Modeling and Simulation of Traffic Aggregation Based SIP over MPLS Network Architecture

Bo Rong, Jacques Lebeau, Maria Bennani, Michel Kadoch
LAGRIT, Department of Electrical Engineering
Ecole de technologie superieure, Universite du Quebec
Montreal, Quebec, Canada, H3C 1K3

Ahmed K. Elhakeem
Department of Electrical and Computer Engineering
Concordia University
Montreal, Quebec, Canada, H3G 1M8

Abstract—**In this paper, a traffic aggregation based SIP over MPLS network architecture is proposed to integrate SIP protocol with traffic engineering (TE) enabled MPLS network seamlessly and speed up the SIP call setup. We rely on OPNET simulation to assess the performance of this new network architecture. As a highlighted part, basic design concepts and implementation details of the simulation model are depicted intensively. The simulation results show that traffic aggregation based SIP over MPLS network architecture has satisfying performance in terms of SIP call setup delay, call blocking probability, and bandwidth utility efficiency.**

I. INTRODUCTION

There are two highly concerned issues on using IP network to carry audio/video traffic. The first one is the signaling protocol to control a call; the second one is how IP network provides real-time audio/video applications with QoS guaranteed service. As for the first issue, Session Initiation Protocol (SIP) has recently gained the increasing interest from universities, standardization organizations and companies [1~8]. As for the second issue, Multi-protocol Label Switching (MPLS) is suggested as an apt switching technology for future core networks [9~13]. Despite the fact that MPLS development started with goal to expedite packet forwarding, the main benefit from MPLS in current network environment is from its traffic engineering (TE) capabilities including QoS guarantee, efficient usage of network resources, resilience, and economy [14~16]. In fact, the traffic engineering capabilities make MPLS network a good choice for carrying SIP signaled audio/video traffics.

Although the architecture of SIP over MPLS has been discussed in [17,18], according to our study, directly using SIP protocol over traffic engineering enabled MPLS network as previous literature suggested can cause an unbearable long SIP call setup delay. In traffic engineering enabled MPLS network, when a new SIP call request comes, usually Constraint-based Routing Label Distribution Protocol (CR-LDP) or Resource Reservation Protocol with Traffic Engineering Extensions (RSVP-TE) will be employed to set up a corresponding Label Switched Path (LSP) dynamically. This kind of dynamic LSP setup can contribute a lot to the whole SIP call setup delay in traffic engineering enabled MPLS network. If we use Label Distribution Protocol (LDP) instead of CR-LDP or RSVP-TE in MPLS network, the SIP call setup delay is short and acceptable. Nevertheless, LDP cannot support traffic engineering function in MPLS network, because it is only suitable for hop-by-hop label distribution and always selects the same physical path as conventional IP routing would select.

To solve the long SIP call setup delay problem discussed above, we propose a traffic aggregation based SIP over MPLS network architecture in this paper. As a crucial component of this new architecture, the SIP-MPLS traffic aggregation server (TA server) is emphasized. The TA servers exchange traffic engineering signaling with MPLS network on behalf of a cluster of SIP clients. As a result, the SIP call setup delay in MPLS network is decreased substantially. Most importantly, all of our theoretical analysis and technical conclusions are supported by the OPNET simulation results, which are achieved on OPNET Modeler 10.0 with SIP and MPLS models.

The rest of the paper is organized as follows. Firstly, we introduce the traffic aggregation based SIP over MPLS network architecture in Section 2. Then, the call admission and bandwidth re-negotiation algorithms running on TA servers are suggested in Section 3. In Section 4, the OPNET simulation results of different call admission and bandwidth re-negotiation algorithms are presented. Furthermore, Section 5 depicts the OPNET simulation study of SIP call setup delay in distinct network architectures. In the end, Section 6 summarizes the results.

II. TRAFFIC AGGREGATION BASED SIP OVER MPLS NETWORK ARCHITECTURE

In order to shorten the long SIP call setup delay caused by CR-LDP or RSVP-TE, a traffic aggregation based SIP over MPLS network architecture is proposed in this section. Fig.1 shows that this network architecture contains 2 parts, the local client network and the MPLS core network. Furthermore, there are two components in the client network, the SIP terminal and the TA server. The TA server can be regarded as an enhanced SIP proxy server, and it negotiates/re-negotiates with the LER of MPLS core network about the overall QoS requirements of local client network on behalf of a cluster of SIP terminals (not only one SIP terminal). Then, the LER exchanges TE signaling with other routers inside the MPLS core network to set up the corresponding LSPs. We use a set of time marks $\{t_0, t_1, t_2, \ldots, t_{n-1}, t_n, t_{n+1}, \ldots\}$ to describe the time in system. If the TA server knows exactly that the overall bandwidth requirement of its local client network during $[t_{n-1}, t_n]$ is $B_{n-1,n}$, then at time t_{n-1},

