DSCTP Congestion Control Algorithm Based on Dynamic Policies

Jie Chang, Wen'an Zhou, Junde Song, Feng Yu, Zhiqi Lin
Beijing University of Posts and Telecommunications, China
cjbupt@gmail.com, zhouwa@bupt.edu.cn, jdsong@bupt.edu.cn, yfbupt@gmail.com, linzhiqi07@gmail.com

Abstract—**This paper introduces DSCTP (Dynamic Stream Control Transmission Protocol), a sender-side, transport-layer protocol that modifies the standard Stream Control Transmission Protocol (SCTP) protocol. Although SCTP provides support for multi-homing，the basic reason for such a provision was to improve reliability of associations, simultaneous transfer of new data to multiple paths is currently not allowed in SCTP. DSCTP adopts SCTP's multi-homing feature to distribute data across multiple end-to-end paths in a multi-homed SCTP association. DSCTP aims at exploiting congestion control algorithm of Transmission Control Protocol (TCP) and SCTP. Through the use of dynamic policy management framework, DSCTP switches the transmission onto the alternate path using DSCTP's flexible path management capabilities. We can gain significant throughput improvement if simultaneously transfer new data across multiple paths to the receiver. In this article，these techniques include transmission start, flow control, network monitoring, generation of policies, routing switching, and congestion recovery. Extensive simulations under different scenarios highlight the superiority of the proposed solution with respect to TCP and the standard SCTP implementation.**

Keywords-DSCTP; dynamic policy management framework; SCTP; congestion control

I. Introduction

Computer networks have experienced an explosive growth over the past few years and with that growth have come severe congestion problems [17]. In recent years, the portfolio has been surging as the network used widely. Especially in recent years, the development of IP telephone (VoIP) and Internet Protocol Television (IPTV), transferring voice, video and multimedia information in the Internet becomes inevitable. The core problem is the transmission of multimedia data services and real-time communications, and how to provide a certain quality of services for these services. From the protocol of the transport layer, the traditional transport protocols, TCP provides both reliable data transfer and strict order-of-transmission delivery of data. Some applications need reliable transmit without sequence maintenance, while others would be satisfied with partial ordering of the data. In both of these cases, the head-of-line blocking offered by TCP causes unnecessary delay [10]. User Datagram Protocol (UDP) lacks reliable guarantee mechanism for the transmission, and because it has no congestion control mechanism, so the unfair competition for bandwidth can causes network congestion even collapse [1]. In recent years, SCTP protocol was proposed by IETF, called as a modified TCP protocol, which has both advantages of TCP and UDP. Congestion control is one of the basic functions of SCTP. For some applications, it may be likely that adequate resources will be allocated to SCTP traffic to ensure prompt delivery of time-critical data -- thus, it would appear to be unlikely, during normal operations, that transmissions encounter severe congestion conditions. However, SCTP must operate under adverse operational conditions, which can develop upon partial network failures or unexpected traffic surges. In such situations, SCTP must follow correct congestion control steps to recover from congestion quickly in order to get data delivered as soon as possible. In the absence of network congestion, these preventive congestion control algorithms should show no impact on the protocol performance [11]. Due to the bottlenecks of the current network equipment handling capability, service needs a lot of data transfer currently, so it often results in the status of network congestion. Although conditions can be alleviated by improving hardware, due to the limit of the development of hardware manufacturing technology and economic costs, frequent replacement of hardware is usually unrealistic and not a long-term plan. Therefore, only from the perspective of improving network congestion control to improve the network condition is feasible.

Congestion control is a method used for monitoring the process of regulating the total amount of data entering the network so as to keep traffic levels at an acceptable value. This is done in order to avoid the telecommunication network reaching what is termed: congestive collapse. Congestion control mostly applies to packet-switching network. A wide variety of approaches have been proposed, however the "objective is to maintain the number of packets within the network below the level at which performance falls off dramatically" [16].

Congestion control is usually focused on the design of transport layer protocol, like TCP. However, whether TCP or SCTP are not have desired performance. It is necessary to provide a new transmission protocol to meet the needs of real-time data for streaming media. The remaining sections of paper are aimed to apply dynamic policies into SCTP protocol. With a comprehensive analysis of the characteristics of the traditional congestion control mechanism and to fully exploit the own characteristics of SCTP, and based on the idea of dynamic policy management, a modified SCTP protocol – DSCTP was proposed. DSCTP is not a new protocol, only to increase the dynamic SCTP congestion controls mechanisms, so that it can always be adjusted in data transmission according to the network environment. Section II overviews several ideas and mechanisms used by SCTP; some are compared with TCP and UDP to highlight similarities and differences. Section III formulates the characteristics and the defect of SCTP congestion control mechanism. Section IV formulates a refined dynamic policy management framework based on Ponder model. In Section V, the framework is applied into SCTP and describing dynamic congestion control algorithm of DSCTP. In Section VI, simulation results show DSCTP dynamic congestion control protocol can not only reduce the risk of overall network congestion occurs, but also improve the overall efficiency of the network.

II. Overview of SCTP

In this section, we provide an overview of the protocols that we are using in this paper: SCTP [14]. SCTP is defined in RFC2960 [14] with changes and additions included in the Specification Errata [15].

A SCTP versus TCP and UDP

Today most applications use either TCP [10] or UDP [1]. Applications that need a reliable in-order delivery of the bytes sent by its peer uses TCP, whereas ones that can tolerate a certain degree of loss prefer UDP, primarily because UDP provides speedier delivery of packets. Most applications prefer TCP over UDP and applications use TCP including file transfer applications, electronic mail and the worldwide web. UDP is used by streaming audio/video applications for which timely delivery is more important than reliability. SCTP was recently adopted by IETF, and is a reliable transport protocol that operates on top of a connectionless packet based network such as IP. It was originally designed to be a general purpose transport protocol for message oriented applications, as is needed for the transportation of signaling data. SCTP is a transport layer protocol and its services are at the same layer as TCP and UDP. Instead of the three phase connection setup for TCP and best effort to delivery for UDP, the initialization of an association is completed after the exchange of four messages. Another important difference between SCTP and TCP is the support for multi-homed nodes in SCTP, i.e. nodes which can be reached using more than one IP addresses [13]. So if the routing is configured in such a way that these IP addresses are accessible through different paths, multi-homing gives SCTP a certain network level fault tolerance.

