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—**The next generation communication system will provide high quality multimedia service in a more flexible and intelligent manner. In this paper, we propose a new SIP over MPLS network architecture to achieve this goal. To integrate SIP protocol with the traffic engineering (TE) function of MPLS network seamlessly and facilitate SIP call setup, the SIP-MPLS traffic aggregation server (TA server) is highlighted in this new architecture. Furthermore, the call admission and bandwidth re-negotiation algorithms running on the TA server are investigated carefully. We rely on OPNET simulation to assess the performance of our network architecture and the capability of different call admission and bandwidth re-negotiation algorithms. The simulation results show that the algorithm of *TBCA-BP* is a suitable candidate for accomplishing the call admission and bandwidth re-negotiation task due to its satisfying performance, and our network architecture is able to shorten the SIP call setup delay greatly by using the TA server.**

I. Introduction

IP telephony is becoming one of the most promising research areas today, because both Internet Sevice Providers (ISP) and traditional telephony companies can benefit from this technology. 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 [1,2,3]. One of the main drivers for this fact is the decision of 3GPP (Third-Generation Partnership Project) that, in the year 2000, SIP was selected as the call control protocol for 3G IP-based mobile networks. As for the second issue, Multi-protocol Label Switching (MPLS) is suggested as an apt switching technology for future core networks [4,5]. 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 [6,7]. 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 [8,9], according to our study, directly using SIP protocol over traffic engineering enabled MPLS network as they suggested can cause an unbearable long delay when the SIP call connection is set up. We reach this conclusion by theoretical analysis and OPNET simulation. For the above reason, we propose a new network architecture of SIP over MPLS to solve this problem. As a crucial part of our architecture, the SIP-MPLS traffic aggregation server (TA server) is highlighted in this paper. 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 greatly.

The rest of the paper is organized as follows. Firstly, we discuss the SIP call setup delay problem existed in SIP over MPLS technology in Section 2. Then, to solve this problem, we propose a new SIP over MPLS network architecture in Section 3. Furthermore, the call admission and bandwidth re-negotiation algorithms running on TA servers are suggested in Section 4, and the OPNET simulation results of these algorithms are presented in Section 5. In the end, Section 6 summarizes our results.

II. The Call Setup Delay Problem in SIP over MPLS Technology

According to our study, directly using SIP protocol over traffic engineering enabled MPLS network as suggested in [8,9] 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. Our study shows that the 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.

Our conclusion is supported by the OPNET simulation results, which are achieved on OPNET Modeler 10.0 with SIP and MPLS models. Fig.1 demonstrates the network deployment of our OPNET simulation scenario, which 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) using RSVP-TE between all the LERs. In our simulation, the SIP clients in all sub-networks are configured to call each other randomly and periodically. As a typical simulation result, the SIP call setup delay of Client_142_2 during 3 hour simulation is demonstrated in Fig.2. 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 lower when Client_142_2 calls its neighbour (Client_142_1) in the same Ethernet based sub-network.