Proceedings of the 38th Annual Simulation Symposium (ANSS'05)
1080-241X/05 $20.00 © 2005 IEEE

the TA server negotiates with MPLS network to get $B_{n-1,\ n}$ outgoing bandwidth by using Common Open Policy Service (COPS) messages. As time goes by, if the TA server knows that the overall bandwidth requirement of its local client network during $[t_n,t_{n+1}]$ changes to $B_{n,\ n+1}$, then at time t_n, the TA server would re-negotiate with MPLS network to increase/decrease the bandwidth requirement to $B_{n,\ n+1}$. By this way, the LSPs needed by SIP telephony are setup in MPLS core network before SIP calls are made. As a result, the SIP call setup delay in MPLS network is decreased greatly. However, it is impossible for the TA server to know the exact value of $B_{n-1,\ n}$ before the time of t_{n-1}. Usually, the TA server can only employ a certain bandwidth prediction algorithm to give an approximate value of $B_{n-1,\ n}$, which can be defined as $B^{pred}_{n-1,\ n}$. If $B^{pred}_{n-1,n}<B_{n-1,n}$ during $[t_{n-1},t_n]$, the local client network doesn't have enough outgoing bandwidth to accommodate all the SIP calls, and the TA server has to utilize admission control algorithm to decline some of the call requests. On the contrary, if $B^{pred}_{n-1,n}>B_{n-1,n}$ during $[t_{n-1},t_n]$, some outgoing bandwidth resource of the local client network would be wasted. From the above discussion, we can conclude that the call admission and bandwidth prediction/re-negotiation algorithm running on TA servers is very important to the overall performance of traffic aggregation based SIP over MPLS network architecture.

Fig.2 shows the signaling flow in this new network architecture. The call set-up starts with a standard SIP INVITE message sent by the caller to the local TA server. The message carries the callee URL in the SIP header and the session specification (session specification describes the QoS requirements of the SIP call) in the body SDP. The caller treats the TA server as a standard SIP proxy server. Regarding the caller ID, the session information and the remaining outgoing bandwidth in local client network, the TA server decides whether this SIP call request is admitted. If the call request is admitted, the TA server would forward the original INVITE message to the callee; Otherwise, it simply sends the caller a DECLINE message to drop the call. Furthermore, no matter the call is admitted or not, it would be registered in the TA server for the purposes of call admission and bandwidth prediction in the future. In addition, if the bandwidth re-negotiation is needed, the TA server re-negotiates with MPLS core network to adjust the bandwidth requirement by using COPS messages.

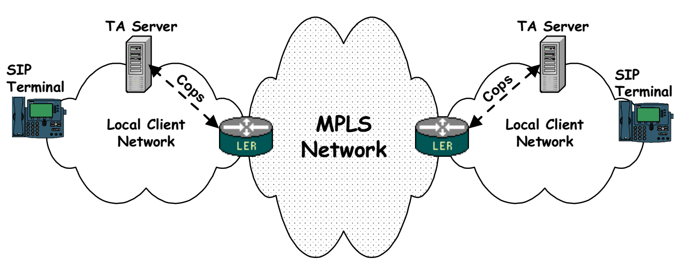


Fig. 1. Traffic Aggregation Based SIP over MPLS Network Architecture

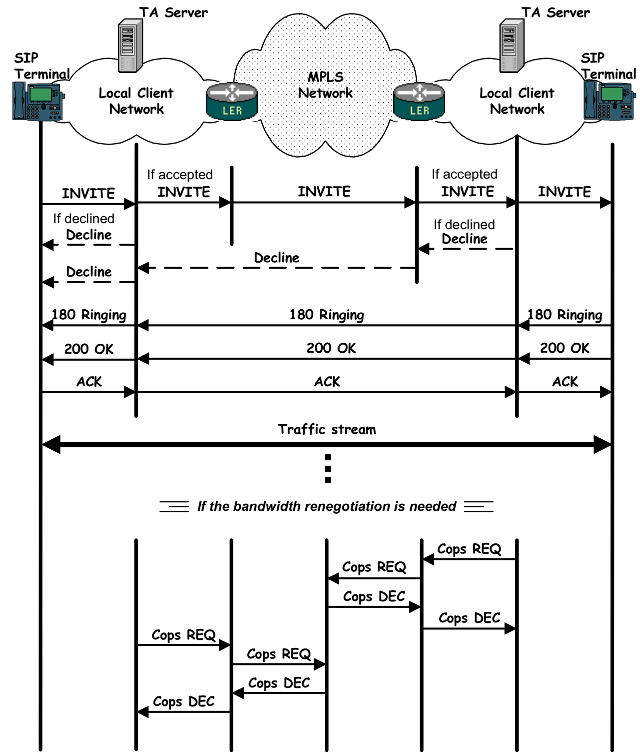


Fig. 2. The Signaling Flow in Traffic Aggregation Based SIP over MPLS Network Architecture

III. The Call Admission and Bandwidth Re-negotiation Algorithm on TA Servers

In this section, the call admission and bandwidth re-negotiation algorithms on TA server are discussed. Three algorithms, including simple and complicated ones, are depicted as follows.

(1) *FCFS with fixed bandwidth contract (FCFS-FB)*

In this case, the local client network has a fixed bandwidth contract with the MPLS core network, and the TA server accepts/declines SIP calls using *FCFS* policy. This algorithm has obvious drawbacks such as high call blocking probability or low bandwidth utility efficiency.

(2) *FCFS with static adaptive bandwidth contract (FCFS-SAB)*

We divide one day into a couple of periods, and in each period the local client network has a specific fixed bandwidth contract with the MPLS core network. Same to *FCFS-FB*, *FCFS* is employed for call admission in this algorithm. If the fixed bandwidth contract of each period can be set up reasonably according to the statistics of everyday traffic load in the local client network, this algorithm has better performance than *FCFS-FB*. However, the traffic load of local client network often behaviors unexpectedly. In this respect, this algorithm is not flexible enough.

