OPNET Simulation of SIP Based IP Telephony over MPLS Network

Bo Rong, Jacques Lebeau, Maria Bennani, Ahmed K. Elhakeem*, and Michel Kadoch
LAGRIT, Department of Electrical Engineering, Ecole de technologie superieure, Universite du Quebec
Montreal, Quebec, Canada, H3C 1K3
E-mail: {bo.rong, jacques.lebeau}@lagrit.etsmtl.ca, {mbennani, kadoch}@ele.etsmtl.ca
**Department of Electrical and Computer Engineering, Concordia University*
Montreal, Quebec, Canada, H3G 1M8
E-mail: ahmed@ece.concordia.ca

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 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.

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) is proposed by Internet Engineering Task Force (IETF) as a standard for IP telephony [1,2,3]. In the panorama of protocols for IP telephony, SIP is not the only possible choice. For instance, ITU-T H.323 is an alternative candidate. However, SIP has recently gained the increasing interest from universities, standardization organizations and companies. 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]. In MPLS network, the packets are labeled at the edge of the network and then routed through the network based on simple labels. This enables explicit routing and differentiated treatment of packets while keeping the core routers simple. 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 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 (TE) enabled MPLS network as previous literature suggested can cause a long delay when the SIP call connection is set up. We reach this conclusion by theoretical analysis and OPNET simulation. In the TE enabled MPLS network, when a new SIP call request comes, usually CR-LDP or RSVP-TE will be employed to set up a corresponding LSP dynamically. Our study shows that the dynamic LSP setup by CR-LDP or RSVP-TE can contribute a lot to the whole SIP call setup delay in TE enabled MPLS network.

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.

The SIP Call Setup Delay Problem in SIP over MPLS Technology
The SIP protocol has been defined by the IETF as a signaling protocol to initiate voice, video, and multimedia sessions, and it is an important candidate for call setup signaling in IP telephony. SIP is a very flexible protocol, because it can easily work together with QoS reservation and/or admission control mechanisms.

Multi-protocol Label Switching (MPLS) is a new technology for IP core network. It does not replace IP routing, but work together with existing and future routing protocols to provide very high-speed data forwarding and guarantee QoS requirements of different users. Many of today's discussions regarding MPLS revolve around traffic engineering. Traffic engineering is aimed to make the best use of resources across entire network by distributing traffic over different paths. Traffic engineering offers VoIP providers the chance of utilizing network resources fully and providing the QoS guaranteed service in a busy network effectively.

It is a good idea to deploy SIP based IP telephony in MPLS network [8,9]. However, according to our study, directly using SIP protocol over TE enabled MPLS network as previous literature suggested can cause an unbearable long SIP call setup delay. In TE enabled MPLS network, when a new SIP call request comes, usually CR-LDP or RSVP-TE will be employed to set up a corresponding LSP dynamically. Our study shows that the dynamic LSP setup can contribute a lot to the whole SIP call setup delay in TE enabled MPLS network. If we use 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 and Fig.2 demonstrate the network deployment and parameter setup in our OPNET simulation scenario.