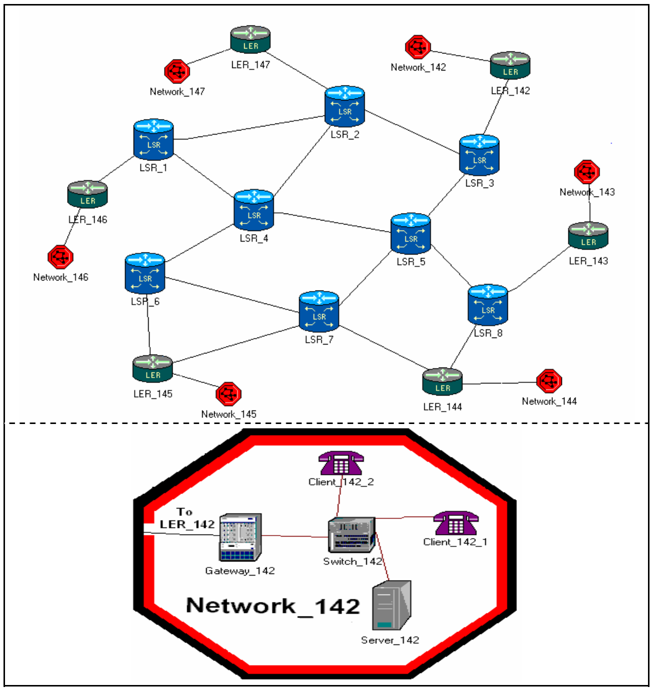

Figure 1: The Network Deployment in Our OPNET Simulation

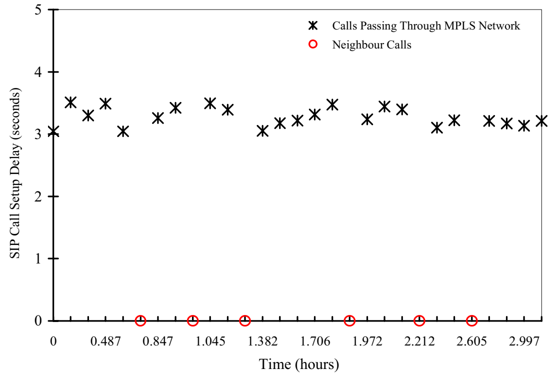

Figure 2: The SIP Call Setup Delay of Client_142_2 during 3 Hours

III. Our New SIP over MPLS Network Architecture

In order to shorten the SIP call setup delay caused by CR-LDP or RSVP-TE, a new SIP over MPLS network architecture is proposed in this section. Fig.3 shows that our SIP over MPLS architecture contains 2 parts, the local client network and 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 our 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}, 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 our SIP over MPLS network architecture.

Fig.4 shows the signaling flow in our SIP over MPLS 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.

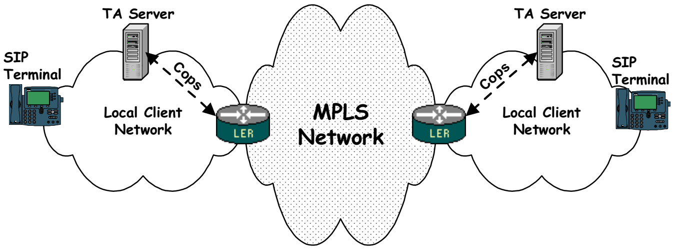

Figure 3: Our SIP over MPLS Architecture

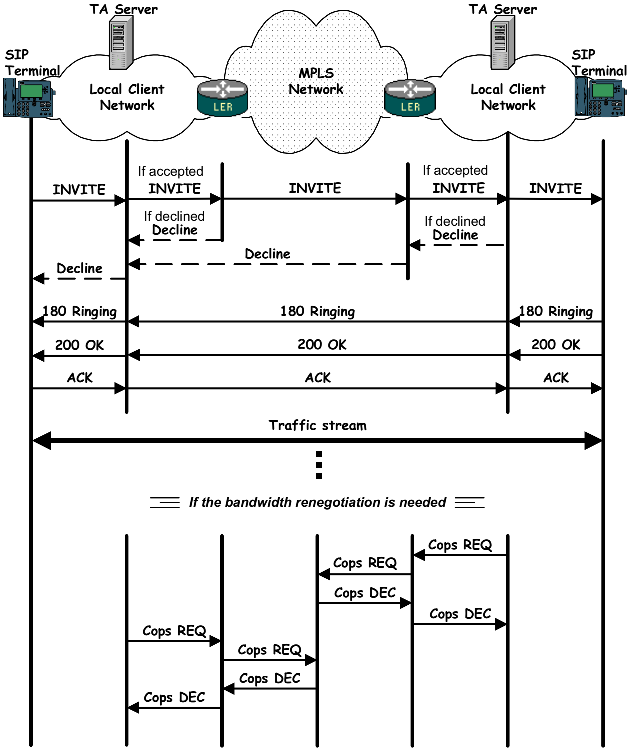

Figure 4: The Signaling Flow in Our SIP over MPLS Architecture

IV. 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 some obvious drawbacks except its advantage of simplicity. When the outgoing traffic of local client network is less than the contracted bandwidth, some bandwidth resource is wasted. On the contrary, when the outgoing traffic of local client network overweighs the contracted bandwidth, the call blocking probability is quite high.

