Impact on network performance

of Common Alerting Protocol over SIP

A dissertation submitted to

University of Alberta

In partial fulfillment of the requirements for the degree of

Master of Science in Internetworking

2011

School of Electric Engineering and Computer Science

Primary supervisor: Prof. Mike MacGregor

Table of Contents

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Abstract

SIP is one of transport media for deliver CAP XML format message. Raj has been completed the simulation of CAP over SIP in his report. When an emergency event happens and most of subscribers are in one place such as University, agent will starts very large number of alerting call concurrently. This research investigates the performance of network and SIP traffic characteristics under large amount of concurrent SIP calls due to CAP alert happens.

Acknowledgements

I would like to express my gratitude to my supervisor Prof. Mike Macgregor, for his support and encouragement throughout. The support of a supervisor, who understands my unusual disposition, and subject matter, has been invaluable.

Mr. Shanawaz Smir gave me the much needed support When I couldn’t keep continue my research and helped me work through to get things done.

Mr.Moninder is a Mac guy and always there to help me with his expertises.

Special thanks to my wife Li Wang, a mother of our three kids, for her constant support, understanding, and love.

1. Introduction

Common Alerting Protocol plays a critical role in emergency altering system used in our everyday's life. It is an open, non-proprietary standard data interchange format that can be used to collect all types of hazard warnings and reports locally, regionally and nationally, for input into a wide range of information management and warning dissemination systems.

SIP (Session Initiation Protocol) is a protocol that initiates and manages interactive user session involving multimedia sessions including voice, instant message. It is widely used in Voice over IP system. With VoIP system deployed massively, it's signalling - SIP becomes the best carrier for CAP message in fact.

When emergency event happen, it is very important to distribute alerting message to all subscribers immediately, long delay and system crash are not expected. There are many factors limit the speed of distribution CAP message at agent. Total lines available at agent (proxy), number of subscriber, network load and bandwidth etc all affect the delay and performance of CAP system.

In this capstone project, the performance of CAP over SIP will be investigated under certain circumstances using OPNET. Through various simulations under realistic networking scenario, this study provides a brief overview the impact of network performance after emergency event start.

2. Overview

2.1 Background

In previous Chapter, the main objective of this research was outlined. This Chapter will provide technologies which used in this research are presented to help understand the subsequent chapters.

2.2 SIP Overview

The description of SIP is RFC 3261. SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions.

SIP only involved in the signalling portion of a communication session. It needs to work together with other protocols such as RTP (Real-time Transport Protocol), SDP (Session Descriptions Protocol) to complete a multimedia service. Features like ring, or busy signal are performed by proxy servers and user agents. So what SIP focus is providing call-setup and signalling for IP-based communication.

There are many definitions in RFC 3261, but followings are very critical to implement successful SIP system:

Client: A client is any network element that sends SIP requests and receives SIP responses. Clients may or may not interact directly with a human user. User agent clients and proxies are clients.

Invitation: An INVITE request.

Invitee, Invited User, Called Party, Callee: The party that receives an INVITE request for the purpose of establishing a new session. A callee retains this role from the time it receives the INVITE until the termination of the dialog established by the INVITE.

Message: Data sent between SIP elements as parts of the protocol. SIP messages are either requests or responses.

Proxy, Proxy Server: An intermediary entity that acts as both a server and a client for the purpose of making requests on behalf of other clients.

User Agent Client (UAC): A user agent client is a logical entity that creates a new request, and then uses the client transaction state machinery to send it. The role of UAC lasts only for the duration of that transaction. In other words, if a piece of software initiates a request, it acts as a UAC for the duration of that transaction. If it receives a request later, it assumes the role of a user agent server for the processing of that transaction.

User Agent Server (UAS): A user agent server is a logical entity that generates a response to a SIP request. The response accepts, rejects, or redirects the request. This role lasts only for the duration of that transaction. In other words, if a piece of software responds to a request, it acts as a UAS for the duration of that transaction. If it generates a request later, it assumes the role of a user agent client for the processing of that transaction.

The role of UAC and UAS, as well as proxy and redirect servers, are defined on a transaction-by-transaction basis. Usually, proxy, location, and registrar servers defined above the logical entities; implementations MAY combine them into a single application.

In OPNET simulation environment, clients and proxy server are essential elements of simulation of basic SIP call. Process model of implementation of this system includes UAC, UAC manager, UAS, UAS manager. Based on this architecture, CAP messages can be easily delivered to subscriber through SIP packets. Below is a simple SIP system diagram.

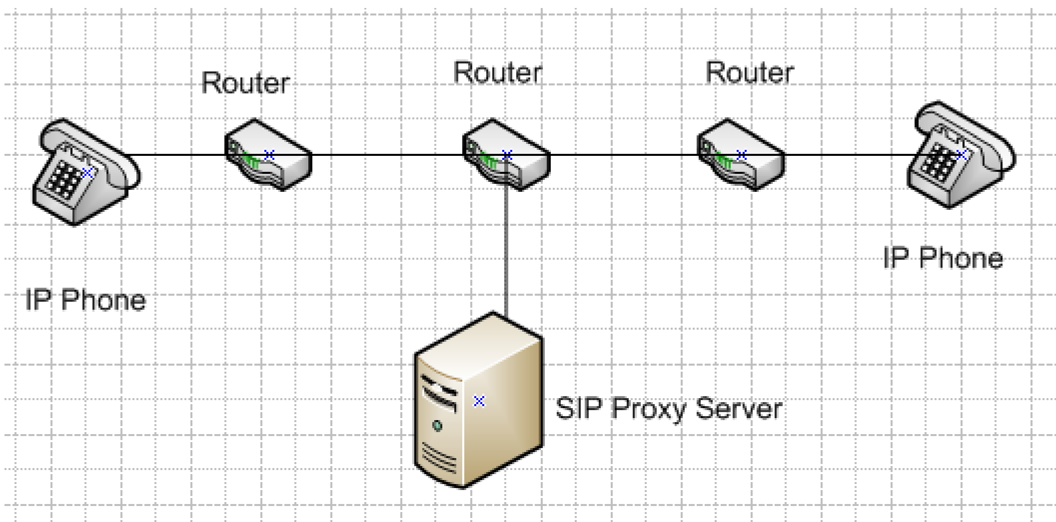


Figure 2-1 Simple SIP System Architecture

2.3 CAP overview

Common Alerting Protocol (CAP) is an XML-based data format exchanging public warnings and emergencies between alerting technologies. Before CAP was introduced there were many warning mechanisms, all of them worthwhile: EAS, Weather Radio, Telephone Alerting, Sirens, Internet, Email etc… People were getting confused by inconsistent messages and added load to emergency officials as well. With the technologies such as GIS, wireless E9-1-1, VoIP become massively deployed and part of people daily life, an exchangeable Alerting Message based on VoIP is urgently needed.

CAP meets the requirement of alerting system and independent to the system technologies which used in real world. It is compatible with SAME, sirens, telephone and other existing alerting systems. Also provides flexible geographic targeting of alerts, subscriber and cancellation features, digital images and audio, this features make CAP becomes more popular and main standard in emergency alerting system.

With the advance of technologies and emergency message digitized, localized area emergency alerting services are possible offered to specific subscriber. People can send subscribe message from cell phone, websites or calling voice system to subscribe an emergency events as they wish to be notified in the future. Subscriber can specify preferred geographical location of the notification, the types of emergency events and alert method of delivery. The subscriber will provide a local phone numbers or email addresses to the notification network, which will then monitor that location of any emergency situation it had registered for and notify it in case of an emergency.