(3) *Threshold based call admission and bandwidth prediction (TBCA-BP)*

Generally speaking, there are two parts in this algorithm, i.e., SIP call admission control and bandwidth prediction/re-negotiation. We integrate these two parts seamlessly in this algorithm, and the details are shown as follows.

Parameter Definitions in *TBCA-BP*:

Variables:

b_{req}: The bandwidth requirement of an incoming SIP call.

$b_{contract}$: The overall outgoing bandwidth that the TA server contracts with the MPLS network.

$b_{occupied}$: The outgoing bandwidth occupied by alive SIP calls.

$b_{acc\text{-}thr}$: The bandwidth threshold to accept all the incoming SIP calls.

$b_{dec\text{-}thr}$: The bandwidth threshold to make a lower bandwidth requirement prediction.

inc: The counter used to make a higher bandwidth requirement prediction.

dec: The counter used to make a lower bandwidth requirement prediction.

$p_{decline}$: The probability to decline the SIP call, when $b_{acc\text{-}thr} < b_{occupied} + b_{req} < b_{contract}$.

Constants:

$N_{dec\text{-}thr}$: The counter threshold to make a lower bandwidth requirement prediction.

$N_{inc\text{-}thr}$: The counter threshold to make a higher bandwidth requirement prediction.

$B_{contract\text{-}step}$: The bandwidth increment/decrement step for $B_{contract}$, if the bandwidth re-negotiation is conducted.

$B_{acc\text{-}thr\text{-}step}$: The bandwidth increment/decrement step for $B_{acc\text{-}thr}$, if the bandwidth re-negotiation is conducted.

$B_{dec\text{-}thr\text{-}step}$: The bandwidth increment/decrement step for $B_{dec\text{-}thr}$, if the bandwidth re-negotiation is conducted.

Relationships:

$b_{contract} > b_{acc\text{-}thr} > b_{dec\text{-}thr}$; $B_{dec\text{-}thr\text{-}step}, B_{acc\text{-}thr\text{-}step} < B_{contract\text{-}step}$

The Algorithm of *TBCA-BP*:

```
For each SIP call arrival of bandwidth requirement b_req
 if b_occupied + b_req > b_contract
   decline the call
   inc= inc +1
   dec= 0
   if inc >N_inc-thr
     b_contract= b_contract+B_contract-step /*make bandwidth prediction*/
     re-negotiate with MPLS network for the new b_contract
     b_acc-thr= b_acc-thr + B_acc-thr-step
     b_dec-thr= b_dec-thr + B_dec-thr-step
     inc=0
 else if b_occupied + b_req <b_acc-thr
   b_occupied = b_occupied + b_req
   accept the call
   inc =0
   if b_occupied <b_dec-thr
     dec= dec +1
     if dec >N_dec-thr
       b_contract= b_contract -B_contract-step /*make bandwidth prediction*/
       re-negotiate with MPLS network for the new b_contract
       b_acc-thr= b_acc-thr - B_acc-thr-step
       b_dec-thr= b_dec-thr - B_dec-thr-step
       dec=0
 else if b_acc-thr < b_occupied + b_req < b_contract
     dec= 0
     p_decline=( b_occupied + b_req- b_acc-thr)/(b_contract - b_acc-thr)
     With probability p_decline, decline the SIP call
     if the call are really declined
       inc= inc +1
       if inc >N_inc-thr
         b_contract= b_contract+B_contract-step /*make bandwidth prediction*/
         re-negotiate with MPLS network for the new b_contract
         b_acc-thr= b_acc-thr + B_acc-thr-step
         b_dec-thr= b_dec-thr + B_dec-thr-step
         inc=0
     else
       b_occupied = b_occupied + b_req
       accept the call
       inc=0
```

From the above description we know, the TA server makes call admission decision abiding by the following rules.

(1)*The rule of accepting a SIP call*: If $b_{occupied} + b_{req} < b_{acc\text{-}thr}$, the local client network is idle and has enough outgoing free bandwidth to accept new calls. As a result, the incoming calls would be always accepted. On the other hand, if $b_{acc\text{-}thr} < b_{occupied} + b_{req} < b_{contract}$, the local client network only has limited free outgoing bandwidth to accept new calls. So, the incoming call is accepted with the probability of 1- $p_{decline}$.

(2)*The rule of declining a SIP call*: if $b_{occupied} + b_{req} > b_{contract}$, the local client network has no free bandwidth to accept a new call, and all the incoming calls have to be declined. Moreover, if $b_{acc\text{-}thr} < b_{occupied} + b_{req} < b_{contract}$, the incoming call will be declined with the probability of $p_{decline}$.

To make bandwidth prediction/re-negotiation, the TA server complies with the following rules:

(1)*Asking for more bandwidth*: If more than $N_{inc\text{-}thr}$ SIP calls are declined consecutively, the TA server will make a prediction of more bandwidth requirement, and conduct bandwidth re-negotiation with MPLS network.

(2)*Asking for less bandwidth*: Upon accepting a SIP call, if the TA server finds the occupied bandwidth $b_{occupied}$ has been below $b_{dec\text{-}thr}$ for $N_{dec\text{-}thr}$ times consecutively, it will make a prediction of less bandwidth requirement, and conduct bandwidth re-negotiation with MPLS network.

