.

1) if node v already has its ID v_{id}, the Request Info packet is ignored.
2) Otherwise, A node v send those calculated numbers piggybacked in Reply Info packet.

V. SIMULATION

In this section, we describe the setup for our considered scenario. Then, we present our estimation results.

A. Setup

Our method was designed to monitor wireless ad hoc mobile network, thus, we evaluated our topology control algorithm through a discrete-event network simulator (ns-3). The evaluation was done in terms of the following topological and spectral metrics:

- $Diameter$: The longest distance between any pair of nodes of a graph. A shorter diameter is a desirable property for large network.
- $Average\ path\ length$: The average of the shortest paths between all pairs of nodes in a graph.
- R_c: The percentage of the number of edges of the resultant graph over that of the original BCG. That is,

$$R_c = \frac{Number\ of\ edges\ in\ resultant\ graph}{Number\ of\ edges\ in\ original\ BCG} \quad (5)$$

In the simulation, we used 506, 1081 and 2265 nodes networks for BCGs. Table I summarizes the parameters values used for simulations. Parameter p and a determine N (the number of nodes) and BCG parameter k. Parameters t_1 and y_1 were used to define the first generator. Parameters t_2 and y_2 were used to define the second generator. For each network size, 100 distinct samples were collected. Nodes of network models were uniformly and randomly distributed in an area of $100 \times 100\ m^2$. For each radio range from $5m$ to $150m$.

We compared our topology control algorithms with existing topology controls: K-Neigh [15] , which the number of neighbors of node bounded by a giver value k, with $k = 6$, Max topology which all nodes within the maximum radio range are logical neighbors, and Dist-swap topology control based on BCGs [23] with a distributed node ID assignment algorithm (Dist-swap). in this section we name our assignment method: Assignment ID (Assign-ID).

TABLE I
PARAMETERS OF BOREL CAYLEY GRAPHS

N	p	k	a	t_1	t_2	y_1	y_2
506	23	22	7	1	7	1	1
1081	47	23	2	1	7	1	1
2265	151	15	2	1	7	1	1

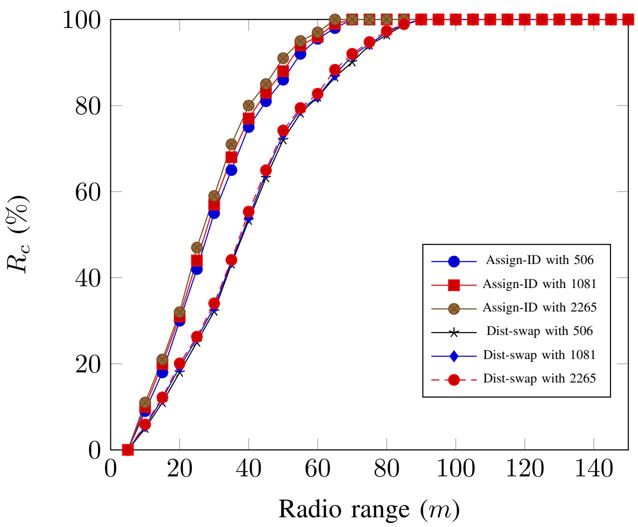

Fig. 2. Edge construction percentage with distributed node ID assignment.

We compute the energy consumed for all the nodes in the network during the multi-hop packet routing environments. We only consider the energy consumptions from data transmission (E_{tx}) and reception (E_{rx}).