CAP provides effective, coordinated use of limited public warning resources, simplified warning workload for hard-pressed emergency officials on their buy days. It benefits everyone who living in the area when an emergency event happened, such as natural disaster.

2.4 SIP based emergency alerting system

Now day's internet telephones and applications largely replaced traditional modes communication. Emergency Notification Systems will take advantage of internet communication technology to deliver emergency messages more effectively. SIP is an application-layer signalling protocol which can be used for events notification. There are two RFC [3265, 3903] standards detailed the special events notification method using SIP.

2.4.1 SIP-Specific Event Notification

RFC3265 defines entities in the network can subscribe to resource or call state for various resources or calls in the network, and those entities (or entities acting on their behalf) can send notifications when those states change. A typical flow of messages would be:

```
Subscriber                        Notifier
|--------------SUBSCRIBE--------→|   Request stat subscription
|<------------------200-----------------|   Acknowledge subscription
```

|<-------------NOTIFY---------------| Return current state information

|-----------------200--------------→|

|<-------------NOTIFY---------------| Return current state information

|-----------------200--------------→|

Subscriptions are expired and must be refreshed by subsequent SUBSCRIBER Messages.

2.4.2 SIP Event State Publication

RFC3903 defines a new SIP method, PUSLISH, for publishing event state. PUBLISH is similar to REGISTRER in that it allows a user to create, modify, and remove state in another entity which manages this state on behalf of the user. Addressing a PUBLISH request is identical to addressing a SUBSCRIBE request. Below is a message flow which combined SUBSCIRBE, NOTIFICATION and PUBLISH.

```
  PUA                     PA                      WATCHER
 (EPA)                   (ESC)
   |                       |                         |
   |                       | <---- M1: SUBSCRIBE --- |
   |                       |                         |
   |                       | ----- M2: 200 OK -----> |
   |                       |                         |
   |                       | ----- M3: NOTIFY -----> |
   |                       |                         |
   |                       | <---- M4: 200 OK ------ |
   |                       |                         |
   |                       |                         |
   | ---- M5: PUBLISH ---> |                         |
   |                       |                         |
   | <--- M6: 200 OK ----  |                         |
   |                       |                         |
   |                       | ----- M7: NOTIFY -----> |
   |                       |                         |
   |                       | <---- M8: 200 OK ------ |
```

```
|                         |                           |
| ---- M9: PUBLISH ---->  |                           |
|                         |                           |
| <--- M10: 200 OK ---    |                           |
|                         |                           |
|                         |                           |
| --- M11: PUBLISH --->   |                           |
|                         |                           |
| <-- M12: 200 OK ----    |                           |
|                         |                           |
|                         | ----- M13: NOTIFY ---->   |
|                         |                           |
|                         | <---- M14: 200 OK -----   |
|                         |                           |
```

A User Agent Client (UAC) that publishes event state is labelled an Event Publication Agent (EPA). The entity that processes the PUBLISH request is known as an Event State Compositor (ESC). PA and PUA are defined in RFC3852.

According to RFC3852, Presence User Agent (PUA): A Presence User Agent manipulates presence information for a presentity. This manipulation can be the side effect of some other action (such as sending a SIP REGISTER request to add a new Contact) or can be done explicitly through the publication of presence documents.

Presence Agent (PA): A presence agent is a SIP user agent which is capable of receiving SUBSCRIBE requests, responding to them, and generating notifications of changes in presence state.

Message flow:

M1: The watcher initiates a new subscription to the presentity@example.com's presence agent.

M2: The presence agent for presentity@example.com processes the subscription request and creates a new subscription. A 200 (OK) response is sent to confirm the subscription.

M3: In order to complete the process, the presence agent sends the watcher a NOTIFY with the current presence state of the presentity.

M4: The watcher confirms receipt of the NOTIFY request.

M5: A presence user agent (acting for the presentity) initiates a PUBLISH request to the presence agent in order to update it with new presence information. The Expires header field indicates the suggested duration for this event soft state.

M6: The presence agent receives, and accepts the presence publication. The published data is incorporated into the presentity's presence information.

M7: The presence agent determines that a reportable change has been made to the presentity's presence information, and sends a new presence notification to the watcher.

M8: The watcher confirms receipt of the NOTIFY request.

M9: The PUA determines that the event state it previously published is about to expire, and refreshes that event state.

M10: The presence agent receives, and accepts the publication refresh. The timers regarding the expiration of the specific event state identified by the entity-tag are updated. As always, the ESC returns an entity-tag in the response to a successful PUBLISH. Note that no actual state change has occurred, so the watchers will receive no NOTIFYs.

M11: The PUA of the presentity detects a change in the user's presence state. It initiates a PUBLISH request to the presence agent to modify the published presence information with the recent change.

M12: The presence agent receives, and accepts the modifying publication. The published data is incorporated into the presentity's presence information, updating the previous publication from the same PUA.

M13: The presence agent determines that a reportable change has been made to the presentity's presence document, and sends a new presence notification to all active subscriptions.

M14: The watcher confirms receipt of the NOTIFY request.

2.4.3 Benefits in using SIP for Emergency Notification

There are many advantages by using SIP for emergency notification system. SIP can be used on different applications and devices, such as internet telephony, multimedia applications, instant messaging, 3G cell phones, PDAs, home PC etc.

Second, More detailed description of event can be carried by SIP message, Multilingual content can included in a single message and receiver of the message can choose a language. Third, the content of SIP message automates subscription and notification process. Fourth, Resource consumption are lower capered with traditional methods.

2.4.4 SIP-Emergency Alerts System Architecture

The basic architecture is straightforward. This hierarchical architecture was proposed by Columbia University study team. Inside subscription system, where alerting message is disseminated from top (national) to bottom (local and user), and message can be generated at any level of government. Users subscribe to servers at the next higher level and may in turn become servers for the levels below. Child nodes must subscribe to their parent node servers and siblings must subscribe to each other.

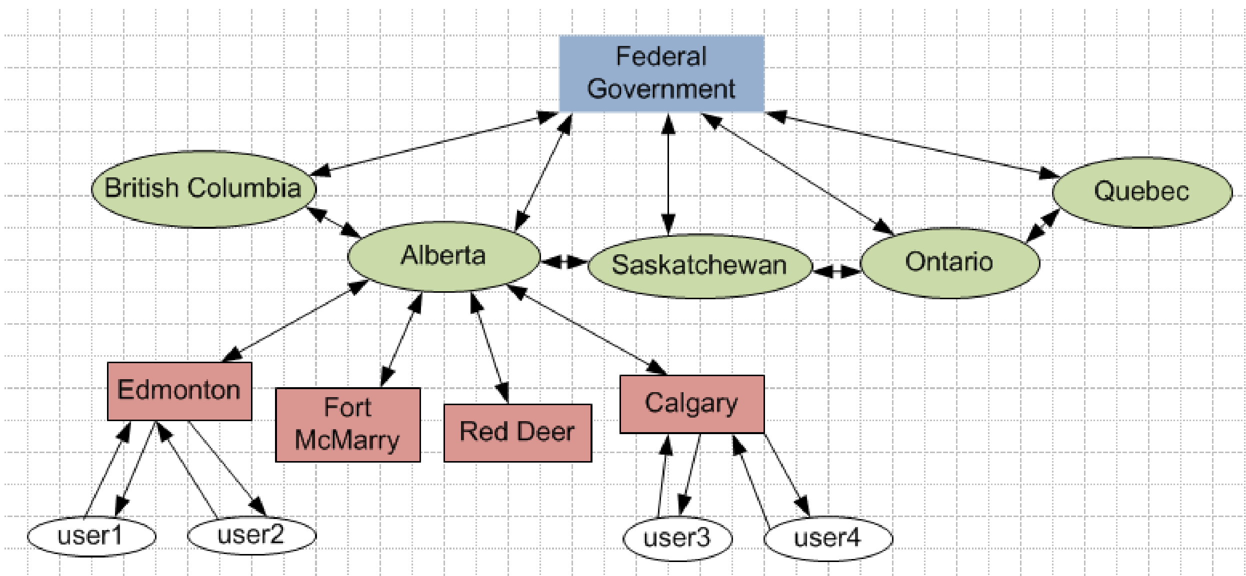


Figure 2-2 CAP System Architecture

3. Research Background

There are some researchers have been done on network performance of SIP system which use simulation tool OPNET. The scenario which they used includes Ethernet LAN and wireless LAN environment, Node number, Voice encoder schemes and traffic arrival distribution are all studied in those papers.