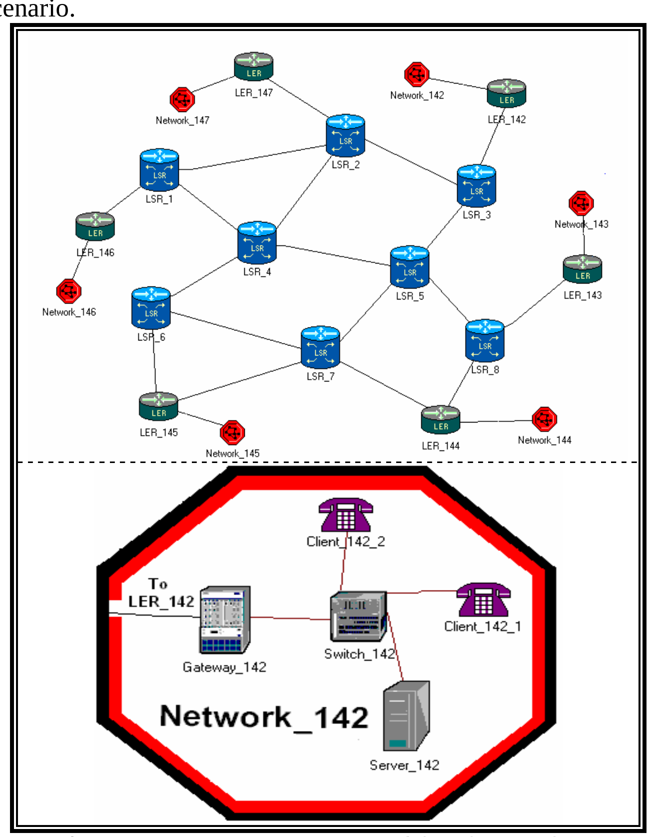
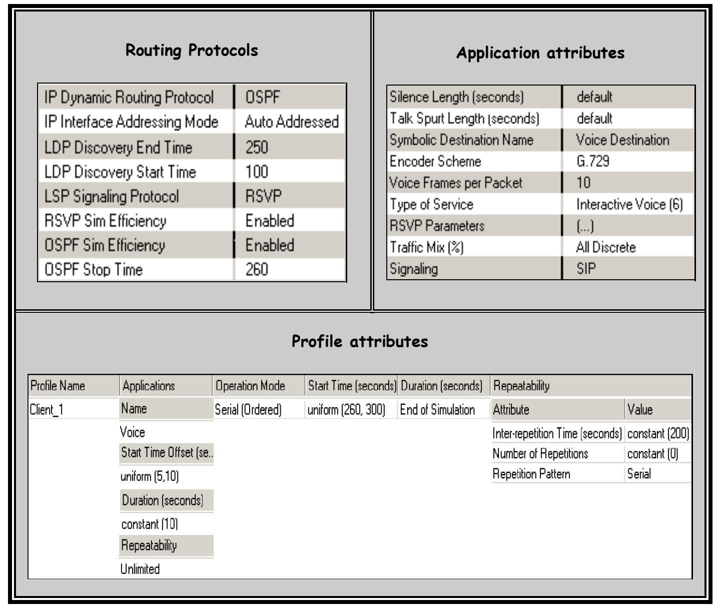


Figure 1: The Network Deployment of Our OPNET Simulation Scenario



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					

Figure 2: The Parameter Setup in Our OPNET Simulation Scenario

Fig.1 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 our simulation, the SIP clients in all sub-networks are configured to call each other randomly according to the profile attributes in Fig.2. In Fig.3, the SIP call setup delay of Client_142_2 during 3 hour simulation is demonstrated. We can see that the SIP call setup delay varies between 3 and 4 seconds when TE enabled MPLS network is used. 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.

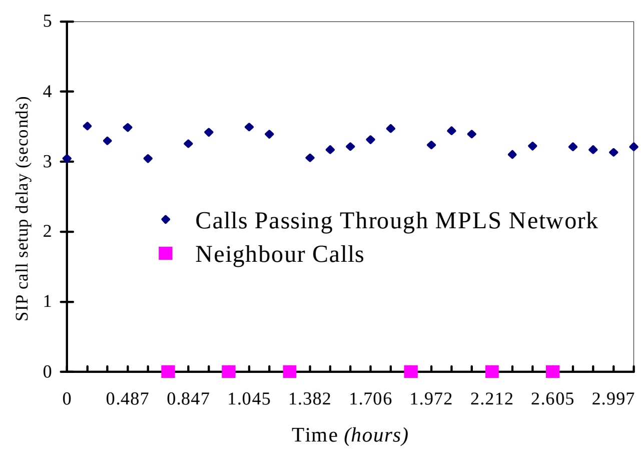


Figure 3: The SIP Call Setup Delay of Client_142_2 during 3 Hours

To understand where the SIP call setup delay comes, we use OPNET Debugger to trace the SIP messages. As shown in Fig.4, it takes the first SIP message (Invite) about 3.3 seconds to travel from Client_142_2 to Client_144_2 because of LSP setup delay. After that, the transmission delay of other SIP messages is very low. Therefore, we can conclude that the SIP call setup delay is mainly caused by the LSP setup between two LERs.