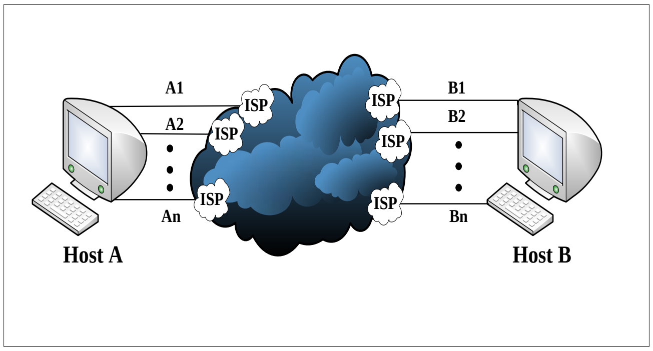


Figure 1. Example Multi-homed Topology

Unlike TCP and UDP, SCTP supports multi-homing at the transport layer to provide redundancy at the path level, thus increasing association survivability in the case of a network path failure [8]. An SCTP endpoint may bind to multiple IP addresses during association initialization. As the Figure 1 shows, we can contrast SCTP with TCP to further explain SCTP's multi-homing feature. There are n^2 distinct TCP connections are possible between Hosts A and B: (A_1,B_1) , (A_1,B_2) , (A_1,B_3) ,…, (A_1,B_n) , (A_2,B_1),…, (A_2,B_n),…, (A_n,B_1),…, (A_n,B_n) .SCTP congestion control algorithms are based on RFC 2581 [12], and include Selective Acknowledgement (SACK)-based mechanisms for better performance. Similar to TCP, SCTP uses three control variables: a receiver's advertised window (RWND), a sender's congestion window (CWND), and a sender's slow start threshold (SSTHRESH).

At association startup, a primary path is defined for each SCTP endpoint, and is used for normal sending of SCTP packets. The idea of SCTP load balancing is to use only primary path to transfer. Because the congestion control mechanism of SCTP is similar to TCP, apply congestion control mechanism to load sharing, there would be unnecessary fast retransmit problems.

A single SCTP association (session), is able to use alternatively anyone of the available IP-addresses without disrupting an ongoing session. However, this feature is currently used by SCTP only as a backup mechanism that helps recovering from link failures. SCTP maintains the status of each remote IP address by sending Heartbeat messages and it is thus able to detect a specific link failure and switch to another IP address. Another novel feature is that SCTP decouples reliable delivery from message ordering by introducing the idea of streams. The stream is an abstraction that allows applications to preserve in order delivery within a stream but unordered delivery across streams. This feature avoids HOL blocking at the receiver in case multiple independent data streams exist in the same SCTP session. Congestion control was defined similar to TCP, primarily for achieving TCP friendliness [14].

B SCTP Reseach Activities

SCTP is standardized in the IETF first in the Signaling Transport Work Group (SIGTRAN WG) and since 2001 it has found a new home in the transport Area Work Group (TSV WG). The Protocol Engineering Laboratory (PEL) directed by Professor Paul Amer at the University of Delaware is dedicated to the research, development, and improvement of new and existing computer network protocols. PEL researchers are investigating innovative transport protocol alternatives to TCP and UDP (such as SCTP) emphasizing these alternatives within army networks to provide efficient communications under mobile, ad-hoc network conditions [9]. The ongoing development of alternative transport protocols (e.g., SCTP) which provide several benefits over traditional transport protocols such as

TCP and UDP, especially in supporting army and/or multimedia applications. Current focus is on transport layer multi-homing and multi-streaming.

III. SCTP CONGESTION CONTROL

A Compare of Network congestion and congestion control

Before SCTP congestion control mechanisms are discussed, we must first clear and definite two concepts: network congestion and congestion control.

When an excessive number of packets reach a certain part of the communication network, there is no time for this part of network to deal with all the data, resulting in decreasing the network performance, and even lead to a suspend of the network communication services, which produce the network congestion. Congestion control has no exact definition, its purpose is to use some means of avoiding the network congestion situation, while in case of congestion the state can resume as soon as possible.

The reason may be due to limited storage space as node cannot cache all the receiving packets and the packet loss or due to data units arrive at the number far exceeds of node's processing capacity and cause delay, and the fundamental reason is that network data traffic flow and node processing speed.

B Congestion Control in TCP

TCP and UDP are the most common IP network transport layer protocol, TCP has its own congestion control mechanism, while UDP has not, although some scholars in recent years have been studying how to add UDP congestion control mechanism, but this is not in the discussion areas.

In general, TCP congestion control mechanism can be divided into open-loop control and closed-loop control of two kinds. Open-loop control focus on prevention, hopes to avoid congestion by perfect design, closed-loop control is solution-focused, trying to relieve and control after the congestion occurs. Therefore, TCP congestion control algorithm designed mainly from two aspects, the basic TCP congestion control algorithms will include: slow start, congestion avoidance, fast retransmit and fast recovery [12].

1) Slow start and congestion avoidance: TCP congestion control requires two main parameters, Congestion Window (CWND) and Slow-Start Threshold (SSTHRESH). Sender sends window SWND = min (CWND, RWND), RWND is the receiver window and it usually has little impact on the send window. Therefore CWND directly determine the size of the send window, and the size of send window directly determine the size of the send packet. CWND = 1 when data sent initially, when each sent data packet is successfully confirmed, CWND will be increased by 1. If all the data packets in the send window have been identified, CWND will be double. The growth process of the congestion window is called slow start; in fact, in the slow start phase, CWND in the ideal case is exponential growth. However, CWND cannot be unlimited growth, we must set a threshold SSTHRESH for it, if CWND> = SSTHRESH, CWND can be increased by 1 only when all the data packets in the send window have been confirmed. This time is called the congestion avoidance phase, CWND increases linearly.