From the above discussion we know that the algorithm of *TBCA-BP* is designed to have more flexibility, higher bandwidth utility efficiency, and lower call blocking probability than other two algorithms. Firstly, when $b_{acc\text{-}thr} < b_{occupied} + b_{req} < b_{contract}$, this algorithm drops calls randomly to avoid the global synchronization of many SIP terminals giving up their call requests at the same time. Secondly, this algorithm can trace the real bandwidth requirement in the local client network, and make an appropriate prediction. In conclusion, *TBCA-BP* is supposed to be a suitable call admission and bandwidth re-negotiation algorithm for TA servers.

IV. OPNET Simulation Study of Different Call Admission and Bandwidth Re-negotiation Algorithms

In this section, the simulation results are presented to demonstrate the performance of the algorithms suggested in Section 3. Although analytical approach can be applied to study these algorithms, it quickly becomes too complicated and intractable when the source model and the algorithm become complex. For this reason, we use OPNET simulation to study their performances. We suppose there are 4 groups of SIP call requests in the local client network, and in each group there are different bandwidth requirement classes.

(a)group G_1: 56Kbps (class1);
(b)group G_2: 200Kbps(class1), 500Kbps(class2);
(c)group G_3: 1.5Mbps(class1);
(d)group G_4: 4Mbps(class1), 6Mbps(class2).

Obviously, G_1 and G_2 are groups of low-bandwidth call requests, while G_3 is the medium-bandwidth call group and

G_4 is high-bandwidth call group. Moreover, call requests of each class in every group are assumed to arrive according to Poisson process independently, and the service times are exponentially distributed. The parameters used in this model are defined below:

$\lambda_{k,m}$(calls/hour)**:** average arrival rate of class *m* requests in G_k.
$1/\mu_{k,m}$(hours/call)**:** average service time for class *m* requests in G_k.
Pb_k(%)**:** the blocking probability of group G_k.
Eff(%)**:** the bandwidth utility efficiency of a certain algorithm.
$B_{contract}$(Mbps)**:** the overall outgoing bandwidth that the TA server contracts with the MPLS network.

The node model used in the simulation is shown in Fig.3. This node model consists of several sources (*G1*, *G2A*, *G2B*, *G3*, *G4A*, and *G4B*), a *call admission and bandwidth re-negotiation* module, and a *sink*. The basic function of this node model is to mimic the real process of handling a SIP call request, including call admission control, resource allocation, resource reclaim, and bandwidth re-negotiation. The sources in Fig.3 are designated to generate call request packets according to programmed schedule and in a special packet format. As shown in Fig.4, the packet format contains 4 fields (*groupID*, *classID*, *bandwidth requirement*, and *call duration*). All the call request packets are sent to the *call admission and bandwidth re-negotiation* module. After receiving a packet, the *call admission and bandwidth re-negotiation* module extracts the call request information and executes the *call admission and bandwidth re-negotiation* algorithm. If the call request is declined, the packet of this call request should be sent to the *sink* and deleted. On the contrary, if the call request is accepted, the *call admission and bandwidth re-negotiation* module has to reserve corresponding bandwidth resource for it until this SIP call is out of service time and deleted. In addition, when bandwidth re-negotiation is needed, the *call admission and bandwidth re-negotiation* module must increase/decrease the contracted bandwidth accordingly.

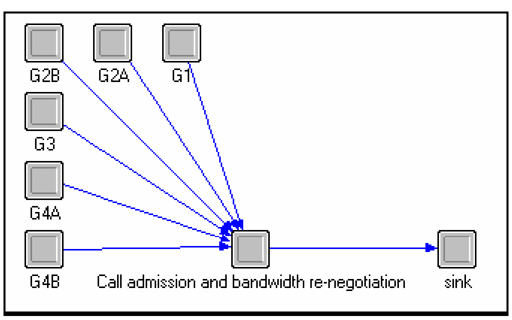


Fig. 3. Node Model Used in the Simulation

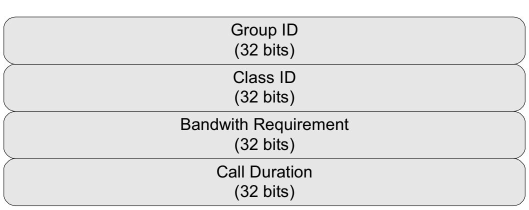


Fig. 4. Packet Format Used in the Simulation

With OPNET simulation, numerical results are attained from Fig.5 to Fig.9 to illustrate the performance of these three algorithms. Parameter setup of the simulation is shown as follows:

(1)*Traffic load in the local client network (a typical scenario from Bell Canada):*

Time of day (hour)	$\lambda_{1,1}$ (Calls/hour)	$\lambda_{2,1}$ (Calls/hour)	$\lambda_{2,2}$ (Calls/hour)	$\lambda_{3,1}$ (Calls/hour)	$\lambda_{4,1}$ (Calls/hour)	$\lambda_{4,2}$ (Calls/hour)
0:00 ~ 3:00	1440	360	600	72	19	14
3:00 ~ 5:00	452	105	300	46	13	7
5:00 ~ 8:00	240	85	87.5	12	5	2
8:00 ~ 11:00	1655	360	645	52	11	19
11:00 ~ 13:00	4610	490	1575	168	27	34
13:00 ~ 17:00	5280	1320	2200	191	70	71
17:00 ~ 21:00	5269	560	1400	245	51	38
21:00 ~ 24:00	2400	350	1125	153	32	16
$\frac{1}{\mu_{k,m}}$ =1 hour/call (any k or m)						