(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*. Nevertheless, the outgoing traffic of local client network often behaviors unexpectedly due to all kinds of reasons. 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., making SIP call admission decision and conducting 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 $inc >N_{inc-thr}$
      /*make bandwidth requirement prediction*/
      $b_{contract}= b_{contract}+B_{contract-step}$
      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 $dec >N_{dec-thr}$
        /*make bandwidth requirement prediction*/
        $b_{contract}= b_{contract} -B_{contract-step}$
        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 $inc >N_{inc-thr}$
          /*make bandwidth requirement prediction*/
          $b_{contract}= b_{contract}+B_{contract-step}$
          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 $b_{occupied} + b_{req} <b_{acc-thr}$, 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 $b_{acc-thr}< b_{occupied} + b_{req} <b_{contract}$, 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 $b_{acc-thr}< b_{occupied} + b_{req} <b_{contract}$, 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 $N_{inc-thr}$ 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 $b_{dec-thr}$ for $N_{dec-thr}$ 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 $b_{acc-thr}< b_{occupied} + b_{req} <b_{contract}$, 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.

V. OPNET SIMULATION

In this section, the simulation results are presented to demonstrate the performance of the algorithms suggested in Section 4. 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.

With OPNET simulation, numerical results are attained from Fig.5 to Fig.9 to illustrate the performance of these three algorithms. The Parameter Setup of our 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:	
$\boldsymbol{b}_{contract}$	*Initial value=150Mbps, when time=0;*
$\boldsymbol{b}_{acc\text{-}thr}$	70% *of* $b_{contract}$
$\boldsymbol{b}_{dec\text{-}thr}$	50% *of* $b_{contract}$

Constants:	
$B_{contract\text{-}step}$	20% *of* $\boldsymbol{b}_{contract}$
$B_{acc\text{-}thr\text{-}step}$	70% *of* $\boldsymbol{b}_{contract\text{-}step}$
$B_{dec\text{-}thr\text{-}step}$	50% *of* $\boldsymbol{b}_{contract\text{-}step}$
$N_{inc\text{-}thr}$	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 un-flexibility. 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.