TCP will set a retransmission timeout RTO when a data packet is sent, when the round-trip time RTT exceeds RTO's setting time, it means that this packet is lost, and has to retransmit this packet, which is called network congestion. At this point, the network come into congestion recovery phase, TCP congestion control algorithm shall do the following:

Set CWND = min (4*MTU, max (2*MTU, 4380byte)) [14] [15];

SSTHRESH = RWND;

Set SSTHRESH = max (CWND/2, 4*MTU);

Re-enter slow start phase

2) Fast retransmit and fast recovery: As waiting for the RTO will make the network transmission efficiency decreased, fast retransmit mechanism would like to find a way to search for an alternative way of congestion discovery. Whenever an endpoint receives a SACK that indicates that some TSNs are missing, it should wait for two further miss indications (via subsequent SACKs for a total of three missing reports) on the same TSNs before taking action with regard to Fast Retransmit. And then enter the congestion recovery phase, during this period the system shall do the following:

Set SSTHRESH = max (CWND/2, 4*MTU);

Set CWND = max (CWND/2, 4*MTU) +3;

Enter the congestion avoidance phase directly;

This is the basic algorithm of fast recovery. During the period of congestion recovery phase, using fast recovery to replace slow start is known as TCP congestion control algorithm.

C SCTP Congestion Control Mechanism

SCTP protocol is called a modified TCP protocol, which has the characteristics of both TCP and UDP. SCTP is connection-oriented in nature, but the SCTP association is a broader concept than the TCP connection. The term "stream" is used in SCTP to refer to a sequence of user messages that are to be delivered to the upper-layer protocol in order with respect to other messages within the same stream. This is in contrast to its usage in TCP, where it refers to a sequence of bytes. TCP guarantees in-sequence delivery of data to its upper-layer protocol within a single TCP session. This means that when TCP notices a gap in the received sequence number, it waits until the gap is filled before delivering the data that was received with sequence numbers higher than that of the missing data. On the other hand, SCTP can deliver data to its upper-layer protocol even if there is a gap in TSN if the Stream Sequence Numbers are in sequence for a particular stream (i.e., the missing DATA chunks are for a different stream) or if unordered delivery is indicated. Although this does not affect CWND, it might affect RWND calculation. The biggest difference between SCTP and TCP, however, is multi-homing.

Because the multi-homing nature, the unique nature of SCTP, its congestion control mechanism is different from TCP. Once the congestion happened on the link, the transmitter will choose another IP address with the same host to continue the transmission. As SCTP provide the connection oriented service, and the connection for each IP address has already established, there is no need to setup a new connection. In this way the overhead of setting up and releasing the connections can be minimized to a large extent. In traditional way of STCP, the slow start and the congestion avoidance phases are still needed by the link, however, because of using SACK in SCTP to confirm, the transmitter only need to retransmit packets which have not be confirmed when the link enter the fast recovery phase after the congestion happened. SACK changes the acknowledge mechanism of TCP, TCP only confirm packets that all ready received, while SACK would send the acknowledgment which contain the disordered information to the receiver, by doing this the transmitter will minimize blindness of the retransmission.

D The Defect of the Traditional Congestion Control Mechanism

The traditional method of congestion control emphasis on closed-loop control, but open-loop control has not much achievement. When dealing with congestion control, both of TCP and SCTP take the way of "congestion discovery"—a kind of mechanism for congestion control. Although SCTP reduces the blindness of retransmission by using the SACK confirmation method, it can still not get rid of hysteresis due to passive wait for the congestion. The method of congestion control is too simple, the initial start is so slow and some other problems are all have defects. Besides, TCP and current SCTP use only one destination address at any given time to transmit new data. SCTP restricts sending new data, which can act as probes for information (such as available bandwidth, loss rate, and RTT) to only one primary destination. Consequently, an SCTP sender has minimal information about other paths to a receiver [7].

In order to improve these deficiencies, this paper presents a SCTP-based congestion control algorithm. The protocol which uses of this new congestion control algorithm is called dynamic SCTP (DSCTP). DSCTP fully exploits SCTP feature, also it is flexible, proactive and accurate to carry on the congestion control. More importantly, it can adjust its congestion control policies according to network conditions before the congestion happen. DSCTP is not a new protocol, it only improves SCTP congestion control mechanism, and the basic idea comes from the dynamic policy management. DSCTP uses multiple destinations simultaneously, CWND growth in DSCTP demands tracking the latest TSN received in order per destination, information not coded directly in a SACK. On the other hand, a DSCTP sender maintains more accurate information about all paths, since new data are being sent to all destinations concurrently. This information allows a DSCTP sender to better decide where to retransmit.

IV. DYNAMIC POLICY MANAGEMENT

Policy is a set of rules for the management and control of network resources. The essence of policy based network control is to view network as a finite state machine, the node within the network is assigned a specific state according policy rule [2].

Traditionally static policy is generated by the PMT [3] through the administrator, while the dynamic policies are generated based on the parameters which are collected by the network, feedback mechanisms and policy algorithms. Networks are controlled by dynamic policies, which fluctuate accompanied by the changing environmental message, feedback from contributing factors, so that dynamic policies are self-adjust and can reflect the real network environment.

Policy repository should be maintained from instant to instant; policies are added, deleted or modified upon the alteration of network environment. Real-time network management can be achieved automatically without intervention of human network operators. This is a big virtue of dynamic policy management process. Dynamic policy management makes full use of network resources, upgrade efficiency and guarantee regular service provision.

Dynamic policy management framework is based on IETF policy management framework [4] and PONDER policy framework. The framework clearly describes how the dynamic policy management network works. In Fig. 1, the following module is shown:

PMT (Policy Management Tool): It provides a visual management interface, the policy administrator through PMT to add, modify and delete policies.