In the case of emergency event happen, all subscribers need to be notified as quickly as possible, that results a large amount of call simultaneously started. So the traffic arrival distribution model determines how SIP calls are generated such as constant rate, poisson distribution and exponential distribution, but no of them are simultaneously. This dissertation investigates the network performance under this extremely situation.

The simulation tool used in this project is OPNET Modular 14.5 education version. OPNET solutions model communications devices, protocols, technologies, and architectures, and simulate their performance in a dynamic virtual network environment. The output result of OPNET is very close to reality and provides full phase of study.

4. Network Modeling and Simulation

4.1 OPNET SIP Simulation

OPNET supports SIP based VoIP application and provides IP phone and SIP Proxy Server. It provides sip_UAC.pr.m, sip_UAC_mgr.pr.m, sip_UAS.pr.m, sip_UAS_mgr.pr.m four standard process models for sip based communications. In our case, one IP phone can acts as a notification server which initiates SIP call to all subscribers. In order to create an instance of an SIP application, following network elements are needed in OPNET:

- a server node (Proxy Server) which is UAS
- a client node(VoIP phone) which is UAC
- an application service template which is parameter setting for this application
- a user profile template that creates an application instance

Here, Client node and subscriber node are the same, because there is no difference in whether use a server node or a client node initiates an emergency call. UAC (server or client) send out packet to UAS (Proxy Server), and then UAS forward SIP packet to UAC (server or client) by which chosen randomly. An application service template specifies protocols and voice encoding method used in the simulation. User profile is to control how traffic is generated for an application. Many nodes can share the same user profile; most of researches adopt this configuration, because they focus on network performance other than the performance of each node.

Modelling Assumptions:

- The local area networks operate at 100Mb/s throughout the simulations.
- There is no other network traffic besides VoIP traffic in this study.
- Backhaul is T1 (1.544Mpbs).

4.2 Basic Configuration of OPNET for a SIP call

To complete a VoIP call, as mentioned previously, we need two nodes and a proxy server, routers and switches also needed to provide internet connection.

Below is network view of OPNET. Caller located in San Francisco, SIP server located in Dallas, Callee located in Pittsburgh.

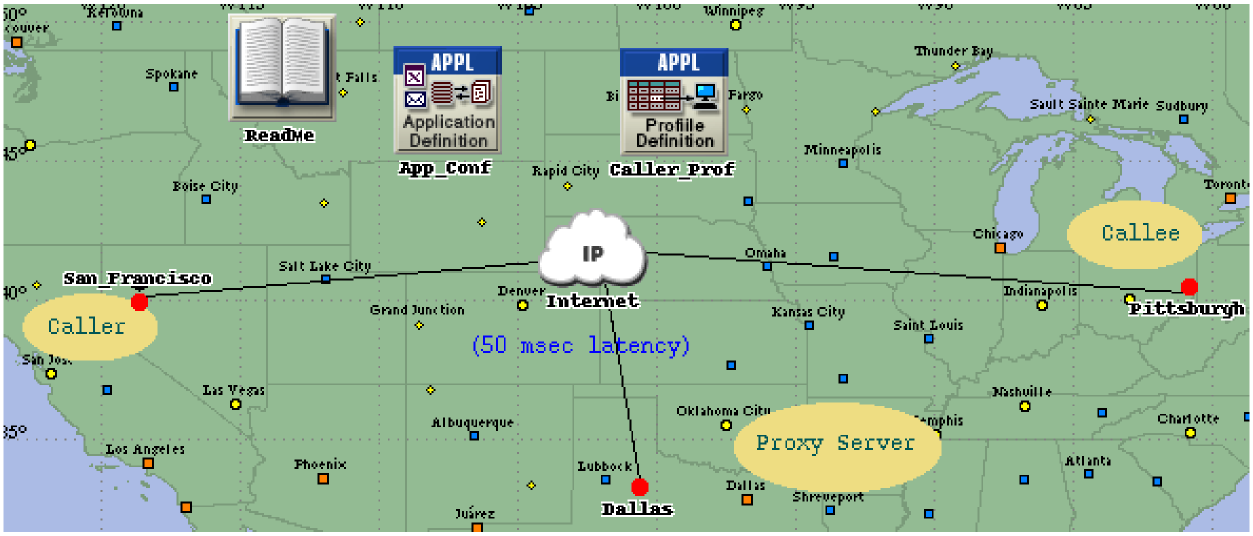


Figure 4-1 Project Overview

App_Conf: Defines Voice Encoder, Signalling Protocol, Packets per second, Packet priority and related parameters used in VoIP system. Caller, Callee and all VoIP clients will need to support this configuration to make system working.

Caller_Prof: Defines WHEN initiate SIP call, Duration of the call and HOW frequent repeats the call. In our simulation, only Caller needs to configure this profile. Because the purpose of this dissertation is to simulate the network performance when emergency events happen, and NOTIFICATION is one way, so all calls will initiate from Caller to Caller, there is no call initiate from Callee.