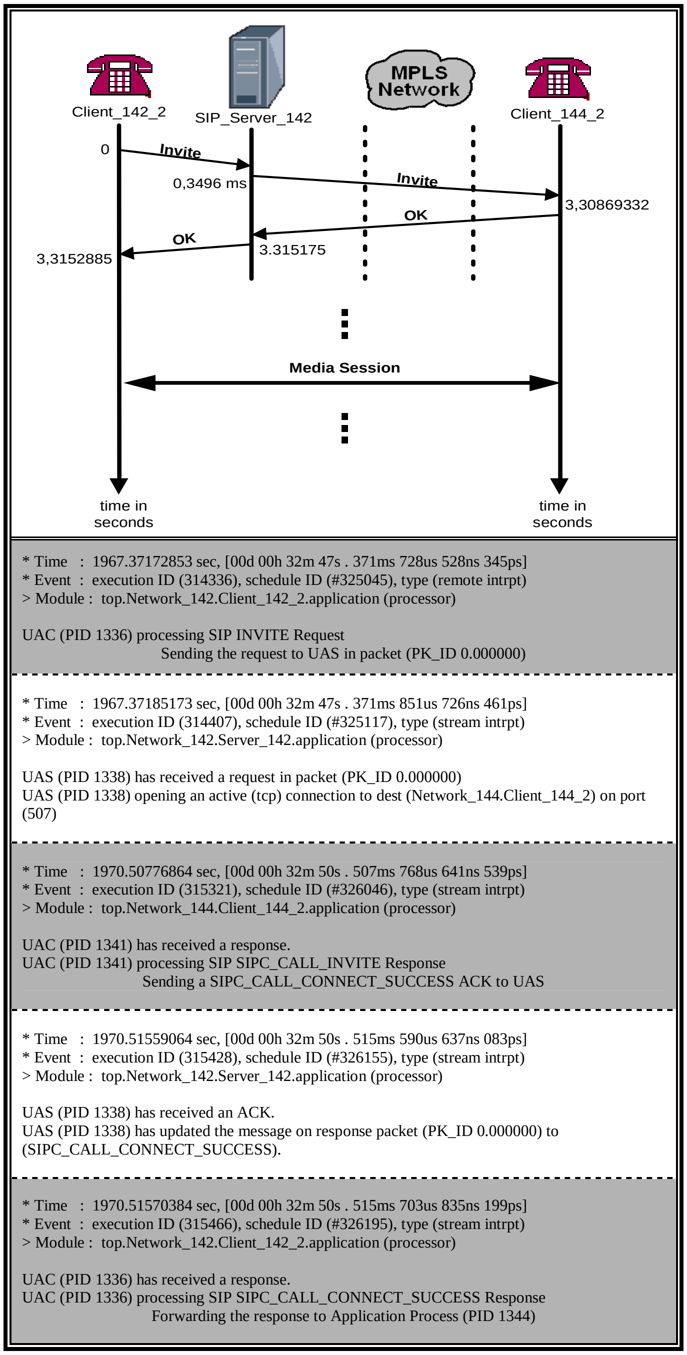


Figure 4: SIP Message Tracing by OPNET Debugger

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.5 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 $B^{pred}_{n-1,\,n} - B_{n-1,\,n} < 0$, 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 $B^{pred}_{n-1,\,n} - B_{n-1,\,n} > 0$ 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.

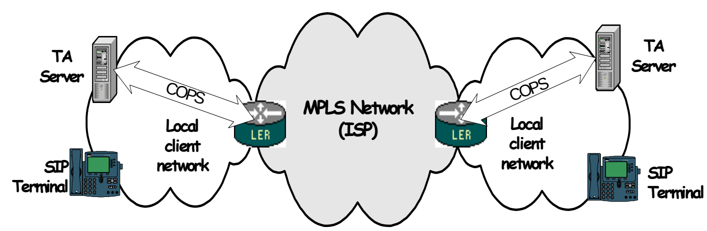


Figure 5: Our SIP over MPLS Architecture

Fig.6 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.