PB (Policy Base): It stores policy.

PEP (Policy Enforcement Point): Execution of specified actions, for example: to request policy, to update policy, to delete policy.

DPS (Dynamic Policy Scheduler): Centralized management and scheduling of policy.

DPIP (Dynamic Policy Information Point): It collects environmental information continually and uses this information to update policy.

DFQB (Forecast Query Builder): It predicts environmental information and assists DPIP to create new dynamic policies.

DPC (Dynamic Policy Cache): The policy pre-distributed by PDP is stored in the policy cache.

DPS (Dynamic Policy Self-management): It detects conflicts between policies.

PEP (Policy Enforcement Point): It executes of specified actions.

PDP (Policy Decision Point): It is composed of trigger detection and handling, rule location and applicability analysis, network and resource-specific rule validation and device adaptation functions.

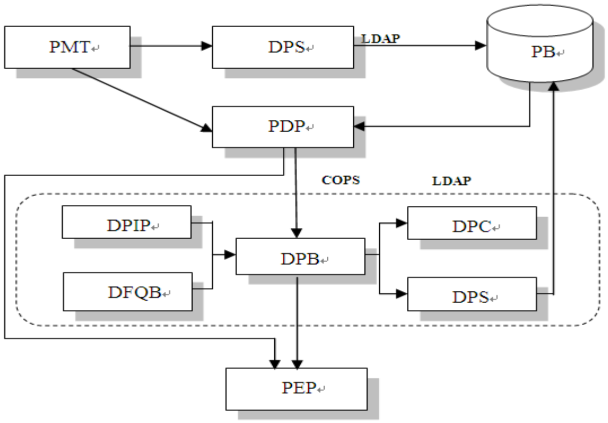


Figure 2. Dynamic Policy Management Framework [5]

PMT dynamically stores policy objects to PB through DPS and maintains PB. When it is necessary, PMT passes policy objects to PDP. PDP can obtain the corresponding policy object from PB according to the policy request submitted by PEP. DPIP continuously collects environmental information, with the assistance of DFQB, in order to adapt to the ever-changing network environment. Then the collected information is passed to PDP to produce new dynamic policies. DPC is used to store policies which are issued in advance by PDP to the DPC. Finally DPS detects policy conflicts, then, it will place policies in PB if no conflict is detected. And then DPB will pass the corresponding dynamic policy rules to PEP. DPS takes a centralized management and scheduling upon policies, taking advantage of the idea of scheduling algorithm. The conventional policy storage is putting policies directly into PB. When request for policies arrives, PDP will search for the suitable policies by doing comparison one policy by one policy. It is such a difficult procedure when it comes to large-scale PB, consuming a long time to search out one policy. On contrary, DPS can centralized manage and schedule policies so that it can decrease the conflicts and increase the efficiency [6].

V. Dynamic Algorithm Description

DSCTP uses dynamic congestion control algorithm which describes the congestion control mechanism in detail in respect of transmission start, flow control, network monitoring, generation of policies, routing switching, and congestion recovery. First, we appoint sender as A, receiver as B, A and B jointly appoint a number of IP address $\{A_1, A_2, ..., A_n\}$ $\{B_1, B_2, ..., B_n\}$. If the service X is intended to send data from A to B, we assume that host A_1 is selected as the sending host.

A Transport Start Phase

Compared with the traditional SCTP, the major improvements of DSCTP are listed in the following aspects：

- When A_1 is intend to send data, firstly, select a host B_i randomly.
- If A_1 had never sent data to B_i, A_1 sends a request REQ to PDP, and REQ includes the current service type X as well as the environmental parameters $PARAM_1, ..., PARAM_n$ collected from the DPIP; If A_1 has ever sent data to B_i, A_1 sends a request REQ to PDP, and REQ includes the previous value of CWND and SSTHRESH in addition to the current service type X and the environmental parameters collected from the DPIP.
- PDP receives REQ, and if A_1 is the first time send data to B_i, PDP sends a query request to PB to query the matching policies. If PDP finds the appropriate policies in PB, then PB returns PDP a policy packet which includes the value of CWND and SSTRESH currently should be set. If A_1 is not the first time to send data to B_i, firstly, PDP queries the policies in the DPC, and if the types of service are the same, then DPC returns the policies to the PDP; else, PDP queries the policies in the DPB, if the match is found, then DPB returns the policies to PDP; otherwise, PDP queries the policies in the PB, if find the appropriate policies, then PB returns it to PDP. The final policies PDP obtains should be assertion whether it is suitable for A_1 send data to B_i.
- If PDP can obtain the return policies, it will further modify the policies according to its own special rules, and return the final result to A_1; if PDP cannot obtain the return the policies, then PDP will return A_1 a policies in accordance with the default rule.
- If A_1 is the first time to send data to B_i, A_1 will set its own CWND and SSTRESH according to the policies received. If A_1 is not the first time to send data to B_i, and return the assertion of policies is TRUE, that means A_1 can send data to the current B_i, and send data according to the value of CWND and SSTRESH before; otherwise, A_1 selects other

IP address to resend to PDP and repeats the process before.

B Congestion Avoidance Phase

In the control of sending quantity of packets, DSCTP also uses TCP and SCTP mechanism. In other words, use CWND and SSTHRESH control of numbers of packets. CWND and SSTHRESH in a similar way with slow start and congestion control. The only difference is CWND not start with 1. Whenever CWND is less than SSTHRESH, an SCTP endpoint must use slow-start algorithm to increase CWND only if the current congestion window is being fully utilized. When CWND is less than or equal to SSTHRESH, Only when all of packets are confirmed can CWND be increased; otherwise, CWND must not be increased.