(2) *The parameter setup of these 3 algorithms*

(a) FCFS-FB: Fixed Bandwidth=250Mbps;

(b) FCFS-SAB:

Time of day (hour)	Bandwidth Contract (Mbps)
0:00 ~ 3:00	100
3:00 ~ 5:00	50
5:00 ~ 8:00	50
8:00 ~ 11:00	100
11:00 ~ 13:00	250
13:00 ~ 17:00	450
17:00 ~ 21:00	300
21:00 ~ 24:00	150

(c) TBCA-BP:

Variables:	
$b_{contract}$	Initial value=150Mbps, when time=0;
$b_{acc\text{-}thr}$	70% of $b_{contract}$
$b_{dec\text{-}thr}$	50% of $b_{contract}$

Constants:	
$B_{contract\text{-}step}$	20% of $b_{contract}$
$B_{acc\text{-}thr\text{-}step}$	70% of $b_{contract\text{-}step}$
$B_{dec\text{-}thr\text{-}step}$	50% of $b_{contract\text{-}step}$
$N_{inc\text{-}thr}$	100
$N_{dec\text{-}thr}$	1400

The simulation results are shown from Fig.5 and Fig.9 to demonstrate the bandwidth contract changing, blocking probability, and bandwidth utility efficiency of these 3 algorithms during one day. Firstly, we discuss the performance of *FCFS-FB*. As shown in Fig.5, *FCFS-FB* has no capability of bandwidth prediction. As a result, Fig.6 shows that the blocking probability of *FCFS-FB* is not stable. When the traffic load in local client network overweighs the outgoing bandwidth contract, the blocking probability is high. On the other hand, when the traffic load is lower than the outgoing bandwidth contract, the blocking probability is low, but a lot of bandwidth resource is wasted. Furthermore, from Fig.9 we know the bandwidth utility efficiency of *FCFS-FB* is varying with the traffic load in local client network uncontrolledly. In a word, the performance of *FCFS-FB* is terrible. Secondly, the performance of *FCFS-SAB* is investigated. Fig.5, Fig.7, and Fig.9 demonstrate that *FCFS-SAB* has a good bandwidth utility efficiency but poor blocking probability when its bandwidth contract is lower than the real traffic load in most time of 1 day. Actually, it is very difficult for *FCFS-SAB* to have both good bandwidth utility efficiency and good blocking probability due to its inflexibility. Thirdly, we focus on the performance of *TBCA-BP*. Fig.5 shows *TBCA-BP* has high bandwidth prediction accuracy, while Fig.8 and Fig.9 illustrate it also possesses satisfying blocking probability and bandwidth utility efficiency. Obviously, *TBCA-BP* has the best performance among these three algorithms, and it is an apt candidate to fulfill call admission and bandwidth re-negotiation task on TA servers.

