SCTP-based Bandwidth Aggregation Across Heterogeneous Networks

Chunlun Huang[1], Zhiyue Zeng[1]
[1] *School of Computer and Information Science*
Southwest University, Chongqing, 400715, China
clhuang@swu.edu.cn, zhiyuezeng@tom.com

Abstract

Nowadays more and more various networks can be used to access the internet. But a lot of them, especially most wireless networks, only offer very limited bandwidth to the end users. In order to gain much wider bandwidth than one network can offer, we modified the Stream Control Transport Protocol (SCTP) so that it can aggregate bandwidth over multiple interfaces of a host to serve an application simultaneously. The main algorithms proposed by us include virtual association, compound congestion control window and mechanisms of transferring and retransferring packets. We also simulated our solution by the open source software Net Simulator version 2 (NS-2), the results show that the efficiency of our solution is over 80% in terms of bandwidth utilization. We still discussed the future studies that have to do before the modification can be used in real more complex topologies networks.

1. Introduction

As technologies of communication and computer continue their explosive growths, it's very clear that a host (PC or any other computing machine) can easily access the internet through multiple technologies such as WiMax, CDMA, SCDMA, PHS, GPRS, LAN, ISDN, optical network, 3rd generation mobile communication technology and so on. But the current TCP/IP protocols only permit a session to use one of these access methods at the same time, although there is more than one interface connecting to the internet. This limits us to gain wider bandwidth by using all available interfaces to service one application especially when we have several interfaces for example GPRS, CDMA and PHS, and each of them can't afford sufficient bandwidth for an application alone. When this case happened one obvious simple way is to take advantages of all the available interfaces simultaneously, which maybe belong to heterogeneous networks. We call this method bandwidth aggregation. Bandwidth aggregation is not only used for achieving wider bandwidth but also for the sake of reliability in some special cases such as military applications [1].

Because different communication technologies are not compatible at physical layer, therefore upper layer's protocols are the best places to gather wider bandwidth across heterogeneous networks. For example, Dhananjay S. Phatak proposed a solution at the network layer[2]. However, solution at network layer is too difficult to keep compatible with other original protocols. So many solutions are derived from transport layer. For example, the pTCP [3] and the R-MTP [4] are both based on the TCP protocol. Another solution based on Stream Control Transmission Protocol (SCTP) [1] is done at the transport layer too.

The Stream Control Transmission Protocol (SCTP) is a reliable new transport protocol that operates on the top of IP network. One of the most important new ideas that SCTP introduces is supporting multi-homing. 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 [1].

In this paper, we also proposed a solution to gain bandwidth aggregation by modifying the SCTP protocol slightly. Since the SCTP already supports multi-homing, we fix our attention on how to schedule all these interfaces to transmit data of an association at the same time. We defined the v-association and designed the v-association table to assist scheduling the available interfaces of a host. We simulated our solution by NS2 and the results are very inspiring.

2. Association of SCTP

Since SCTP is a connection-oriented protocol, before data can be transmitted between two endpoints a reliable communication relationship between them, which is called an association, must be set up first. In SCTP the endpoint is identified by SCTP transport address which is defined as a combination of one or more IP address and an

978-0-7695-3490-9/08 $25.00 © 2008 IEEE
DOI 10.1109/PACIIA.2008.53

SCTP port. For the multi-homing reason, one endpoint can include several IP addresses but only one SCTP port, therefore a possible endpoint may be defined like this: [5]

Endpoint A = [160.15.82.20 16.10.8.221:100]

An SCTP association can be conveniently denoted by a pair of two SCTP endpoints. An example of SCTP association is like this: [5]

Association S = {[160.15.82.20 16.10.8.221:100]: [128.33.6.12:200]}

During association establishing the endpoints must notify the other which IP addresses it would use during the lifetime of this association, but during application data transporting the endpoints just selects one address as the primary and the other as backup. All chunks are transported trough the primary address except when the chunks are to be retransmitted.

These characters of SCTP association offer some preparations for aggregating bandwidths from several interfaces of a host.

3. Proposed Algorithms

Although it is relevantly convenient to realize bandwidth aggregation in SCTP, there is still a long way to go for the complexities of the networks and the procedure of transmission. In our project we deliberately managed with the following problems and completed the simulation for our solutions.

3.1. V-association

Generally, the sending dataflow of SCTP is : Application→Fragmentation→Chunks Queue→Bounding →Sending to network layer [5]. The Sending part is the key to aggregate bandwidth. We referred to pTCP [2] and defined virtual association, v-association in brief. A v-association represents one possible link between the source and destination hosts. Before sending data, one v-association should be selected first. By selecting different v-association for different data packages alternately, bandwidth aggregation is acquired.