C Network Monitoring Phase

Detection of network conditions is a core operation of dynamic adaptive network. A has to constantly monitor the link status of each B_i. It should need to take appropriate actions or continue to complete sending segment of packet or continue to send packet after switching IP address when certain link of B_i is found of congestion. Detailed process is described as follows:

- The request of determining the status of connection should be sent from A to PDP, which includes of its CWND and SSTHRESH.
- The value of M should be calculated according to certain rules of CWND, SSTHRESH, environmental parameters of DPIP and forecast parameters of DFQB. M is a number from 0 to 10, which indicates the level of the current status of connection. The greater the value of M indicates that the worse the status of connection should be, and then the less appropriate the data should be sent.
- PDP should get the value of M, then according to the current type of service and other special rules which determine the level of the current link status. In addition, PDP should write the new policies into DPB. The new policies should include the current environment parameters, such as the value of CWND and SSTHRESH and the level of current links. The meaning of the generation of policies should be that the level of link environment is LEVEL in the current environment when all the environmental parameters do not exceed the current value, CWND and SSTHRESH also do not exceed the current value.
- Write the current generation of policies into DPC and replace the original policies. The latest policies should be saved in DPC forever. If the next is the same type of service, check whether the matching policies of DPC first, because the policies of DPC is closest to the current network status.
- PDP should return a result to A_1 according to the calculated M and its own special rules whether the current link for sending data.

D Dynamic Policy Generation Phase

We explain the dynamic generation of policies. PDP can obtain the corresponding policy object from PB according to the policy request submitted by PEP. Then collect environmental information by DPIP and dynamically determine threshold M.

1) Add Global Load Map: In order to elevate efficiency in the bi-driven algorithm [18], an entity called Global Map (GM) is added to the receiver, indicating the each host's instantaneous load value of the system, such as server load, server busy factor, number of active connections, server hardware and so on. The load value is calculated in accord with the pre-defined rules, taking into weighting factors such as the number of pending requests, request types, host processing capacities etc. The algorithm should be executed as follows. New requests should be assigned to light-load nodes according to pre-defined rules. Meanwhile, all host load values are polled; when one idle host is detected, an uncompleted task in high load value host will be assigned to idle host.

2) Dynamically Determine Threshold M: Dynamic scheduling algorithm use dynamic policy management framework to determinate the threshold M dynamically. DPIP collects performance parameters of every host. When new request comes, the load balancer makes a request to PDP and then should contain operations described as follows:

- The value of each parameter will be mapped to a value between 0 and 10, then according to the importance of each parameter of the system calculate the weighted average value, this is the value of the current system performance:

  $$v' = \frac{v - \min_A}{\max_A - \min_A} * 10$$

  where $\min_A$ is the minimum value of parameter, $\max_A$ is the maximum value, v is the current value and v' is the standard value.

  $$F = (f_1, f_2, \ldots f_n)$$

  where f_i is the weighted average value and $f_i = \mu * v'$. μ is the weight value expressing the importance of parameter. F is the standardization of vector.
- According to F, it would be easy to find w in PB, which is pre-defined.
- According to the vector of λ in DFQB, M value would be calculated:

  $$m_{(t,i)} = \sum_{\alpha \in A} \varphi(\alpha) * \omega_{(t,\alpha)}$$

$m_{(t,i)}$ represents the performance value of host i at the time t, $\varphi(\alpha)$ represents mapping function of attribute α . $\omega_{(t,\alpha)}$ is the weight value of attribute α at the time t.

$$\lambda_{(t,i+1)} = \lambda_{(t,i)} \left| m_{(t,i)} - m_{(t-1,i)} \right| \Big/ \sum\nolimits_{i=1}^{n} \left| m_{(t,i)} - m_{(t-1,i)} \right| \lambda_{(t,i)}$$

$\lambda_{(t,i)}$ represents the parameter that effects the system of host i at the time t.

$$\mathrm{M}_t = \mathrm{M}_{t-1} + \sum_{i=1}^{n} \lambda_{(t,i)} \left(m_{(t,i)} - m_{(t-1,i)} \right)$$

where M_t is the threshold of the system. As it can be different in the effect to the whole system when the performance of each host changes, we have to get a average value as the threshold for this system.

E Routing Switch Phase

When A receives policies to change routing, the link appears in a congestion situation, then the routing must be switched.

On the one hand the transmitter should continue to send data to the original address, but on the other hand the process above described should be carried out again when it is in the sending process of a data segment, Once it selected an appropriate IP address B_k, then it should immediately stop sending data to B_i and resend all packets to confirm of the current send window, it also need to send a notification along the original link. The passed node should clear the cache of packets from A_1 after it received notice. Then it will send the remaining data of this data segment to B_k. From A to B_i, CWND and SSTHRESH should not be changed and should be retained.

F Congestion Recovery Phase

The dynamic policy generation is based on the forecast and it is characterized by bias. When the congestion occurs, the congestion recovery mechanism of DSCTP is relatively simple and it only waits for enough time to recovery, then send the packet according to the algorithm of DSCTP. It will not affect network communications even when the waiting time is long enough because there are several routings to choose for SCTP, so the sending endpoint can wait for a long time before sending packet when there has congestion. In addition, after congestion, all nodes on the congested link will clear part of the buffer after receiving instructions. Therefore it can be assumed that it makes the link A to link Bi return to normal after the waiting time. Then the data can be sent by the algorithm of DSCTP. Before congestion occurs DPB should write the policies into PB (It should be the same as the tactics stored in DPC) and set the network status into poor. As this policy was originally wrong, so this policy should be removed from the DPB and DPC. This is the process of dynamic error correction.