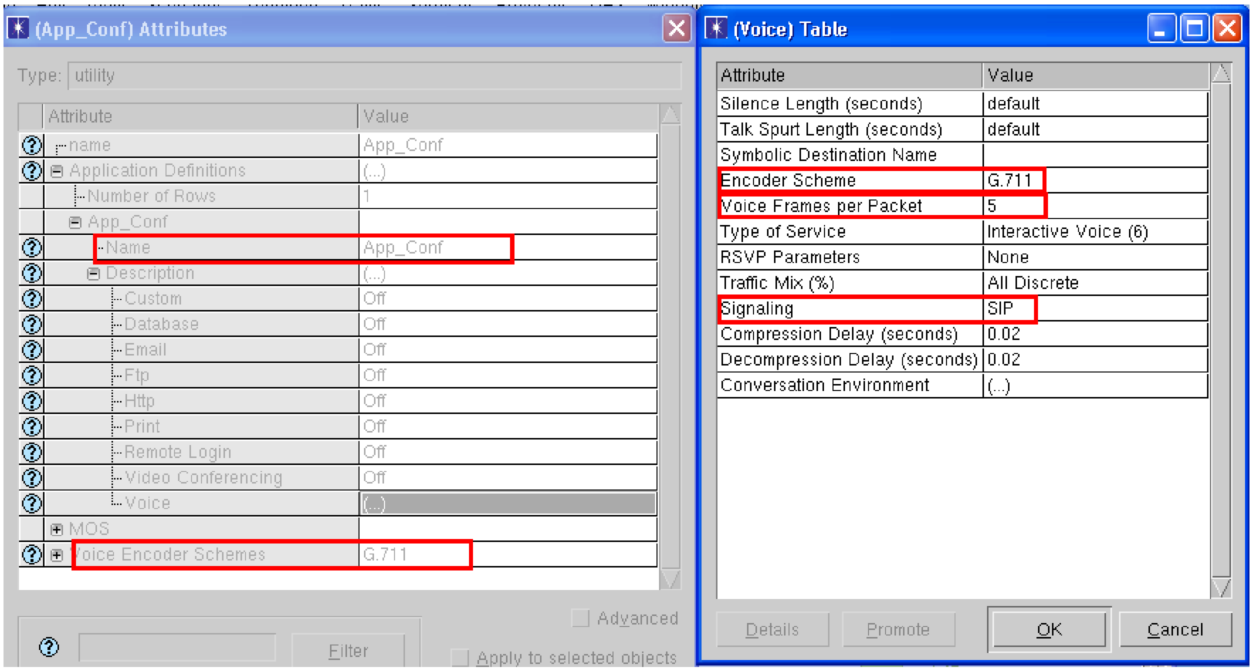


Figure 4-2 VoIP Application Setting

Signaling setting is SIP, this is the most important parameter. Without signalling, no phone call can be established.

Below is Attributes setting of Caller Profile and Proxy Server. **Profile name** is SIP_Prof and can specify more than one kind of application, here under Applications, only one App_Conf was specified.

Operation Mode: defines how applications will start. We only have one Application, so this configure left as default.

Start Time (Seconds): defines when during the simulation the profile session will start. Uniform (100,110) means any time between 100 and 110 seconds. Usually after 100 seconds network connectivity are settled, such as routers and switches are settled done with their routing and switching table.

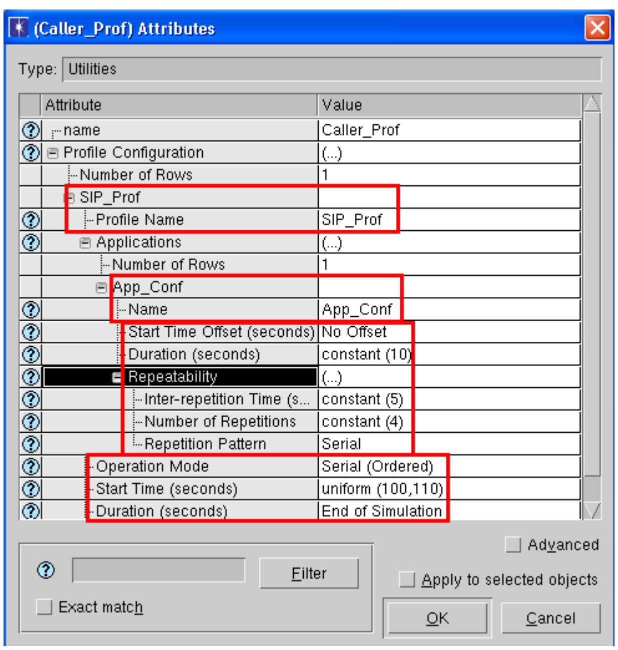


Figure 4-3 Caller Profile

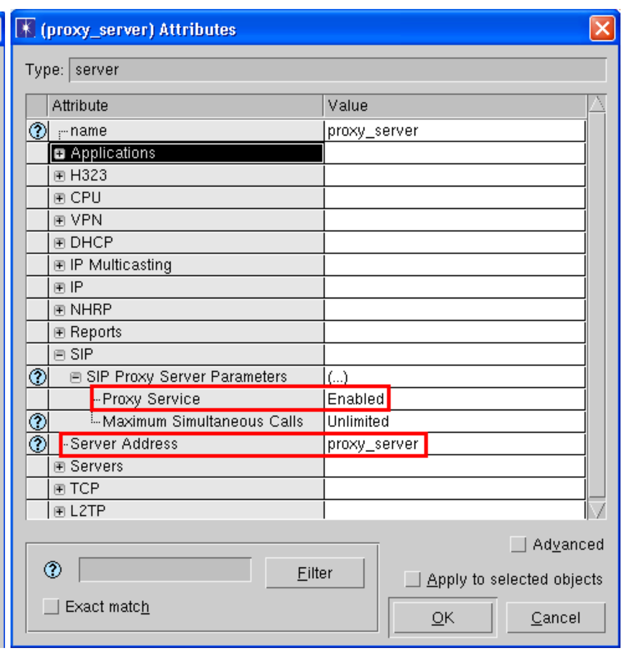


Figure 4-4 Proxy Server

Under the **App_Conf**:

Start Time Offset (Seconds): Defines this application start time related with Profile Start Time.

Duration (Seconds): The maximum amount of time allowed for an application session before it aborts. This is often used as a timeout.

Repetition Pattern: Defines when the next session of the application will start. **Serial** - The start time of the next application is computed by adding the inter-repetition time to the time at which the previous session completed. **Concurrent** - The start time of the next application is computed by adding the inter-repitition time to the time at which the previous session started. When set to concurrent, the mean outcome should not be zero. A mean of zero would case sessions to be created at an infinite rate.

We will talk about Repetition Pattern more detail later.

Simulation result of this configuration is as below:

```
|-------------------------------------------------------------------------------|
| Beginning Simulation.                                                         |
|-------------------------------------------------------------------------------|
my node objid is 7.
my node objid is 12.
UAC (PID 234) has received a request.sim time:101.699300
IN UAC conn_open_acitve:UAC (PID 234) opening an active (tcp) connection to dest (proxy_server) on port (506)
 UAC(PID 234) Intrpt_Preprocess. time 101.699300
 UAC(PID 234) Intrpt_Preprocess. time 101.816111
UAC (PID 234) sending the INVITE request to UAS. sim time: 101.816111.
UAS: Relay request msg to DEST UAC: Voice_Network.Pittsburgh.Callee
UAC (PID 216) is sending invite accept msg to UAS. 102.054294
UAC (PID 216) is processing response for INVITE. 102.054294.
UAC (PID 234) SIP CALL CONNECT SUCCESS! sim time: 102.172422.
UAC (PID 234) is processing response for CONNECT SUCCESS. 102.172422.
UAC (PID 234) has received a request.sim time:112.172422
UAC (PID 234) sending SIP BYE Request to UAS.
UAC (PID 234) is processing BYE Request. 112.172422
UAC (PID 216) is RESPONSE to BYE msg. sim time: 112.290549.
UAC (PID 216) is processing response for BYE. 112.290549.
UAC Disconnect success! Call connect time: 102.172422;Call duration time: 10.236255
UAC (PID 252) has received a request.sim time:116.699300
IN UAC conn_open_acitve:UAC (PID 252) opening an active (tcp) connection to dest (proxy_server) on port (506)
 UAC(PID 252) Intrpt_Preprocess. time 116.699300
 UAC(PID 252) Intrpt_Preprocess. time 116.816111
UAC (PID 252) sending the INVITE request to UAS. sim time: 116.816111.
UAS: Relay request msg to DEST UAC: Voice_Network.Pittsburgh.Callee
UAC (PID 257) is sending invite accept msg to UAS. 117.054294
UAC (PID 257) is processing response for INVITE. 117.054294.
UAC (PID 252) SIP CALL CONNECT SUCCESS! sim time: 117.172422.
UAC (PID 252) is processing response for CONNECT SUCCESS. 117.172422.
UAC (PID 252) has received a request.sim time:127.172422
UAC (PID 252) sending SIP BYE Request to UAS.
UAC (PID 252) is processing BYE Request. 127.172422
UAC (PID 257) is RESPONSE to BYE msg. sim time: 127.290549.
UAC (PID 257) is processing response for BYE. 127.290549.
UAC Disconnect success! Call connect time: 117.172422;Call duration time: 10.236255
UAC (PID 266) has received a request.sim time:131.699300
IN UAC conn_open_acitve:UAC (PID 266) opening an active (tcp) connection to dest (proxy_server) on port (506)
 UAC(PID 266) Intrpt_Preprocess. time 131.699300
 UAC(PID 266) Intrpt_Preprocess. time 131.816111
UAC (PID 266) sending the INVITE request to UAS. sim time: 131.816111.
UAS: Relay request msg to DEST UAC: Voice_Network.Pittsburgh.Callee
```