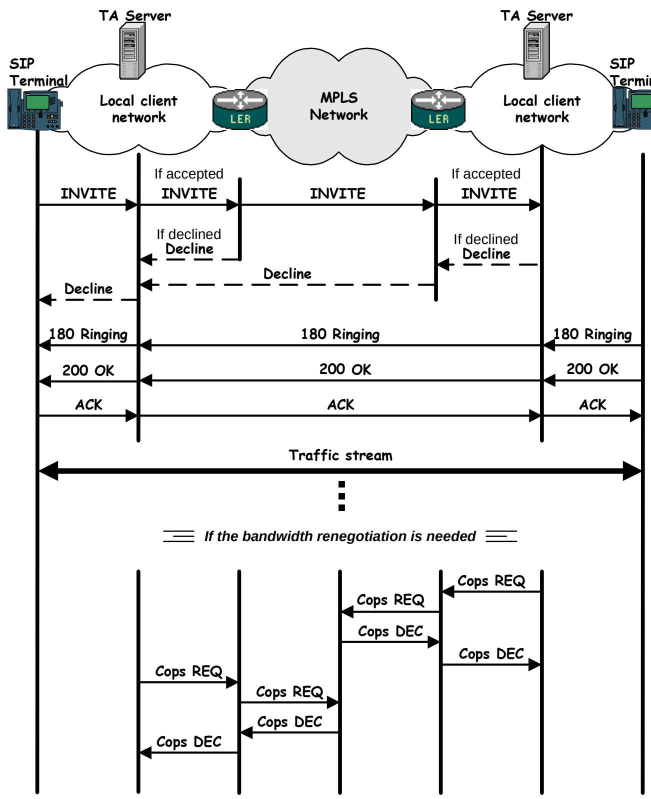


Figure 6: The Signaling Flow in Our SIP over MPLS Architecture

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 First Come First Serve (*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.

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:

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 $inc > N_{inc\text{-}thr}$
      /*make bandwidth requirement prediction*/
      $b_{contract} = b_{contract} + B_{contract\text{-}step}$
      re-negotiate with MPLS network for the new $b_{contract}$
      $b_{acc\text{-}thr} = b_{acc\text{-}thr} + B_{acc\text{-}thr\text{-}step}$
      $b_{dec\text{-}thr} = b_{dec\text{-}thr} + B_{dec\text{-}thr\text{-}step}$
      $inc = 0$
  else if $b_{occupied} + b_{req} < b_{acc\text{-}thr}$
    $b_{occupied} = b_{occupied} + b_{req}$
    accept the call
    $inc = 0$
    if $b_{occupied} < b_{dec\text{-}thr}$
      $dec = dec + 1$
      if $dec > N_{dec\text{-}thr}$
        /*make bandwidth requirement prediction*/
        $b_{contract} = b_{contract} - B_{contract\text{-}step}$
        re-negotiate with MPLS network for the new $b_{contract}$
        $b_{acc\text{-}thr} = b_{acc\text{-}thr} - B_{acc\text{-}thr\text{-}step}$
        $b_{dec\text{-}thr} = b_{dec\text{-}thr} - B_{dec\text{-}thr\text{-}step}$

dec=0
else if $b_{acc\text{-}thr} < b_{occupied} + b_{req} < b_{contract}$
$dec= 0$
$p_{decline}=(b_{occupied} + b_{req}- b_{acc\text{-}thr})/(b_{contract} - b_{acc\text{-}thr})$
With probability $p_{decline}$, decline the SIP call
if the call are really declined
$inc= inc +1$
if $inc >N_{inc\text{-}thr}$
/*make bandwidth requirement prediction*/
$b_{contract}= b_{contract}+B_{contract\text{-}step}$
re-negotiate with MPLS network for the new $b_{contract}$
$b_{acc\text{-}thr}= b_{acc\text{-}thr} + B_{acc\text{-}thr\text{-}step}$
$b_{dec\text{-}thr}= b_{dec\text{-}thr} + B_{dec\text{-}thr\text{-}step}$
inc=0
else
$b_{occupied} = b_{occupied} + b_{req}$
accept the call
inc=0

As revealed by the above description, 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.