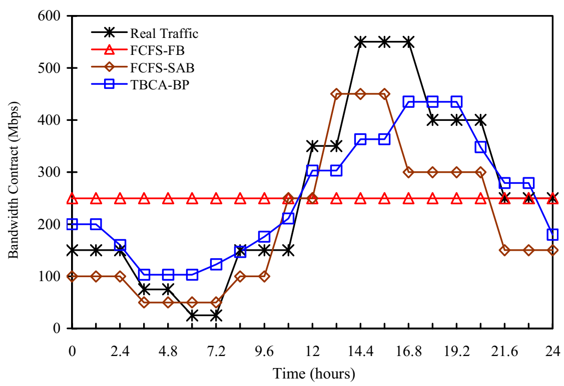


Fig. 5. The Bandwidth Contract Changing of All 3 Algorithms during 1 Day

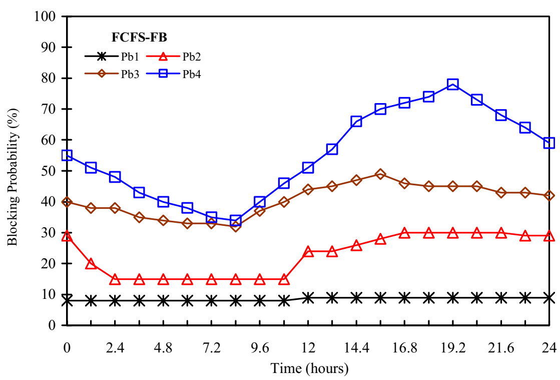


Fig. 6. The Blocking Probability of *FCFS-FB* during 1 Day

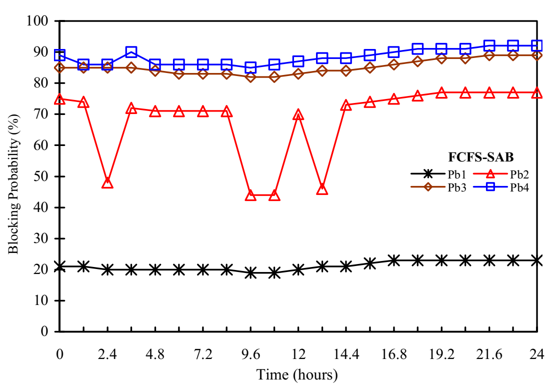


Fig. 7. The Blocking Probability of *FCFS-SAB* during 1 Day

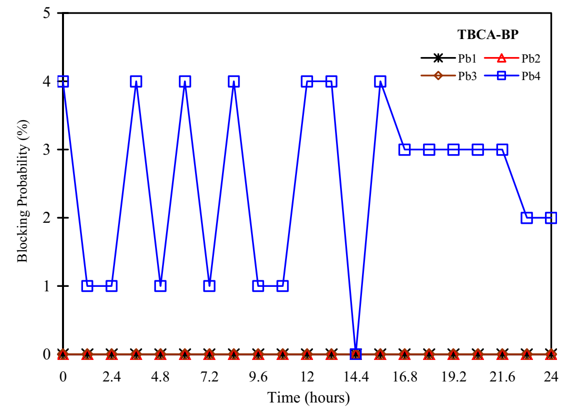


Fig. 8. The Blocking Probability of *TBCA-BP* during 1 Day

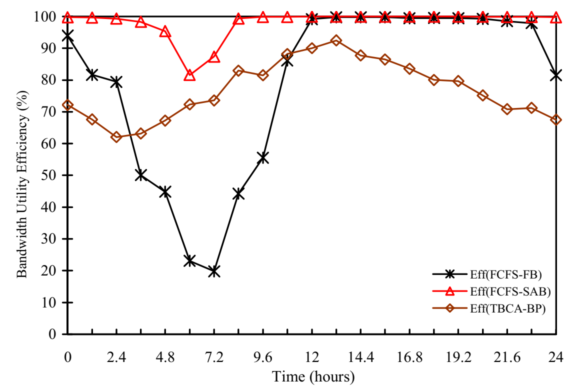


Fig. 9. The Bandwidth Utility Efficiency of All 3 Algorithms during 1 Day

V. OPNET SIMULATION STUDY OF SIP CALL SETUP DELAY

This section depicts the OPNET simulation study of SIP call setup delay in both traditional and traffic aggregation based SIP over MPLS network architectures. Fig.10 and Fig.11 demonstrate the network deployment and parameter setup of the OPNET simulation scenario for traditional SIP over MPLS network architecture. Fig.10 shows that the network contains 8 Label Switching Routers (LSRs) and 6 Label Edge Routers (LERs). In each sub-network attached to the LER, there are 2 SIP clients, 1 SIP server, 1 switch and 1 gateway. To enable MPLS switching, we set up 36 dynamic label switched paths (dynamic LSPs) between all the LERs. In this simulation, the SIP clients in all sub-networks are configured to call each other randomly according to the profile attributes in Fig.11. As a typical simulation result, the SIP call setup delay of Client_142_2 during 3 hour simulation is demonstrated in Fig.12. We can see that the SIP call setup delay varies between 3 and 4 seconds when traffic engineering enabled MPLS network is passed and dynamic LSP setup is needed. On the other hand, the delay is much shorter when Client_142_2 calls its neighbour (Client_142_1) in the same Ethernet based sub-network.

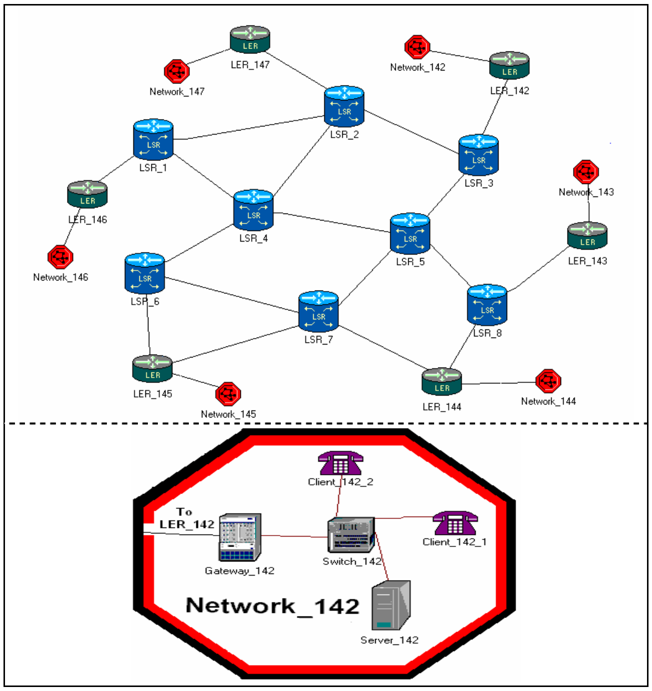


Fig. 10. Network Deployment in the OPNET Simulation of Traditional SIP over MPLS Network Architecture

Routing Protocols

IP Dynamic Routing Protocol	OSPF
IP Interface Addressing Mode	Auto Addressed
LDP Discovery End Time	250
LDP Discovery Start Time	100
LSP Signaling Protocol	RSVP
RSVP Sim Efficiency	Enabled
OSPF Sim Efficiency	Enabled
OSPF Stop Time	260

Application attributes

Silence Length (seconds)	default
Talk Spurt Length (seconds)	default
Symbolic Destination Name	Voice Destination
Encoder Scheme	G.729
Voice Frames per Packet	10
Type of Service	Interactive Voice (6)
RSVP Parameters	(...)
Traffic Mix (%)	All Discrete
Signaling	SIP