Figure 4-5 SIP Signalling Follow

Inside the top rectangle, UAC (PID 234) is Caller, received interrupt at 101.699300 seconds, and sent out INVITE request to UAS. UAC (PID 216) is Callee, received Call INVITE and processing this message at 102.172422 seconds. In blue rectangle, Caller received CALL Accept message from UAS and means SIP CALL CONNECT SUCCESS! This simulation is to make a simple SIP call. So there are no packets exchanged between Caller and Callee. We can see UAC of Caller received a interrupt to send BYE message 10 seconds after SIP call established, this is the setting configured in Profile - **Durations**. Callee UAC response to BYE message inside the yellow rectangle. This is a complete SIP call signaling processes. In Raj's report, he added Notify() function inside SIP CALL CONNECT SUCCESS, means after call established successfully, Caller (Notifier) will send emergency alerting messages to Callee (Subscriber).

4.3 Simultaneously SIP call simulation

Simultaneously SIP call from Caller to multiple Callee is the scenario we try to implement. OPNET has limitation on how application instances are created, these application instances using the same application service template.

Currently OPNET do not support instantiation of multiple concurrent applications from a single server to many nodes. In this dissertation, the scenario is that many calls start concurrently from one node to multiple nodes.

The following section discusses the configuration of project, particulars in profile parameter settings.

4.3.1 Profile Settings

A profile describes user activity over a period of time. A profile consists of many different applications. For example, a "Human Resources" user profile may contain "Email", "Web" and "Database".

We can specify various loading characteristics for the different applications on this profile. Each application is described in detail within the application configuration object. The profiles created on this object will be referred by the individual workstations to generate traffic.

There are three types of operation mode of all profiles in the profile – Serial (Random), Serial (Ordered) and Simultaneous. If set all applications to the same application and operation mode set to “Simultaneous”, we are able to achieve the scenario which we want.

Below is a example about Profile “SIP_Prof” which includes 8 applications and all those application set to same VoIP application. This means the nodes which use this profile will start all 8 VoIP applications simultaneously.

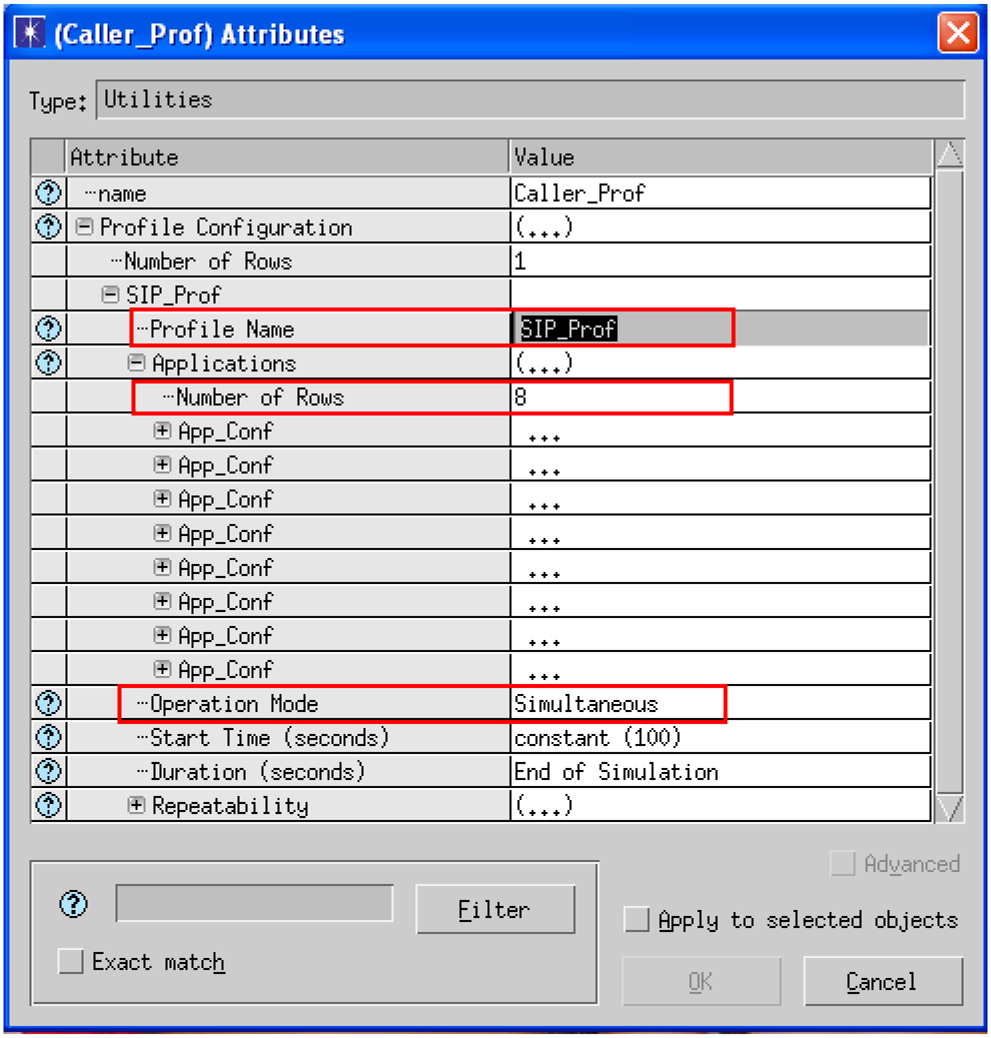


Figure 4-6 Profile Settings

Number of Rows is 8 means there are 8 applications, but the application name all set to “App_Conf”.

In each application, Start Time Offset (seconds) is “No Offset” means will start application the same time this Profile started. Duration (seconds) is “5” means this VoIP call will last for 5 seconds. Repeatability describes this application will repeat a total of 4 times, and new call will start 0.5 second after previous call completed.