The above discussion shows 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.

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.

The node model used in our simulation is shown in Fig.7. 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*.

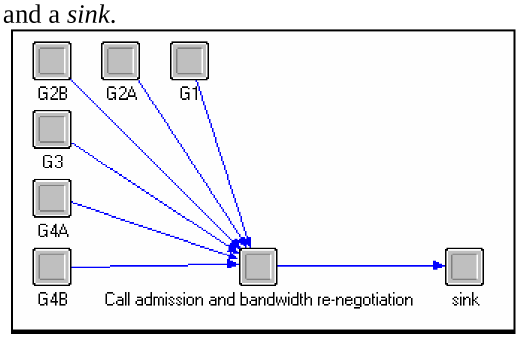


Figure 7: Node Model Used in Our Simulation

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.7 are designated to generate the call requests according to our schedule and in our own packet format. As shown in Fig.8, our packet format contains 4 fields (*groupID*, *classID*, *bandwidth requirement* and *call duration*). Generated packets from all sources are sent to the *call admission and bandwidth re-negotiation* module.

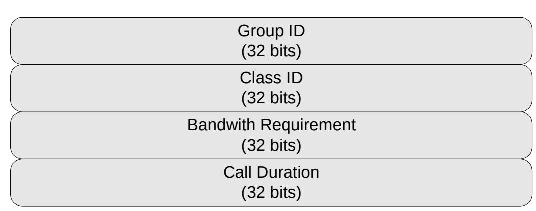


Figure 8: Packet Format in Our Simulation

After receiving a packet, the *call admission and bandwidth re-negotiation* module extracts the corresponding information of this call request from the received packet 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 and keep the information of this alive SIP call until it is out of service time and deleted. In addition, when bandwidth re-negotiation is needed, the *call admission and bandwidth re-negotiation* module would increase/decrease the contracted bandwidth accordingly.

With OPNET simulation, numerical results are attained from Fig.9 to Fig.13 to illustrate the performance of these 3 algorithms. The Parameter Setup of our simulation is shown as follows:

(1)Traffic load in the local client network:

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.9 and Fig.13 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.9, *FCFS-FB* has no capability of bandwidth prediction. As a result, Fig.10 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.13 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.9, Fig.11, and Fig.13 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.9 shows *TBCA-BP* has high bandwidth prediction accuracy, while Fig.12 and Fig.13 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.