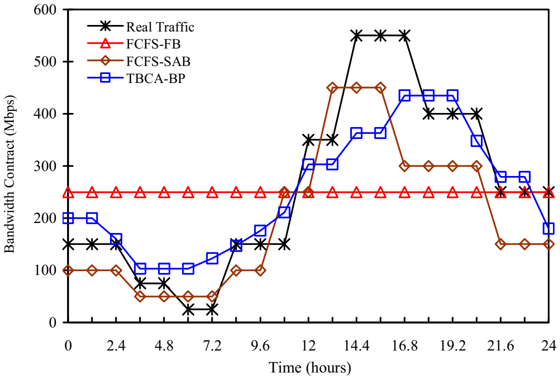

Figure 5: The Bandwidth Contract Changing of All 3 Algorithms during 1 Day

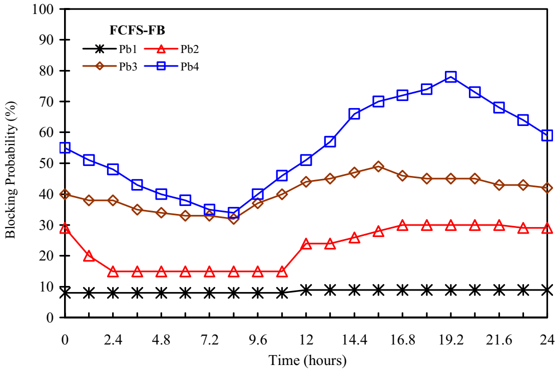

Figure 6: The Blocking Probability of *FCFS-FB* during 1 Day

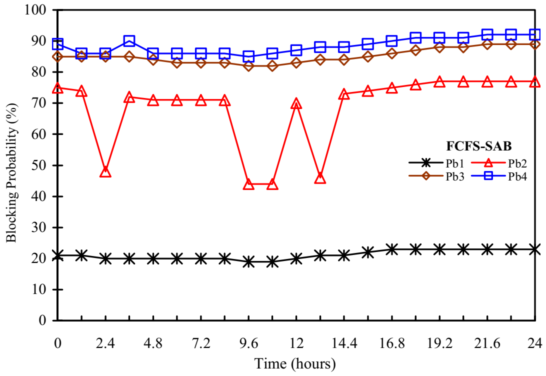

Figure 7: The Blocking Probability of *FCFS-SAB* during 1 Day

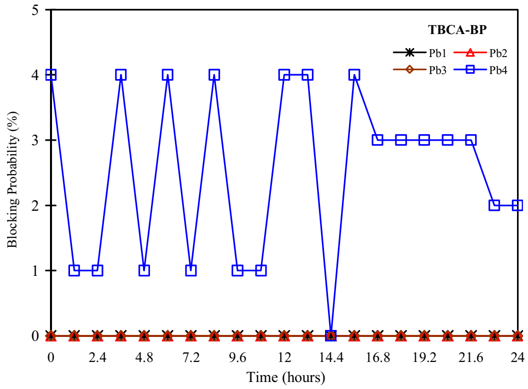

Figure 8: The Blocking Probability of *TBCA-BP* during 1 Day

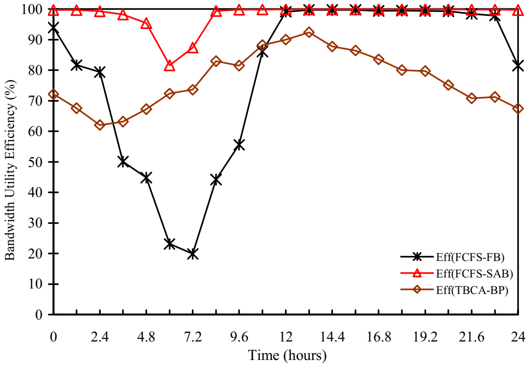

Figure 9: The Bandwidth Utility Efficiency of All 3 Algorithms during 1 Day

As mentioned in section 2, one main aim of our new SIP over MPLS architecture is to shorten the SIP call setup delay. For the sake of this goal, we build the algorithm of *TBCA-BP* into the TA server, and do the OPNET simulation again with the same network deployment as demonstrated in Fig.1. The simulation result is shown in Fig.10, which can convince us that our SIP over MPLS architecture has a good performance in terms of call setup delay. Compared with the result in Fig.2, we can see very clearly that the SIP call passing through MPLS network in our architecture usually has a call setup delay below 0.6 second, while its counterpart in traditional SIP over MPLS architecture has to suffer a call setup delay above 3 seconds.

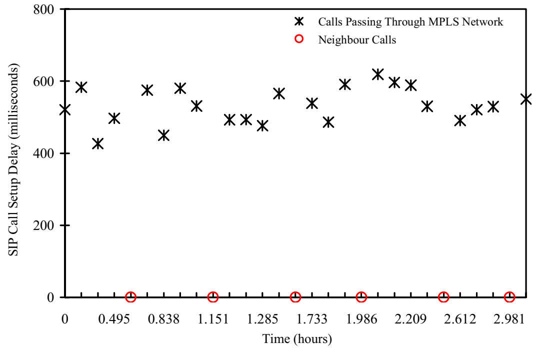

Figure 10: The SIP Call Setup Delay of Client_142_2 in Our SIP over MPLS Architecture during 3 Hours

VI. Conclusion

We have presented a new architecture of SIP telephony over MPLS to overcome the long call setup delay problem in traffic engineering enabled MPLS network. In order to integrate SIP protocol with the traffic engineering function of MPLS network seamlessly and facilitate SIP call setup, the TA server is highlighted as a key part of our architecture. Furthermore, the call admission and bandwidth re-negotiation algorithms running on the TA server are investigated carefully. In this paper, we rely on OPNET simulation to show the performance of different call admission and bandwidth re-negotiation algorithms. From the simulation results we can conclude that the algorithm named *TBCA-BP* is a suitable call admission and bandwidth re-negotiation algorithm for TA servers, due to its satisfying performance of high bandwidth prediction accuracy, low call blocking probability, and high bandwidth utility efficiency.

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] 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.

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

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

[6] 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.

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

[8] 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.

[9] 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.