Profile attributes

Profile Name	Applications	Operation Mode	Start Time (seconds)	Duration (seconds)	Repeatability	
Client_1	Name	Serial (Ordered)	uniform (260, 300)	End of Simulation	Attribute	Value
	Voice				Inter-repetition Time (seconds)	constant (200)
	Start Time Offset (se..				Number of Repetitions	constant (0)
	uniform (5,10)				Repetition Pattern	Serial
	Duration (seconds)					
	constant (10)					
	Repeatability					
	Unlimited					

Fig. 11. Parameter Setup in the OPNET Simulation of Traditional SIP over MPLS Network Architecture

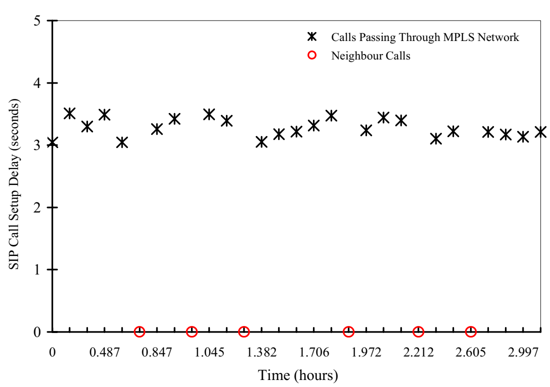
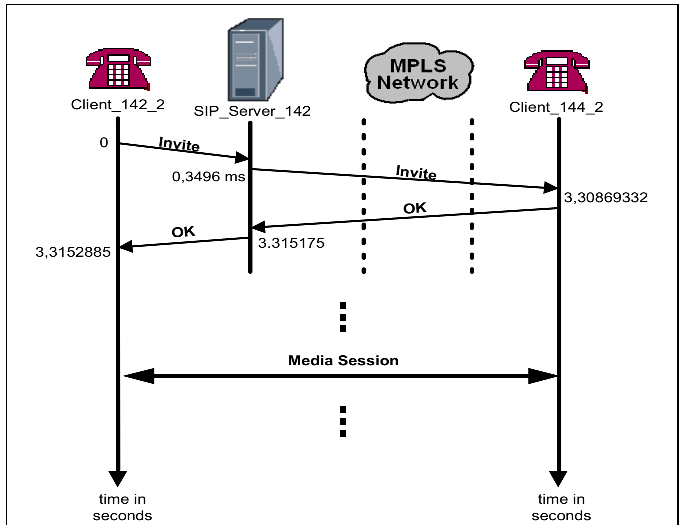


Fig. 12. SIP Call Setup Delay of Client_142_2 during 3 Hour Simulation of Traditional SIP over MPLS Network Architecture

To understand where the SIP call setup delay comes, we use OPNET Debugger to trace the SIP messages. As shown in Fig.13, it takes the first SIP message (Invite) about 3.3 seconds to travel from Client_142_2 to Client_144_2 due to LSP setup delay. After that, the transmission delay of other SIP messages is very short. Therefore, we can conclude that the SIP call setup delay is mainly caused by the LSP setup between two LERs.



* Time : 1967.37172853 sec, [00d 00h 32m 47s . 371ms 728us 528ns 345ps]
* Event : execution ID (314336), schedule ID (#325045), type (remote intrpt)
> Module : top.Network_142.Client_142_2.application (processor)

UAC (PID 1336) processing SIP INVITE Request
Sending the request to UAS in packet (PK_ID 0.000000)

* Time : 1967.37185173 sec, [00d 00h 32m 47s . 371ms 851us 726ns 461ps]
* Event : execution ID (314407), schedule ID (#325117), type (stream intrpt)
> Module : top.Network_142.Server_142.application (processor)

UAS (PID 1338) has received a request in packet (PK_ID 0.000000)
UAS (PID 1338) opening an active (tcp) connection to dest (Network_144.Client_144_2) on port (507)

* Time : 1970.50776864 sec, [00d 00h 32m 50s . 507ms 768us 641ns 539ps]
* Event : execution ID (315321), schedule ID (#326046), type (stream intrpt)
> Module : top.Network_144.Client_144_2.application (processor)

UAC (PID 1341) has received a response.
UAC (PID 1341) processing SIP SIPC_CALL_INVITE Response
Sending a SIPC_CALL_CONNECT_SUCCESS ACK to UAS

* Time : 1970.51559064 sec, [00d 00h 32m 50s . 515ms 590us 637ns 083ps]
* Event : execution ID (315428), schedule ID (#326155), type (stream intrpt)
> Module : top.Network_142.Server_142.application (processor)

UAS (PID 1338) has received an ACK.
UAS (PID 1338) has updated the message on response packet (PK_ID 0.000000) to (SIPC_CALL_CONNECT_SUCCESS).

* Time : 1970.51570384 sec, [00d 00h 32m 50s . 515ms 703us 835ns 199ps]
* Event : execution ID (315466), schedule ID (#326195), type (stream intrpt)
> Module : top.Network_142.Client_142_2.application (processor)

UAC (PID 1336) has received a response.
UAC (PID 1336) processing SIP SIPC_CALL_CONNECT_SUCCESS Response
Forwarding the response to Application Process (PID 1344)

Fig. 13. SIP Message Tracing by OPNET Debugger

To prove the advantage of traffic aggregation based SIP over MPLS network architecture, we build the algorithm of *TBCA-BP* into the TA server, and do the OPNET simulation again with the same network deployment and parameter setup as demonstrated in Fig.10 and Fig.11. The simulation result is shown in Fig.14, which can justify that the traffic aggregation based SIP over MPLS network architecture does have a good performance in terms of call setup delay. Compared with the result in Fig.12, we can see very clearly that the SIP call passing through MPLS network in the new network architecture usually has a call setup delay below 0.6 second, while its counterpart in traditional SIP over MPLS network architecture has to suffer a call setup delay above 3 seconds.