V-association has the following attributes: Identifier (ID), Source IP (S-IP), Destination IP (D-IP), Round Transfer Time (RTT), Round Transfer Time Out (RTO), Congestion Control Window (cwnd), Path Max Transmission Unite (PMTU). All these attributes are recorded in v-associations table, table 1. The most outstanding differences between association and v-association are that a v-association only has one source and one destination IP-addresses, and no SCTP ports, it shares the same source and destination SCTP ports with the association it belongs to.

Table 1: V-associations Table

ID	S-IP	D-IP	RTT	RTO	cwnd	PMTU

3.2. V-association's Setup

In our project, the association still keeps its original definition and its setup procedure almost un-impacted. We just insert a v-association setup process between association setup and beginning to transmit data to get ready for scheduling v-associations. We suppose that the sender has m (≥1) interfaces (m S-IPs) and the receiver has n (≥1) interfaces (n D-IPs). So there will be m × n possible v-associations between the sender and receiver. But the useful v-associations are no more than max (m, n). So how to select the max (m, n) v-associations of m × n v-associations is a key problem. We define the criterion is that the subclass should include all S-IPs and D-IPs and their RTTs should be less than the left v-associations' except when one's source and destination IPs are both included by those v-associations which RTT is less than it's. Therefore we designed the following v-association setup algorithm in which we used the HEARTBEAT chunks (more details please refer reference 5) to calculate the RTT for a v-association. Our algorithm is:

①. By sending HEARTBEAT chunks and receiving HEARTBEAT-ACK chunks on m × n possible v-associations we can get m × n possible RTTs recorded as $RTT_{1,1}$, $RTT_{1,2}$…$RTT_{m,n}$. If a v-association is not reachable, its RTT would be recorded as -1(∞).

②. Sort v-associations according to $RTT_{1,1}$, $RTT_{1,2}$ … $RTT_{m,n}$ by ascending order. And label the sorted RTTs as RTT_1, RTT_2…$RTT_{m\times n}$.

③. Select the first L (= max(m, n)) v-associations as proposals. If some (say s) proposals' RTTs equate -1 or their ratio to RTT_1 are greater than a threshold θ (for example θ = 4) then L = L – s.

④. If proposed v-associations include all S-IPs and D-IPs then go to ⑥.

⑤. Select an S-IP or a D-IP missed by the proposed v-associations. If there are some v-associations in the left subclass which contain the same S-IP or D-IP and their RTTs are less than the product of θ and RTT_1, take the one whose RTT is lest to replace the one whose S-IP and D-IP are both repeated by other proposed v-associations and whose RTT is longer than other proposed v-associations which repeat the same S-IP and D-IP. Repeat ⑤ until all missed IPs are dealt with.

⑥. Resort the proposed v-associations by ascending order and fill them in the v-associations table in order.

⑦. Count the PMTU, RTO and cwnd of each proposed v-association. Their calculating methods are consistent with SCTP standard.

This algorithm has effect on bundling data with COOKIE-ECHO and COOKIE-ACK chunks. In our solution we disabled this function.

3.3. Compound Congestion Window

The congestion control function is a crucial part of any transport protocol. After adding bandwidth aggregation function, there are two ways about how to deal with it. One way is to take a uniform control on an association. The other is to take a separate control on every link (v-association) of an association. The first way may be inefficient and complicated when there are several v-associations and some v-associations are through different networks. And the second method has difficulties in keeping the continuity of the TSN of chunks assigned to every v-association and keeping every windows sliding according to the uniformed TSN sequence. So we designed a compound congestion window as table 2 which contains all sent chunks' TSN, the v-association ID every chunk is sent to and the current status of each chunk. For each chunk in congestion control window we defined three statuses: waiting for acknowledge (Wait), acknowledged (Ack), failed and waiting for retransmission (Fail).

Table 2: Compound Congestion Window

TSN	…	1004	1003	1002	1001	1000
ID	…	1	2	2	1	1
status	…	Fail	Ack	Wait	Wait	Wait

Then our congestion control algorithm is:

①. For each v-association the basic parameters (cwndi, ssthresh, flightsize and partial bytes acknowledge, see [5] for more information) of congestion control have the same meanings and algorithms with as described in SCTP specification. Thus the compound congestion window's size cwnd is equal to the sum of all v-associations' $cwnd_i$. That is:

$cwnd = cwnd_1 + cwnd_2 + \ldots + cwnd_L$

②. One chunk can be sent to the v-association whose ID is i only when the number of i in compound congestion window is less than $cwnd_i$. As soon as a chunk is sent to v-association i, a new record made of TSN, i and wait status is added to compound congestion window's tail.

③. The compound congestion window's sliding rule is the same as SCTP's sliding window. And only when the compound congestion window slides can the buffers of the acknowledged chunks be released. The compound congestion window's sliding controls the shifts of all windows of v-associations.