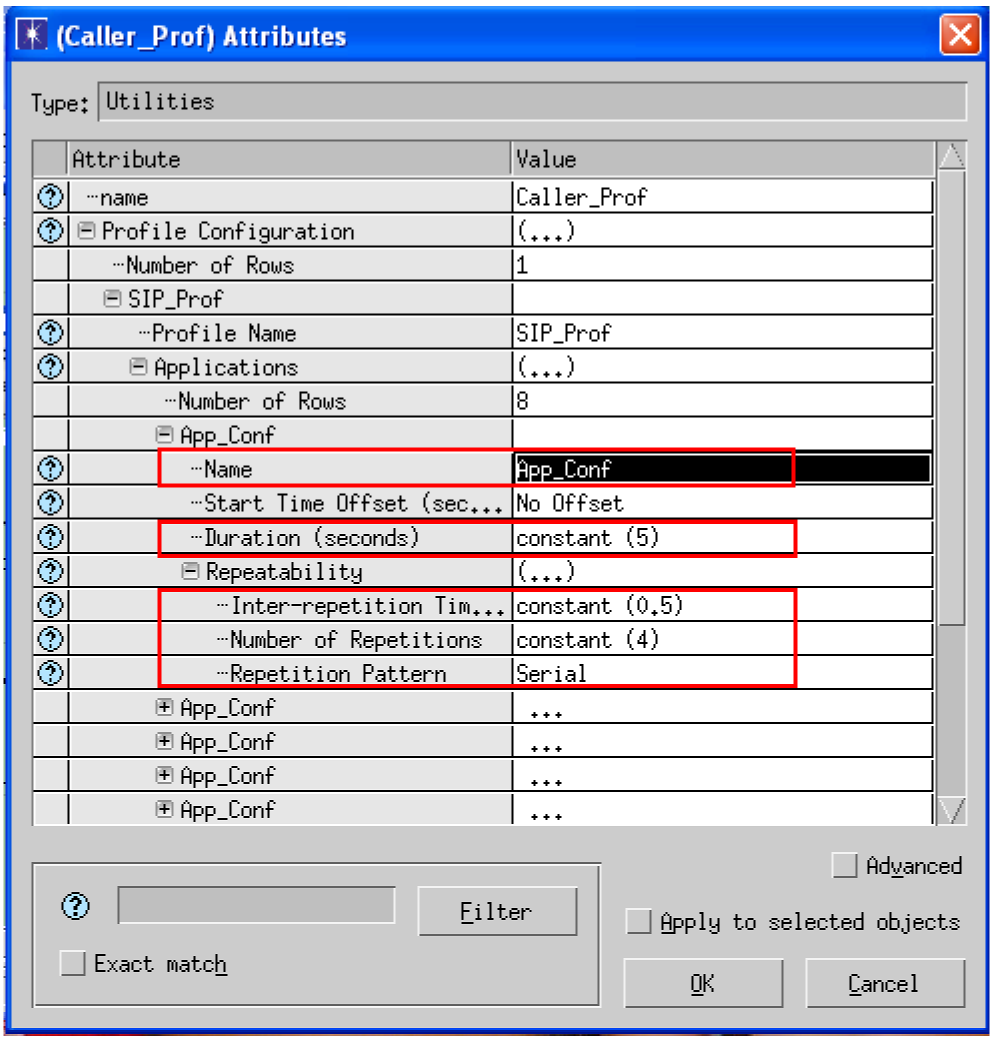


Figure 4-7 SIP Profile Settings

For destination, just place one or more nodes use the same Application's Configuration and SIP_Proxy server address.

Below is this simulation results of only 2 VoIP applications configured in the profile for reducing the length of log:

|--|

| Beginning Simulation. |

|--|

UAC (PID 258) has received a request.sim time:100.000000 /*UAC258 has received request from UAC manager*/

IN UAC conn_open_acitve:UAC (PID 258) opening an active (tcp) connection to dest (proxy_server) on port (506)

UAC (PID 288) has received a request.sim time:100.000000

IN UAC conn_open_acitve:UAC (PID 288) opening an active (tcp) connection to dest (proxy_server) on port (506)

UAC(PID 258) Intrpt_Preprocess. time 100.000000 /*UAC258 starts process interrupt */

UAC(PID 288) Intrpt_Preprocess. time 100.000000

UAC(PID 258) Intrpt_Preprocess. time 100.116811 /*UAC258 starts process interrupt */

UAC (PID 258) sending the INVITE request to UAS. sim time: 100.116811. /*UAC258 initiate SIP call to UAS */

UAC(PID 288) Intrpt_Preprocess. time 100.117075

UAC (PID 288) sending the INVITE request to UAS. sim time: 100.117075.

UAS: Relay request msg to DEST UAC: Voice_Network.Pittsburgh.Callee_0 /*UAS forward UAC258's invite to Callee_0 */

UAS: Relay request msg to DEST UAC: Voice_Network.Pittsburgh.Callee

UAC (PID 268) is sending invite accept msg to UAS. 100.354994 /*UAC268 accept the invite from UAC258 */

UAC (PID 268) is processing response for INVITE. 100.354994.

UAC (PID 239) is sending invite accept msg to UAS. 100.355487

UAC (PID 239) is processing response for INVITE. 100.355487.

UAC (PID 258) SIP CALL CONNECT SUCCESS! sim time: 100.473122.

UAC (PID 258) is processing response for CONNECT SUCCESS. 100.473122.

UAC (PID 288) SIP CALL CONNECT SUCCESS! sim time: 100.473614.

UAC (PID 288) is processing response for CONNECT SUCCESS. 100.473614.

UAS received the Notify. sim time 100.531496. /*UAS has received the Notify msg from UAS 258*/

UAS received the Notify. sim time 100.531988.

UAC (PID 258) has received a request.sim time:105.473122 /*UAC258 request terminate the call after 5 seconds*/

UAC (PID 258) sending SIP BYE Request to UAS. /*UAC258 request terminate the call after 5 seconds*/

UAC (PID 258) is processing BYE Request. 105.473122 /*UAC258 send BYE msg to destination UAC*/

UAC (PID 288) has received a request.sim time:105.473614

UAC (PID 288) sending SIP BYE Request to UAS.

UAC (PID 288) is processing BYE Request. 105.473614

UAC (PID 291) has received a request.sim time:105.500000

IN UAC conn_open_acitve:UAC (PID 291) opening an active (tcp) connection to dest (proxy_server) on port (506)

UAC(PID 291) Intrpt_Preprocess. time 105.500000

UAC (PID 268) is RESPONSE to BYE msg. sim time: 105.591250. /*UAC268 send BYE msg to UAC258*/

UAC (PID 268) is processing response for BYE. 105.591250. /*UAC268 send BYE msg to UAC258*/

UAC (PID 239) is RESPONSE to BYE msg. sim time: 105.591742.

UAC (PID 239) is processing response for BYE. 105.591742.

UAC(PID 291) Intrpt_Preprocess. time 105.616811

UAC (PID 291) sending the INVITE request to UAS. sim time: 105.616811.

UAS: Relay request msg to DEST UAC: Voice_Network.Pittsburgh.Callee

UAC Disconnect success! Call connect time: 100.473122;Call duration time: 5.236255 /*UAC258 Closed*/

UAC Disconnect success! Call connect time: 100.473614;Call duration time: 5.236255

UAC 258 AND UAC 288 start SIP call process at the same time exactly. Due to time delay of the network, there is tiny difference of each activity between two calls. We could initiate many SIP call simultaneously through adding many applications to profile setting and using the same application definition. In the following section, more results of simulation will be provided.

The statistic results of 8 simultaneous SIP call is as below. Each of 8 calls repeated 4 times respectively.