G Algorithm Analysis

Compared with the traditional SCTP, the major improvements of DSCTP are listed in the following aspects：

- There are more than one transport addresses in the active state that can be used as a destination address to reach that endpoint, so they can share the network load. The traditional SCTP usually uses the same destination address until being instructed by the upper layer to do, otherwise SCTP may change to an alternate destination in the event an address is marked inactive [11]. However, DSCTP can send data to multiple IP address simultaneously, and no matter data is sent to any one IP address, all of which can be correct confirmation, thus this can greatly enhance the flexibility and the overall processing capabilities of the network.
- According to multiple network parameters, DSCTP can implement the congestion control. The traditional congestion control in SCTP only concerned with some congestion control parameters, such as Congestion Window (CWND), Slow Start Threshold (SSTHRESH), Round-trip Time (RTT) and so on. The monitor of congestion appears very rough. However, DSCTP not only concern the above parameters, but also implement congestion control by environmental parameters. In fact, many environmental parameters can be an indicator to some degree: such as the processor payload of the receiver, the size of the cache, the total number of streams on the current transmission link, the total number of router on the link and the CPU used in the router and so on. All of these parameters may indicate a certain level of congestion.
- DSCTP monitor the network status， and also feedback information to predict the status of link. To avoid network congestion, CWND should be incremented by 1*MTU per RTT if the sender has CWND or more bytes of data outstanding for the corresponding transport address. Or passive monitoring the congestion status and then rapid recovery from congestion state through SACK. To predict the possibility of congestion in DSCTP. As potential congestion, if they predict potential congestion, they should take appropriate action to ensure congestion avoidance immediately.
- Like TCP and SCTP, a DSCTP endpoint uses the following three control variables to regulate its transmission rate, such as RWND, CWND and SSTHRESH. SCTP and DSCTP also require one additional control variable, partial_bytes_acked, which is used during congestion avoidance phase to facilitate CWND adjustment. Beginning data

transmission into a network with unknown conditions or after a sufficiently long idle period requires SCTP to probe the network to determine the available capacity. Through the phases of the slow start and the congestion avoidance, to make the network best transport efficiency, but some urgent high-speed data services have to experience this slow process. DSCTP maintains these modes of the slow start and congestion avoidance, but the value of the initial congestion window is not equal to 1 and after recovery from congestion, according to the state of the sending endpoint and the current network environment, the values of CWND and SSTHRESH are not fixed and by all of these environment parameters to define.

- All network activities in DSCTP are in the guidance of dynamic policies. Based on the service characteristics, the requirements of the network and so on, the network administrators create enough permanent policies in advance, which are widely used in the vast majority situations of the network. And also through the feedback mechanism and prediction mechanism to create temporary policies, which are effective for a given period of time because they collect the current network environment parameters. These policies can either be replaced by new dynamic policies, or be deleted by themselves at the end of the life period.

H Deficiency of DSCTP

- DSCTP cannot avoid congestion. In fact, any kind of congestion control mechanism could not completely avoid any congestion. It can only try to reduce congestion. At the same time, it can ensure implementations of fast-retransmit and fast-recovery as fast as possible. The essence of DSCTP is to see the receiving endpoint as a cluster system, by adding dynamic scheduling algorithms to make the network load more balancing. By this way it can increase the processing capacity of the receiving endpoint, it can also reduce the load of the single host. The ultimate goal is to avoid congestion phases. However, when the network load is big enough, DSCTP cannot avoid this situation.
- DSCTP cannot pay much attention to congestion recovery. As the above mentioned, DSCTP mainly focused on how to avoid congestion, but to the question of how to recovery from congestion, it only obtains the values of CWND and SSTHRESH, it does not provide other effective solutions. This is largely based on DSCTP is multi-homed, for DSCTP, it can has multiple links at the same time, if one of the link has not send data for a long time, it would not have too much influence on the network. So if the link has some problem, it should be wait enough time for the network recovery, not pay much attention on how to fast-recovery from congestion.
- All endpoints in DSCTP must reserve some resources to tackle additional communication information all the time, such as environmental parameters report and delivery, policy request and policy delivery and so on. This extra cost will definitely reduce the efficiency of the transport network, especially when the network load is huge, it will make the network environment more terrible. However, resource reservation algorithm in DSCTP is different from the traditional resource reservation algorithm. It does not need all the resources for transferring data in network, just keep small part of the resources to handle control events. If this method can greatly reduce the probability of congestion is also worth it because congestion in the transport network will lead to low-level efficiency.

VI. SIMULATION TEST

In order to verify the dynamic algorithm for the congestion control is more efficient than the traditional congestion control algorithm, not only can greatly reduce congestion occurred, but also can improve the efficiency of the network transmission. The system testing must be performed using VS2008 simulation for two kinds of algorithms of congestion control by comparing two mechanisms of sending the same quantity of packets, from congestion numbers (CN) and experimental time (ET). EV is used to simulate values of network conditions.

A Experimental condition:

To simplify the protocol we should make some hypothesis:

- Only concerned with receiving buffer occupancy, hops from source to destination and round-trip time three environment variables, and ignored other environment variables.
- The service of sending data in the network belongs to the same type.
- Using one-to-many communication mode, which is a source point correspond to multiple destination points. The sending endpoint send data to any one of the receiving endpoint and get the confirmation are all represented the success of sending data.
- In the real network, it often occurs that too large data traffic can cause congestion in any subnet. Adding a variable EV in this experiment to simulate. EV is a value from 0 to 1, when EV is close to 1, the network load is in a high situation, but when EV is close to 0, the load of network is in a low situation. In short, we can see EV as the ratio of all of data in the network and carrying capacity of network, so it can reflect the current network load conditions.

Because there are none of perfect network protocols based on dynamic congestion control algorithm, considering of fairness, not to choose the existing SCTP as a experimental subject and construct two experimental objects based on TCP protocol.

Object 1: SCTP simulation:

- One source endpoint can establish TCP connections with 10 destination endpoints;
- Each connection has its own parameters of congestion control;
- From initial status, the source endpoint only chooses one destination endpoint and sends data to it;
- The source endpoint always remains sending data to the same destination endpoint;
- The sending endpoint uses a slow start and congestion avoidance to regulate the congestion window;
- Using SACK to confirm;
- Once the source endpoint has congestion with the current destination endpoint. The sender has to choose another destination endpoint to send packet, and send congestion window at a fixed value.