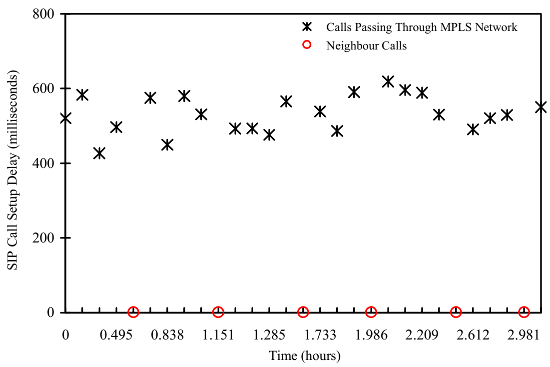


Fig. 14. SIP Call Setup Delay of Client_142_2 during 3 Hour Simulation of Traffic Aggregation Based SIP over MPLS Network Architecture

VI. Conclusion

We have developed a set of SIP over MPLS simulation models using OPNET network simulation environment to investigate the performance of the traffic aggregation based SIP over MPLS network architecture. Numerical results from the simulation prove that this new network architecture has satisfying performance due to its short call setup delay, low call blocking probability, and high bandwidth utility efficiency. Currently, we are extending our research to the 3G IP-based mobile networks, where SIP has been selected as the call control protocol and MPLS technology is utilized in the core network.

Acknowledgment

The authors would like to thank Bell Canada and NSERC for their financial support.

References

[1] Deering S.E., "SIP: Simple Internet Protocol," *IEEE Network*, Volume: 7, Issue: 3, May 1993, pp. 16-28.

[2] Yuan Zhang, "SIP-based VoIP network and its interworking with the PSTN," *Electronics & Communication Engineering Journal*, Volume: 14, Issue: 6, Dec. 2002, pp. 273-282.

[3] Sisalem D., Fiedler J., Ruppelt R., "SIP and IPv6: why and how?," *Proceedings of Symposium on Applications and the Internet Workshops 2003*, Jan. 2003, pp. 222-225.

[4] Bozinovski M., Gavrilovska L., Prasad R., "Fault-tolerant SIP-based call control system," *Electronics Letters*, Volume: 39, Issue: 2, Jan. 2003, pp. 254-256.

[5] Kawarasaki Y., Shibata T., Takahashi T., "IPv4/IPv6 SIP interworking methods in dual-stack network," *The 9th Asia-Pacific Conference on Communications*, Volume: 3, Sept. 2003, pp. 1124-1128.

[6] Faccin S.M., Lalwaney P., Patil B., " IP multimedia services: analysis of mobile IP and SIP interactions in 3G networks," *IEEE Communications Magazine*, Volume: 42, Issue: 1, Jan. 2004, pp. 113-120.

[7] Hyun Wook, Jaecheon Han, Miyoung Huh, Sunok Park, ShinGak Kang, "Study on robust billing mechanism for SIP-based internet telephony services," *The 6th International Conference on Advanced Communication Technology*, Volume: 2, 2004, pp. 756-759.

[8] Zahariadis T., Spanos S., "The clearest voice [SIP vs H.323]," *Communications Engineer*, Vol.2, Issue: 2, April-May 2004, pp. 14-17.

[9] Le Faucheur F., "IETF Multiprotocol Label Switching (MPLS) Architecture," *1st IEEE International Conference on ATM (ICATM-98)*, June 1998, pp. 6-15.

[10] Hyeong Ho Lee, Bu Ihl Kim, Jae Sup Lee, Chu Hwan Yim, "Structures of an ATM switching system with MPLS functionality," *GLOBECOM'99*, Volume: 1B, 1999, pp. 616-620.

[11] Li T., "MPLS and the evolving Internet architecture," *IEEE Communications Magazine*, Volume: 37, Issue: 12, Dec. 1999, pp. 38-41.

[12] Armitage G., "MPLS: the magic behind the myths [multiprotocol label switching]," *IEEE Communications Magazine*, Volume: 38, Issue: 1, Jan. 2000, pp. 124-131.

[13] Kankkunen A., "MPLS and next generation access networks," *ECUMN 2000*, Oct. 2000, pp. 5-16.

[14] Awduche D.O., "MPLS and traffic engineering in IP networks," IEEE Communications Magazine, Volume: 37, Issue: 12, Dec. 1999, pp. 42-47.

[15] Xipeng Xiao, Hannan A., Bailey B., Ni L.M., "Traffic engineering with MPLS in the Internet," IEEE Network, Volume: 14, Issue: 2, March-April 2000, pp. 28-33.

[16] Jong-Moon Chung, "Analysis of MPLS traffic engineering," *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems*, Volume: 2, Aug. 2000, , pp. 550-553.

[17] Zhang C., Guy C.G., "TE-SIP server design for a SIP-over-MPLS based network," *Proceedings of International Conference on Communication Technology 2003 (ICCT 2003)*, Volume: 2, April 2003, pp. 1758-1761.

[18] Salsano S., Veltri L., "QoS control by means of COPS to support SIP-based applications," *IEEE Network*, Volume: 16, Issue: 2, March-April 2002, pp. 27-33.