④. The receiver cumulates duplicate TSN and Gap Ack block according to the status of compound congestion window not the buffer status of v-association.

3.4. Data Transmission and Retransmission

The most important two goals of data transmission algorithm are to gain the biggest bandwidth and the shortest delay time. But these two goals are mutually exclusive sometimes. According to reference [2] the RTT of link reflects the effects of bandwidth and time delay synthetically because of

$$RTT = p / b + \varepsilon$$

Where p is packet's length, b is bandwidth and ε is time delay. Thus we decided to schedule v-associations in accordance with the "Lest RTT First" rule. Hence we got the sending data algorithm as following:

①. Set v-association ID k = 1.

②. If the buffer of v-association k is not full then
    send a chunk to v-association k
  else
    k = k+1
    if k <= L
      go to ②
    else
      return fail

③. If there are still chunks waiting to transmit then
    go to ②

④. return success.

The retransmission algorithm is different with that depicted in SCTP specification in how to select the retransmission link too. In our solution we still stick to the "Lest RTT First" rule but excluding the v-association to which the error chunks were sent before.

This transmission and retransmission algorithm may induce a dead lock. Think that the chunk waiting for retransmission is the first chunk in compound congestion window and when begin to retransmit it all the other v-associations' buffers are full. Although there may be some chunks which has been acknowledged their buffers couldn't be released because of the head congestion. Thus the compound congestion window can't slide and no chunks could be transmitted any longer include the first chunk waiting for retransmission. When this case happens we revoke one chunk or more in the v-association which is scheduled to retransmit the first chunk.

When the number of continuously retransmitting on a certain v-association is more than a threshold we consider that v-association is unavailable and then set its RTT as -1 which means we will no longer send any chunks to it.

4. Simulation and Results

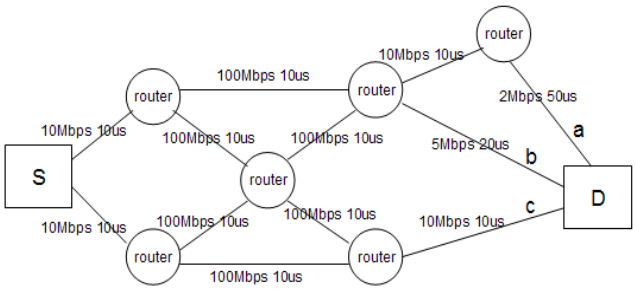

Figure 1: Network Topology

Table 3: Utilization of Bandwidth

Bandwidth Mbps			Utilization of Bandwidth		Utilization of Each Link		
a	b	c	Mbps	Rate	a	b	c
2	2	5	7.19	79.9%	7.2%	26.1%	66.8%
2	5	5	10.42	86.9%	8.8%	45.5%	45.7%
5	5	5	13.38	89.2%	29.1%	35.4%	35.5%
2	5	10	13.82	81.3%	0.0%	30.4%	69.6%
5	5	10	17.16	85.8%	16.9%	27.6%	55.5%

We used the ns-allinone-2.30 simulator[6] which includes the SCTP module. The generic topology for our simulation is shown in figure 1 where the S presents the source host and the D presents the destination host. Each link's bandwidth and delay are labeled in the figure. Here we presented representative results concerning two different scenes: in one scene the delay is invariable and the bandwidth is variable and in the other scene the bandwidth is invariable and the delay is variable. We changed the bandwidth or delay of link a, b or c to change the bandwidth or delay of v-association a, b or c for in our simulation the three links are all included almost every time. Packet loss rates from 0.001% to 0.1% are used in the simulations to simulate the retransmission mechanism.