Object 2: DSCTP simulation:

- One source endpoint can establish connections with 10 destination endpoints;
- Each connection has its own parameters of congestion control;
- The sending endpoint can sends data to any host at any time;
- The sending endpoint can process data and environmental information at the same time;
- The sending endpoint based on environmental information to predict the status of the network;
- According to the predicted value the sending endpoint choose to continue to send data to the current host, or choose to send data to other hosts;
- If you need a switch, the sending endpoint retransmit all data in the send window to a new destination endpoint, and notify the original host to clear receiver's buffers and use of the size of the original send window to set a new send window.

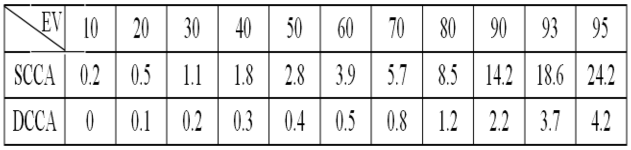
In fact, the essence of simulation is to contrast two experimental subjects. CN and ET correspond with the changes of EV.

B Test Results:

Assume that transmit 100000 packets, each size of packet is 1500 byte (the maximum size of Ethernet), the sum size is 150 MB. Using of two kinds of congestion control mechanisms. From the below we can see that the value of CN and ET (s). As shown in the below, we can divide EV into three parts: first EV is from 0 to 40, and then EV is from 40 to 90, the last EV is from 90 to 100. The value of EV is from 0 to 100, when the value of EV is equal to 0, network is in an idle status; when the value of EV is equal to 100, network is in a fully congestion status; we only test the value of EV from 10 to 95 because the value of EV is lower than 10, the network basically has no data traffic. But if the value of EV is higher than 95, the network is in a state of paralysis. The value of ET is the time interval from the first packet transmitted to the last packet, which expresses the efficiency of the transport network. The smaller the ET, the higher transmit efficiency is. The value of CN is the times of congestion which expresses the ability of controlling network congestion. The smaller the CN, the higher congestion capability is. SCCA expresses static congestion control algorithm and DCCA expresses dynamic congestion control algorithm.

Table I and Table II list the required EV of 10, 20, 30, 40, 50, 60, 70, 80, 90, 93, 95 respectively, under SCCA and DCCA. What Table I contains is CN. From Table I we can see that CN of SCCA is from 0.2 to 24.2, while CN of DCCA is from 0 to 4.2. Based on Table I, draw a two-dimensional graph of Fig. 3. What Table II contains is ET. From Table II we can see that ET of SCCA is from 41.1 to 2505.4, while CN of DCCA is from 50.7 to 2641.7. Based on Table II, draw a two-dimensional graph of Fig. 4.

TABLE I. COMPARE CN OF DCCA WITH SCCA



EV	10	20	30	40	50	60	70	80	90	93	95
SCCA	0.2	0.5	1.1	1.8	2.8	3.9	5.7	8.5	14.2	18.6	24.2
DCCA	0	0.1	0.2	0.3	0.4	0.5	0.8	1.2	2.2	3.7	4.2

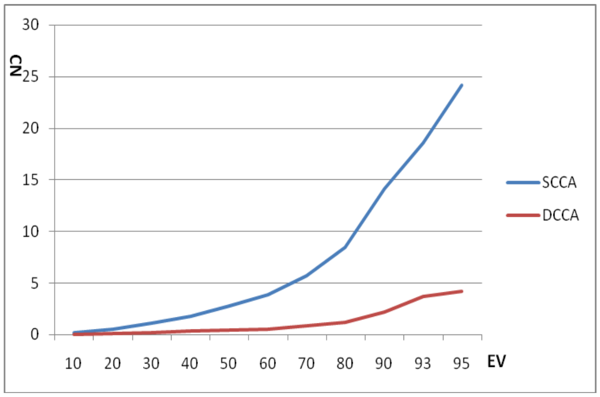


Figure 3. Compare CN of DCCA with SCCA

From Table I and Fig. 3 we can see that the value of CN in the SCCA and the value of CN in the DCCA are nearly equal to 0 when the value of EV is lower than 20. When the value of EV is higher than 95, the increase of CN will increase drastically eventually converge towards endless. From this we can see that DCCA is always better than SCCA.

Table I and Table II list the required EV of 10, 20, 30, 40, 50, 60, 70, 80, 90, 93, 95 respectively, under SCCA and DCCA. What the table contains is CN. From Table I we can see that CN of SCCA is from 0.2 to 24.2, while CN of DCCA is from 0 to 4.2. Based on Table I, draw a two-dimensional graph of Fig. 3.

TABLE II. COMPARE ET OF DCCA WITH SCCA

EV	10	20	30	40	50	60	70	80	90	93	95
SCCA	41.1	72.6	117.8	193.5	286.1	404.1	569.8	834.2	1457.8	1590.4	2505.4
DCCA	50.7	86.8	137.2	187.7	254.2	301.3	475.9	723	1384.9	1865.2	2641.7

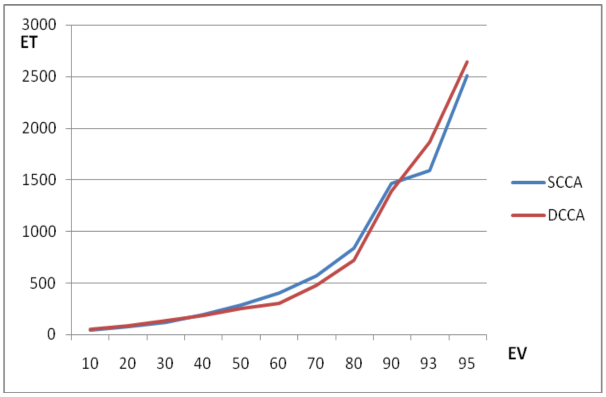


Figure 4. Compare ET of DCCA with SCCA

From Table II and Fig. 4 we can see that the values of ET in the DCCA and in the SCCA have two cross points, one is EV=40 and the other is EV less than 90. This is because DCCA use the dynamic congestion control mechanisms to reduce the probability of congestion occurs and improve the overall efficiency of transport network.