1. Call set up time always around 0.5 second.
2. Voice packet end-to-end delay around 0.2 second.

3. Peak backhaul utilization is around 15kbps. This is far less than 1.544Mpbs bandwidth.

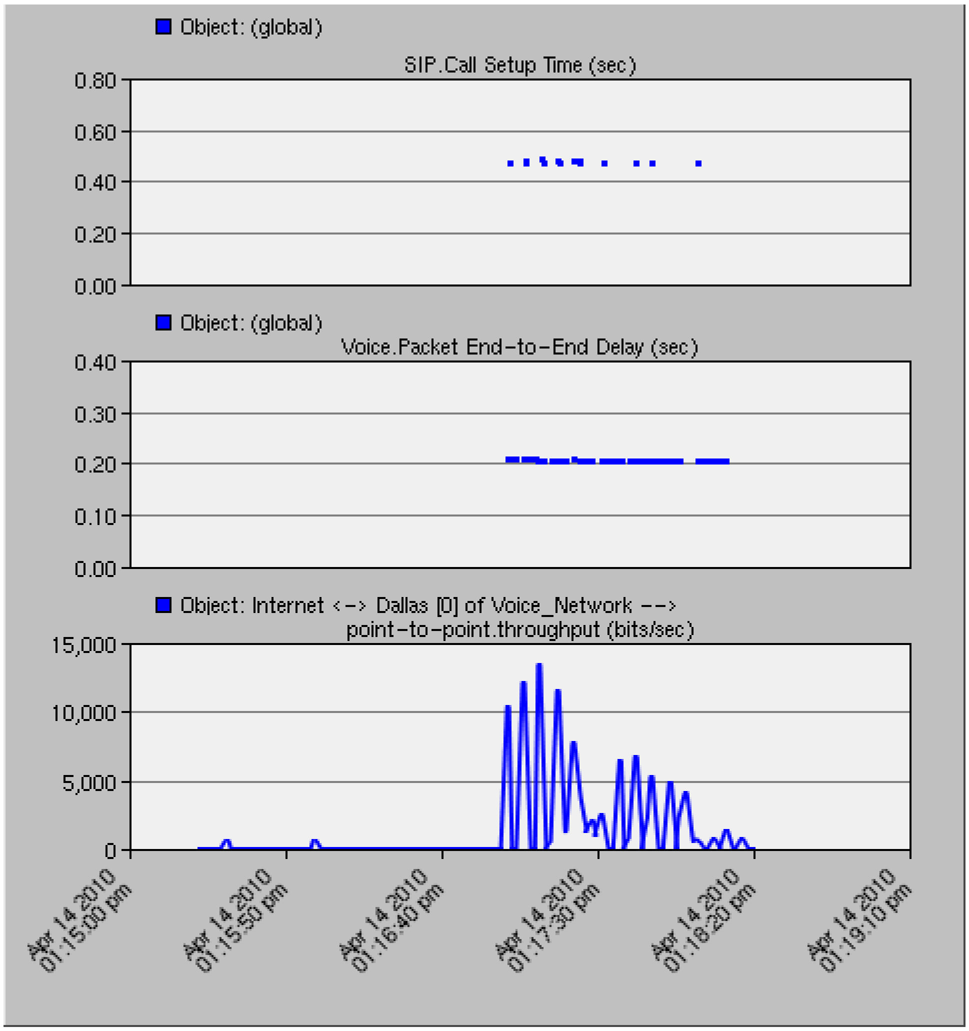
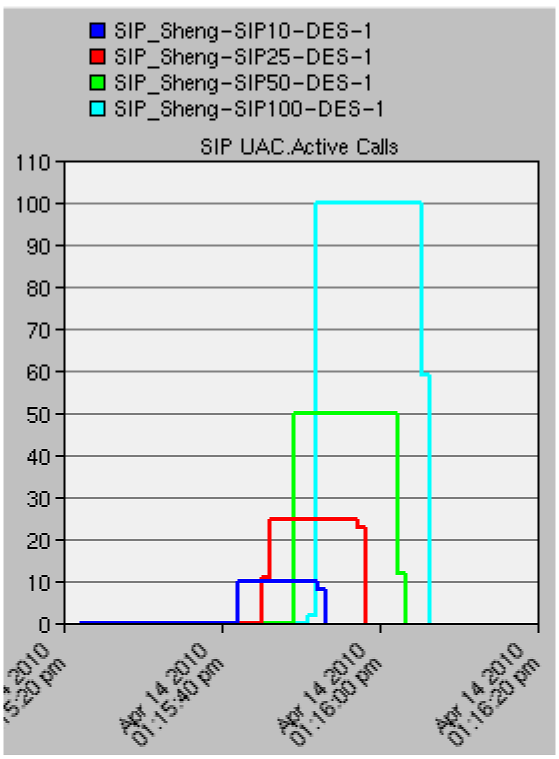


Figure 4-8 8 Simultaneous SIP Calls

4.3.2 Increase the number of simultaneously call

In this section, we will implement 10, 25, 50 and 100 SIP calls simultaneously initiated from caller. The analysis of result also provided. For a better comparison, all calls duration are set to 10 seconds and only simulate once. The start time of each scenario are set to 20, 23, 26 and 29 seconds after simulation started.



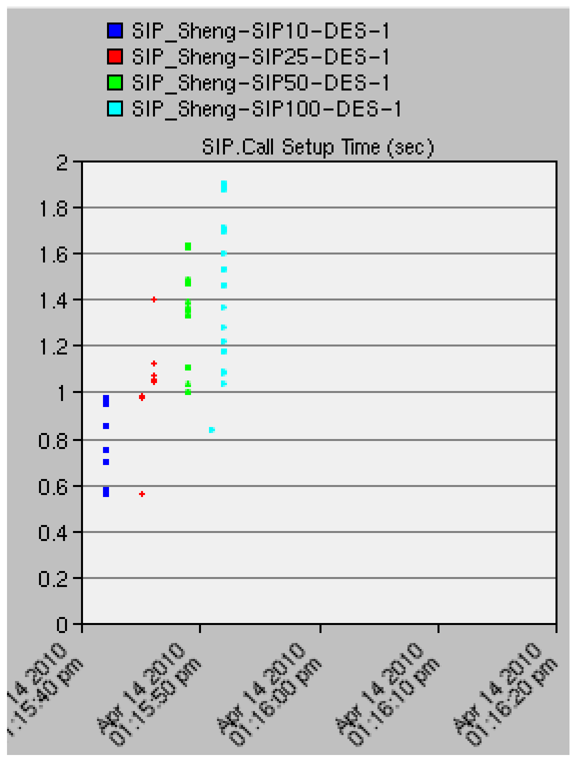
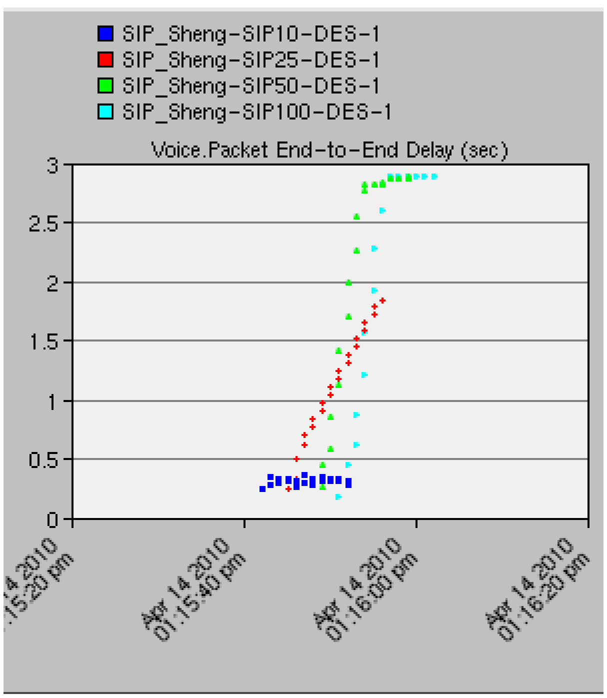


Figure 4-9 Active Calls **Figure 4-10** Call Setup Time

It shows how many calls established simultaneously in the Figure. On the right ones, SIP call setup time increased with the number of call increased. Maximum call setup time delay almost reached 2 seconds when 100 calls start simultaenoulsy, it is almost two times of 10 simultaneous SIP calls delay. At this stage most of traffic on the network is SIP traffic and it won't use out all backhaul bandwidth, though this delay is mostly due to UAC processing time delay.