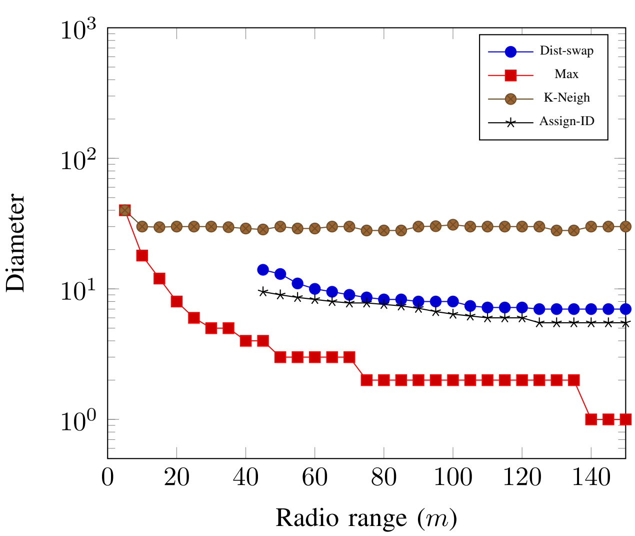

Fig. 3. Comparison of diameters

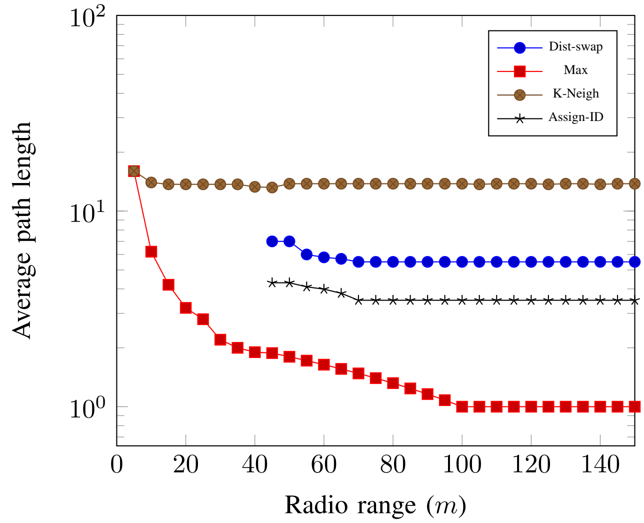

Fig. 4. Comparison of average path lengths

B. Simulation Results

Figure 2 shows R_c of Assign-ID and Dist-swap assignment methods. The Assign-ID shows that R_c is a slightly better than that of the Dist-swap assignment algorithm when the radio range is less than $80m$. For $R_c = 95\%$, the Dist-swap and Assign-ID needed $55m$ and $70m$ respectively. Overall, he Assign-ID showed better performance of R_c.

Figures 3 and 4 show the diameter and average path length versus the transmission range, respectively. We only compared the cases where the resultant graph is a connected graph. Thus, Dist-swap and Assign-ID both require $45m$ radio range to produce a connected network. From these figures, we confirm again that the Assign-ID have the smallest diameter and the shortest average path length for a given radio range except Max topology. This is because as a radio range increases, BCGs have a constant number of logical neighbors, while Max topology has much more logical neighbors than others.

In Figure 5, an interesting observation is that the energy consumed at each node is determined by the average path length and the diameter of the network topologies. Another observation is that the nodal energy consumption for Assign-ID is consistently smaller that that of the k-Neigh with $k = 6$.

Even though Max topology has the smallest diameter and the shorter average path length for a given radio range, due to its large number of logical neighbors, it consumed 60 times more nodal power than Assign-ID in Figure 5.

Max topology has a larger number of logical neighbors and hence larger power is consumed at the receiver modules.

VI. CONCLUSION

In this paper, we proposed the BCG-based topology control algorithm to impose the predefined BCG topology on wireless ad hoc networks. We proposed a new ID assignment method to overly our host network and have in return a new topology with a number of logical connections following predefined graph connection rules in a distributed manner.

 ISBN: 978-1-61208-920-1

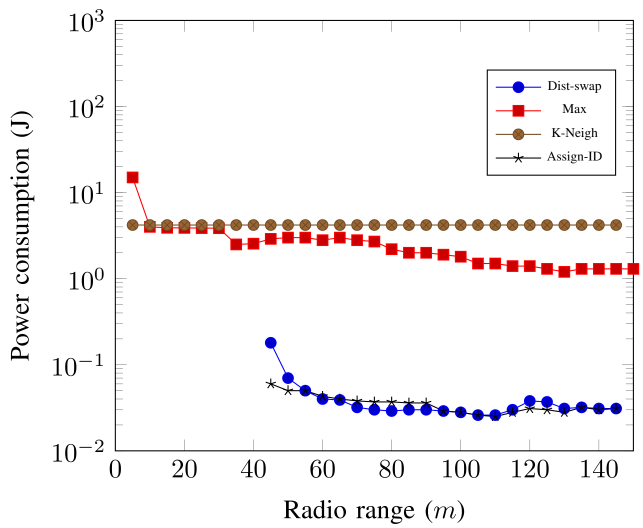

Fig. 5. Comparison of power consumption

Before assigning logical IDs to physical nodes in the network, our algorithm adjust node transmit powers in order to maintain a connected topology using minimum power and produce a topology with physical node degrees that are upper bounded by d_{BCG}. Our simulation results over a range of network sizes 500 and 2300 nodes showed that our node ID assignment method requires a small radio range $45m$ to produce a connected network with a small diameter and a small average path length.

In the future, we plan to expand our investigation to more realistic network environments involving topology control and routing protocol. we will also investigate the effects of interference on BCG-based networks.

REFERENCES

[1] R. Rajaraman, "Topology control and routing in ad hoc networks: a survey," SIGACT News 33, 2, pp. 6073, June 2002.

[2] R. Ramanathan, and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," Proceedings IEEE INFOCOM 2000, Tel Aviv, Israel, pp. 404-413 vol.2, March 2000.