At the point of EV is equal to 40, two curves of ET gradually begin to coincide. This is because in a network with high load situation, no way can be taken to avoid network congestion occurs. The network was in a state of paralysis.

We can see that ET of DCCA is lower than ET of DCCA when the value of EV from 40 to 90. Because DCCA uses the dynamic congestion control mechanisms to reduce congestion occurs and also improves the network transmission.

When the value of EV is higher than 90, the increase of ET will increase drastically eventually converge towards endless. Because there is no way to avoid network congestion occurs when they are in a heavy-load conditions. From this we can see that DCCA is always better than SCCA.

C Result Analyzing

From the above results can draw the following conclusions, according to Table I, Table II, Fig. 3 and Fig. 4:

In the good network conditions, transport efficiency of dynamic congestion control policy is a little below the traditional congestion control policy. Considering the network was in an unoccupied state, so that the performance of internet would not cause too much influence. Because there is hardly any task in the transmit link in the light-load conditions, there is no significant difference between DCCA and SCCA.

In the normal-load conditions, using dynamic congestion control algorithm could reduce the probability of congestion occurs and improve the network transmission. In the heavy-load conditions, using dynamic congestion control algorithm and traditional congestion control algorithm could not change the network congestion situation.

Thus, the dynamic congestion control algorithm does help to reduce the network congestion occurs, and also help to enhance overall efficiency of the transport network.

VII. SUMMARY AND CONCLUSION

The main idea of this paper is applying the dynamic policy management framework to SCTP. Congestion control is a method used for monitoring the process of regulating the total amount of data entering the network so as to keep traffic levels at an acceptable value. The traditional SCTP initial start is so slow and SCTP use only one destination address at any given time to transmit new data. Therefore, the new algorithm design uses dynamic congestion control algorithm which describes the congestion control mechanism in detail in respect of transmission start, flow control, network monitoring, generation of policies, routing switching, and congestion recovery. Detection of network conditions is a core operation of dynamic adaptive network. The actions such as dynamically determining the value of the threshold and dynamically transferring tasks are practically based on the principle of prediction and feedback. However, merely using prediction and feedback is not enough to complete dynamic scheduling. What is more important is to integrate the dynamic policy management framework. Developed on the basis of conventional static policy management framework, this dynamic framework has the characteristics of self-government, which allows the whole algorithm to keep working without the involvement of the administrator. In this way, the policy scheduling can guarantee that it will keep up with the rhythm of change on the system, thereby, significantly enhancing the transferring efficiency.

ACKNOWLEDGMENT

The project of the research on hierarchy and protocol of dynamic control and management for packet service oriented Heterogeneous / Converged Network is supported by Ministry of Technology and Science of People's Republic of China under Grant No.2007AA01Z204 and Sino-Swedish collaboration research program under Grant No. 2008DFA11950.

REFERENCES

[1] Postel J., "User Datagram Protocol". RFC 768, IETF, Aug. 1980.

[2] http://www.rfc-editor.org/rfc/rfc3060.txt. [accessed: January 20, 2011]

[3] http://tools.ietf.org/html/draft-ietf-policy-arch-00. [accessed: January 20, 2011]

[4] http://www.rfc-editor.org/rfc/rfc3318.txt. [accessed: January 20, 2011]

[5] Dulay N., Lupu E., Sloman M., and Damianou N., "A Policy Deployment Model for the Ponder Language", An extended version

of paper in Proc. IEEE/IFIP International Symposium on Integrated Network Management(IM' 2001), Seattle, May 2001, IEEE Press.

[6] Liu X. J., Liu Y. H., Wei D., and Liu H. Y., "Dynamic Policy Based Network Management Scheme in Mobile Environment", 2008 International Symposium on Computer Science and Computational Technology.

[7] Iyengar J. R., Amer P. D. and Stewart R., "Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths", Proc. IEEE/ACM Transactions on Networking, Vol. 14(5), pp. 951-964, Oct 2006.

[8] Fracchia R., Casetti C., Chiasserini C. F. and Meo M., "WiSE: Best-Path Selection in Wireless Multihoming Environments" Proc. IEEE Transactions on Mobile Computing, Vol. 6(10), pp. 1130-1141, Oct. 2007.

[9] http://pel.cis.udel.edu/. [accessed: January 20, 2011]

[10] Postel J., "Transmission Control Protocol". RFC 793, IETF, Sept. 1981.

[11] http://www.rfc-editor.org/rfc/rfc4960.txt. [accessed: January 20, 2011]

[12] Allman M., Paxson V., and Stevens W., "TCP Congestion Control". RFC 2581, IETF, Apr. 1999.

[13] Humaira K., Brad P. and Alan W., "SCTP versus TCP for MPI", Proc. Thirteenth International Symposium on Temporal Representation and Reasoning, TIME 2006.

[14] Stewart R., Xie Q., Morneault K., Sharp C., Schwarzbauer H., Taylor T., Rytina I., Kalla M., Zhang L., and Paxson V., "Stream Control Transmission Protocol," RFC 2960, Oct. 2000.

[15] Stewart R., Arias-Rodriguez I., Poon K., Caro A., and Tuexen M.,"Stream Control Transmission Protocol specification errata and issues,"draft-ietf-tsvwg-sctpimpguide-16.txt, Apr. 2006.

[16] http://en.wikiversity.org/wiki/What_is_Congestion_control%3F. [accessed: January 20, 2011]

[17] Jacobson V.,"Congestion Avoidance and Control". Proc. Computer Communication Review, Vol. 25(1), pp. 157-173, Jan. 1995.

[18] Eager D. L., Lazowska E. D. , and Zahorjan J., "A Comparison of Receiver-Initiated and Sender-Initiated Adaptive Load Sharing", Performance Evaluation,Elsevier, Amsterdam, Holland, Vol. 6. pp. 53-68,1986.