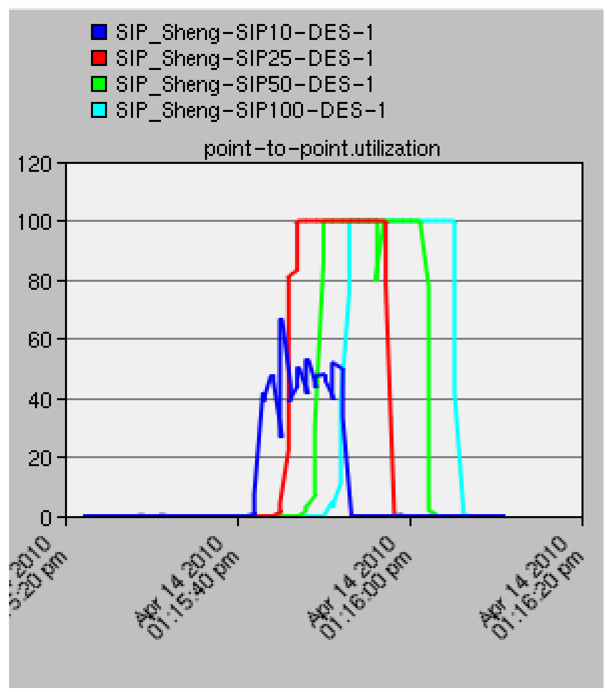
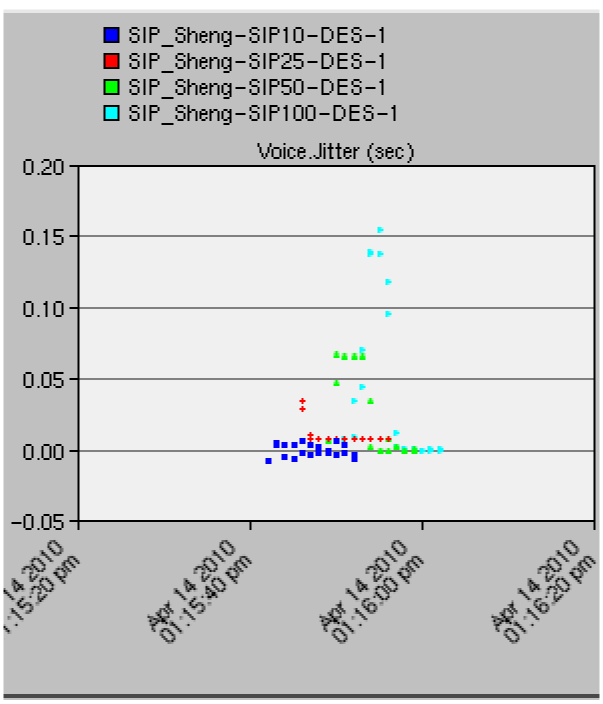


Figure 4-11 Voice Packet Delay **Figure 4-12 Utilization**

Above statistic graphics are about the Voice Packet delay and Voice Jitter. Utilization is the link usage of point-to-point link from

Internet to San_Francisco. Except SIP10 scenario, all other scenarios are reached the maximum bandwidth – 1.544mbps. There is very small delay of SIP10, and for SIP25, the delay and time is linear relation until simulation completed. SIP50 and SIP100’s packet delay are almost the same. Usually for SIP call, maximum packet delay should less than 150 milliseconds. In this simulation, the coding scheme for voice is G.711, it will tak64kbps per call, so 1 DS1 (1.544mbps) can only support 24 calls simultaneously.



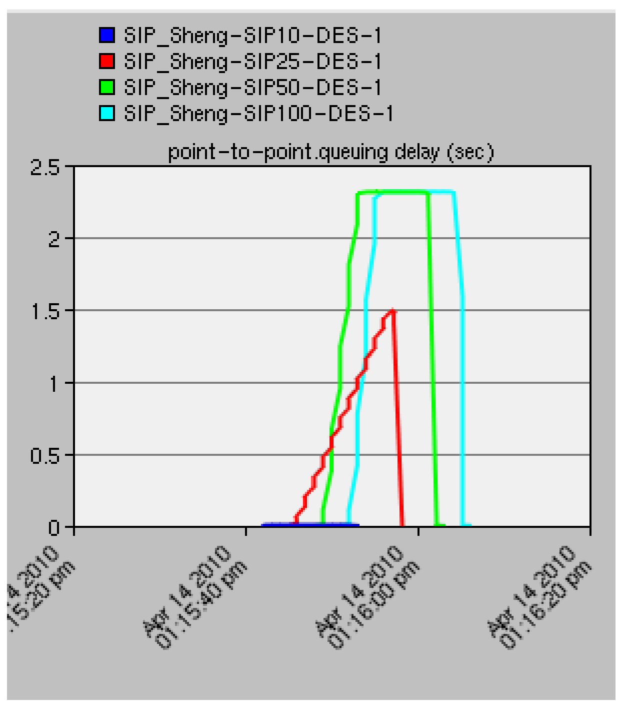


Figure 4-13 Voice Jitter **Figure 4-14 Queuing Delay**

Voice jitter is a very important KPI for the quality of VoIP calling. All calling parties have great call quality if jitter less than 60 milliseconds. SIP10 and SIP25 are good at less than 50 ms level. The Queuing Delay is for point-to-point link of San_Francisco to Internet.

5. Future Work

When any emergency happens, all subscribers have subscribed that event need to be notified as quickly as possible, sever or agent initiates SIP session simultaneously, the number of parallel call is determined by the capacity and how many phone lines are available at that time. This report presents a method of how to simulate 100 SIP calls simultaneously using OPNET and provides comparison results of 4 scenarios. Based on this report, CAP message can be integrated into notification process in the future work.

6. References

1. http://www.incident.com/cap/what-why-how.html
2. www.ic.gc.ca/eic/site/...CANALERT.../2-Botterell_CANALERT.pps
3. http://www.cs.columbia.edu/~knarig/911.pdf
4. www.cs.columbia.edu/.../SIPBasedEmergencyNotificationSystemPres..
5. RFC 3265: Session Initiation Protocol (SIP) – Specific Event Notification by A.B. Roach, June 2002
6. RFC 3903: Session Initiation Protocol (SIP) Extension for Event State Publication by A.Niemi, October 2004
7. RFC 3261: Session Initiation Protocol by J. Rosenberg, June 2002
8. OASIS Standard CAP-V1.1, October 2005
9. http://en.wikipedia.org/wiki/Common_Alerting_Protocol
10. Latency and QoS for Voice over IP http://www.sans.org/reading_room/whitepapers/voip/latency-qos-voice-ip_1349
11. An OPNET-based Simulation Approach for Deploying VoIP by K. Salah
12. Practical Considerations for Extending Network Layer Models with OPNET
13. Performance Studies of VoIP over Ethernet LANs by Di Wu, 2008

14. Common Alerting Protocol (CAP) over Session Initiation Protocol (SIP) by

Raj Gondara, 2011