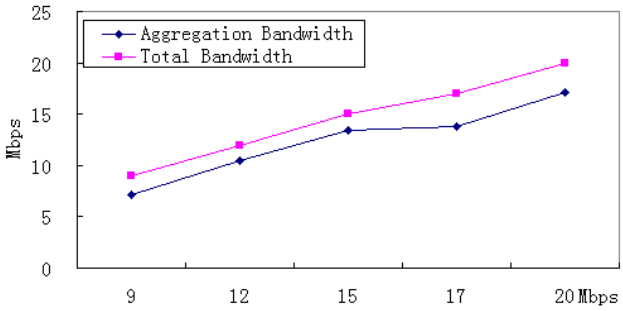

Figure 2: Aggregation Bandwidth and Total Bandwidth

In first scene we keep all the links' delay invariable as labeled in figure one and vary the bandwidth of link a, b and c in 2, 5 or 10 Mbps. The simulation results are presented in table three and figure two. From table 3 we can deduce the following conclusions: ① the average rate of bandwidth aggregation is about 84.6% which exemplified that we can get more bandwidth by aggregating heterogeneous links together, the compare of total Bandwidth and aggregation bandwidth is shown in Figure 2; ② in an association, the larger the bandwidth is, the more data the link transmits; ③ if the ratio of a v-association's bandwidth to the largest one's is less than a quarter the slow v-association will not be used; ④ the more similar the bandwidths of every v-associations were, the higher the rate of bandwidth aggregation was gotten. There are two main reasons for the rate of bandwidth aggregation is less than 1.The first is that it took a long time to set up the v-association table. This reason is very disadvantageous when to transmit a small number of data. But its effect can be omitted when to transmit a large enough number of data. The second reason is that when the fast link is busy the scheduling algorithm still tests it first and then tests other link. This effect can be reduced greatly by improving the scheduling algorithm. Another possible reason is that the little different RTT may have some effect on the efficiency.

Table 4: Effect of different RTT

Delay Time 10 us			Bandwidth Utilization		Utilization of Each Link		
a	b	c	Mbps	Rate	a	b	c
2	48	63	15.69	92.3%	11.7%	29.5%	58.9%
2	8	23	15.63	91.9%	11.4%	29.5%	59.1%
2	48	23	14.58	85.8%	12.6%	31.7%	55.7%
2	8	3	13.82	81.3%	0.0%	31.8%	68.2%
2	48	143	14.80	87.1%	12.5%	31.2%	56.3%
2	8	143	14.72	86.6%	11.8%	31.5%	56.7%
62	3	53	13.39	78.8%	0.0%	35.7%	64.3%
2	128	143	14.39	84.7%	13.4%	29.3%	57.3%
2	48	303	6.48	38.1%	28.7%	71.3%	0.0%
2	128	303	6.40	37.6%	29.1%	70.9%	0.0%

In scene two, we fixed the PMTU as 1500 bit, the bandwidth of link a, b and c as 2Mbps, 5Mbps and 10Mbps. And then we changed the delay time as shown in table 4 so that the RTT of each v-association abode the rules which a:b:c is 1:1:1, 1:1:2 etc. From this table we can conclude that when the RTTs of every v-associations are approximately equal the aggregation is most efficient, contrarily the more different the RTTs, the less efficient the aggregation is. When the difference of RTT is inevitable the long delay on the slow link has less effect than that on the fast link. Worse of all, if the long delay on the fast link is longer than the threshold the aggregation efficiency will lower than 50% which means the aggregation will be unacceptable. In our solution we didn't consider that the faster link had the longer delay which may occur in wireless networks though. To deal with this case better we suggest adding a bandwidth weighting coefficient so that the faster link with longer delay can be advanced in v-association table.

5. Conclusion and Future Work

In this paper we modified the current SCTP protocol so that it supports bandwidth aggregation across heterogeneous networks. Our simulation results show that the modified protocol keeps most compatible with the SCTP specification and can seamlessly schedule all available interfaces of a host.

Well, there are still many further studies to do before we can put it in use. The further studies include how to deal with shared bottle neck, how to add a new v-association to and subtract a failed v-association from the active association dynamically and how to realize seamlessly shift across different networks etc.

Acknowledgement

It is a project supported by the youth foundation of the school of computer and information science of the Southwest University of China.

References

[1] Antonios Argyriou, Vijay Madisetti. "Bandwidth Aggregation with SCTP". *IEEE GLOBECOM 2003* pp. 3716-3721

[2] Dhananjay S. Phatak, Tom Goff. "A Novel Mechanism for Data Streaming Across Multiple Ip Links for Improving Throughput and Reliability in Mobile Environments". *IEEE INFOCOM 2002*. pp.773-781

[3] Hung-Yun Hsieh, Raghupathy Sivakumar. "A Transport Layer Approach for Achieving Aggregate Bandwidths on Multi-home Mobile Hosts". *MOBICOM'02, September 23-28, 2002, Atlanta, Georgia, USA*. pp. 83-94

[4] Luiz Magalhaes, Robin Kravets. "Transport Level Mechanisms for Bandwidth Aggregation on Mobile Hosts". *IEEE GLOBECOM 2001*. pp.165-171

[5] Randall R. Stewart, Xie qiaobing. *Stream Control Transmission Protocol (SCTP) A Reference Guid*. China Tsinghua Publishing House, Beijing, 2003.

[6] S. McCanne, S. Floyd. "Ns Network Simulator". http://www.isi.edu/nsnam/ns

[7] Hung-Yun Hsieh, Kyu-han KIM, "Raghupathy Sivakumar. An End-to-End Approach for Transparent Mobility across Heterogeneous Wireless Networks". *Mobile Networks and Applications* 9, 2004. pp.363-378