[3] P. Santi, "Topology control in wireless ad hoc and sensor networks," ACM Computing Surveys, vol. 37, no. 2, pp. 164-194, 2005

[4] Li Li, J. Y. Halpern, P. Bahl, Yi-Min Wang, and R. Wattenhofer, "A cone-based distributed topology-control algorithm for wireless multi-hop networks," in IEEE/ACM Transactions on Networking, vol. 13, no. 1, pp. 147-159, Feb. 2005

[5] S. Narayanaswamy , V. Kawadia , R. S. Sreenivas , and P. R. Kumar, "Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW Protocol," in Proc. of European Wireless 2002, Next Generation Wireless Networks: Technologies, Protocols, Services and Applications, Florence, Italy, pp. 156-162, February 2002.

[6] V. Rodoplu, and T. H. Meng, "Minimum energy mobile wireless networks," IEEE J. Selected Areas in Communications, vol. 17, no. 8, pp. 1333-1344, August 1999.

[7] K. N. Sivarajan, and R. Ramaswami, "Lightwave networks based on de Bruijn graphs," in IEEE/ACM Transactions on Networking, vol. 2, no. 1, pp. 70-79, Feb. 1994.

[8] SH. Lu, KC. Li, KC. Lai, and YC. Chung "A scalable P2P overlay based on arrangement graph with minimized overhead," Peer-to-Peer Netw. Appl. 7, pp. 497510, 2014.

[9] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," ACM SIGCOMM Computer Communication Review, vol. 31, no. 4, pp 149160, 2001.

[10] H. Shafiei, Z. Aghazadeh, A. Khonsari, and M. Ould-Khaoua, "PeerStar: An attractive alternative to existing peer-to-peer topologies," 2009 IEEE Symposium on Computers and Communications, pp. 128-134, 2009.

[11] A. Abu Taleb, J. Mathew, and D. K. Pradhan, "Fault diagnosis in multi layered De Bruijn based architectures for sensor networks," 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 456-461, 2010.

[12] A. A. Al-Mamou, and H. Labiod, "ScatterPastry: An Overlay Routing Using a DHT over Wireless Sensor Networks," The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007), pp. 274-279, 2007.

[13] K. Tang, and B. Arden, "Representations of Borel Cayley graphs," in The Fourth Symposium on the Frontiers of Massively Parallel Computation, McLean, VA, USA, pp. 194-201, 1992.

[14] B.W. Arden, and K.W. Tang, "Representations and routing for Cayley graphs," (computer networks), IEEE Trans. Commun. 39 (1991), pp. 15331537.

[15] D. M. Blough, M. Leoncini, G. Resta, and P. Santi, "The k-Neighbors Approach to Interference Bounded and Symmetric Topology Control in Ad Hoc Networks," in IEEE Transactions on Mobile Computing, vol. 5, no. 9, pp. 1267-1282, Sept. 2006.

[16] R. Wattenhofer, and A. Zollinger, XTC: a practical topology control algorithm for ad-hoc net-works, in 18th International Parallel and Distributed Processing Symposium, 2004. Proceedings, 2004, pp. 216223.

[17] N. Li, J. Hou, and L. Sha, Design and analysis of an mst-based topology control algorithm,IEEE Transactions on Wireless Communications, vol. 4, no. 3, pp. 11951206, 2005.

[18] Z. H. Mir, and Y. Ko, Adaptive neighbor-based topology control protocol for wireless multi-hopnetworks,EURASIP J. Wirel. Commun. Netw., vol. 2012, p. 97, 2012.

[19] S. Narayanaswamy, V. Kawadia, R. Sreenivas, and P. Kumar, Power control in ad-hoc networks:Theory, architecture, algorithm and implementation of the compow protocol,Proc. of theEuropean Wireless Conf., 01 2002.

[20] J.-C. Bermond, C. Delorme, and J.-J. Quisquater, "Strategies for interconnection networks: Some methods from graph theory," Journal of Parallel and Distributed Computing, Vol. 3, no. 4, 1986, pp. 433-449.

[21] Arden, and Hikyu Lee, "Analysis of Chordal Ring Network," in IEEE Transactions on Computers, vol. C-30, no. 4, pp. 291-295, April 1981.

[22] T. S. Rappaport, Wireless Communications, Principles and Practice, Prentice-Hall, 1996.

[23] R. Junghun, Y. Jaewook, N. Eric, and K. W. Tang, "Borel Cayley Graph-Based Topology Control for Consensus Protocol in Wireless Sensor Networks," ISRN Sensor Networks, vol. 2013, pp. 1-15, 2013.