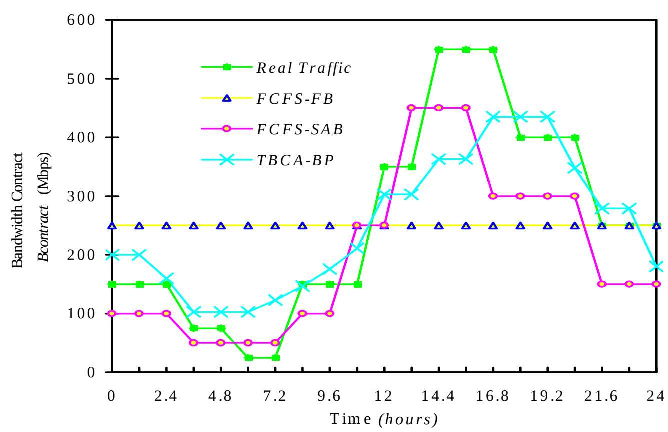


Figure 9: The Bandwidth Contract Changing of All 3 Algorithms during 1 Day

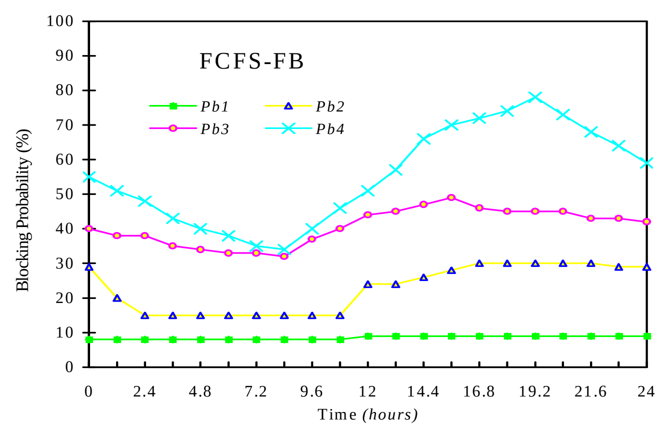


Figure 10: The Blocking Probability of *FCFS-FB* during 1 Day

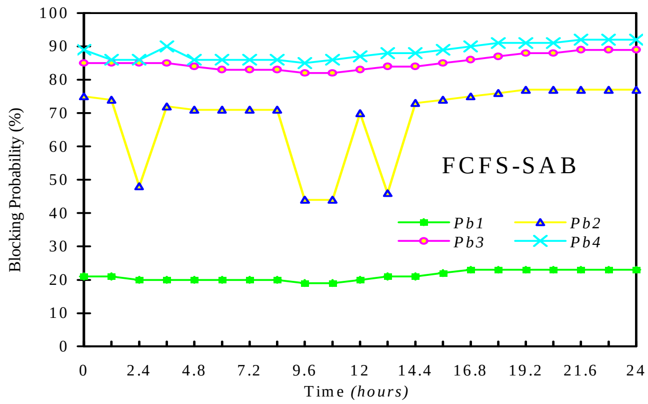


Figure 11: The Blocking Probability of *FCFS-SAB* during 1 Day

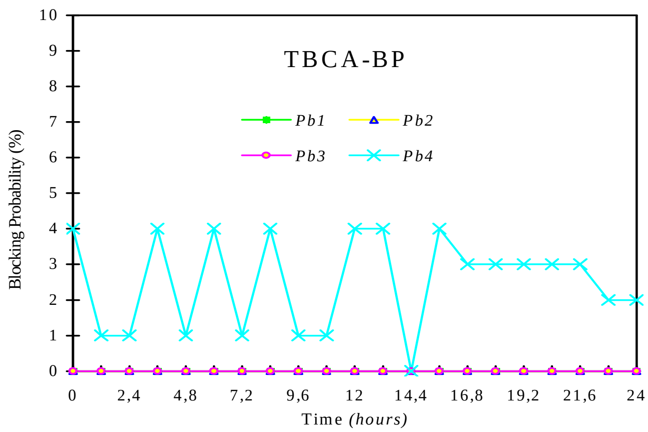


Figure 12: The Blocking Probability of *TBCA-BP* during 1 Day

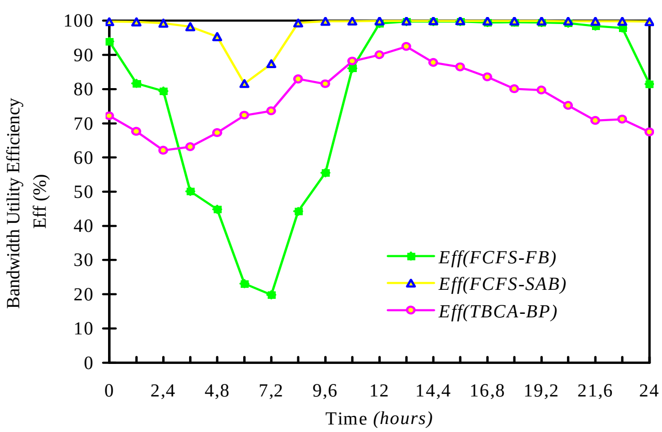


Figure 13: The Bandwidth Utility Efficiency of All 3 Algorithms during 1 Day

As mentioned in Section 2, one main aim of our new SIP over MPLS architecture is to shorten the SIP call setup delay. To prove our new architecture can reach this aim, we build the algorithm of *TBCA-BP* into the TA server, and do the simulation shown in Fig.1 and Fig.2 again. The simulation result shows that SIP call setup delay in our new architecture is similar to that of the neighbour calls in Fig.3.

Conclusion

We have presented a new architecture of SIP telephony over MPLS to overcome the long call setup delay problem in TE 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.

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.