Bandwidth Allocation and Session Scheduling using SIP

Hassan HASSAN, Jean-Marie GARCIA and Olivier BRUN
LAAS-CNRS, Toulouse, France
Email: {hhassan}@laas.fr

***Abstract*— Session Initiation Protocol (SIP) is a new signaling protocol designed to establish multimedia sessions in telecommunication networks. In this paper, we suggest the extension of SIP functionalities to coordinate QoS mechanisms deployed in IP networks, and especially in DiffServ domain. Indeed, the interaction between small and big TCP sessions may have dramatic consequences on small TCP sessions. Hence, we use SIP to achieve QoS management on a session basis, in which the over all activity of the user during the session is considered. The suggested mechanisms deal with two issues: first, session scheduling based on session duration and/or volume, and second bandwidth allocation on a per-flow basis using equivalent bandwidth estimation techniques. The proposed mechanisms are implemented in the SIP proxy server as QoS management algorithms, and they are validated by simulations.**

***Index Terms*—SIP, DiffServ, Web, QoS, Bandwidth allocation, Scheduling.**

I. INTRODUCTION

The evolution of telecommunication networks made of the Internet a universal platform to support most forms of modern communications including voice, video and data applications. However, the Internet Protocol (IP) was developed based on a connectionless model where simple metrics (e.g. delay or hope count) are used to achieve routing. The simple concept of IP is behind its success and its ability to scale to very large networks. Unfortunately, no Quality of Service (QoS) was planned with this approach. Over years, many enhancements to QoS support were implemented in IP packet networks. Hence, two QoS architectures DiffServ and IntServ were introduced to carry out application QoS requirements. Moreover, new protocols such as MPLS were conceived to extend the best-effort service of IP networks. Besides, the convergence to all-IP network came with new signaling protocols to handle user sessions in all access networks regardless their particularities. Thus, the Session Initiation Protocol (SIP) was proposed as a signaling protocol to establish and release sessions between end users.

SIP is very general and can be used for any kind of sessions in all communication networks. Moreover in the year 2000, SIP was selected by the Third Generation Partnership Project (3GPP) as the call control protocol for the 3G IP-based mobile networks. On the other hand, the successful deployment of Multi-Label Switching Protocol MPLS in DiffServ networks delegates label switching as an apt switching technology for the future core networks especially for its Traffic Engineering (TE) capabilities. Indeed, combining both technologies introduces a new vision of QoS management at the application level. The autonomous structure of SIP makes it possible to manage user sessions as classic phone calls, from the beginning to the end. However, SIP can achieve much more than signaling the beginning and the end of a communication session. In particular, it can host traffic engineering intelligence. Thus, the use of SIP over DiffServ networks allows flexible QoS management as it combines DiffServ facilities with SIP supervision.

In this paper, we propose a QoS management framework based on SIP over DiffServ environment. Where QoS management mechanisms are implemented and supervised by the SIP proxy server. The proposed mechanisms concerns: first, session scheduling based on session duration and/or session data volume exchanged during a session and second, bandwidth allocation on a per-flow basis using equivalent bandwidth estimation techniques. The paper is organized as follows: first, we give a brief overview of SIP and its associated Session Description Protocol (SDP), and then we explain the suggested SIP over DiffServ architecture. Second, we present equivalent bandwidth estimation techniques used for bandwidth allocation. Finally, we present the session based QoS algorithms implemented along with simulation results.

II. SESSION INITIATION PROTOCOL

Session Initiation Protocol is an application layer control protocol designed and developed by the IETF [14]. The specification is available in form of several RFCs, the most important one is RFC3261 [15] which contains the core protocol specification. The easy implementation, flexibility and good scalability are the main motivations considered while designing this protocol.

The main task of the protocol is to set up and release sessions between end users. The session refers to the activity between sender and receiver when the whole state is maintained during the communication. Classic sessions include Internet telephone calls, but it may also be multimedia conference session, Web session, distributed computer game session, etc.

The communication itself between devices is achieved by other protocols (often RTP, RTCP and SDP) as the purpose of SIP is to initiate communications only. Real Time Protocol RTP carries the real-time application data (including audio and video) by splitting and encoding data into packets to allow per-packet transport on the Internet, and SDP describes and encodes capabilities of sessions. Indeed, the characteristics of the session are negotiated between participants. The negotiation includes the type of codecs used to encode media in order to facilitate decoding process, maximum allowed bit rates, the transport protocol, etc.

The end-to-end model of SIP complies with the Internet architecture. Indeed, all the intelligence is stored in end devices, including state. This protects from single point failure while preserving scalability in networks. In contrast with Public Switched Telephone Network (PSTN) where state and intelligence are stored in the network while terminals are dump. However, SIP can provide the same functionality as PSTN with the possibility to implement end-to-end services that are hardly configured in PSTN.

Finally, it is clear that the scalability and decentralization of SIP come at the cost of end-to-end message overhead. In fact, SIP is based on HTTP protocol which is widely used on the Web. Actually, HTTP can be seen as a signaling protocol also, as web browsers tell HTTP servers about the documents they need. The encoding of message headers in both protocols (HTTP and SIP) have been inherited from RFC822 [16]. This encoding has already showed robustness and flexibility with HTTP.

The physical elements of a SIP network fall into two categories: clients and servers. Fig. 1, illustrates the architecture of a SIP network.

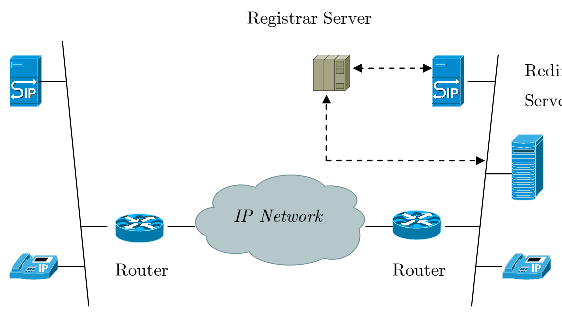

Figure 1. SIP Architecture.

Client in SIP is a general concept. It could be any device initiating sessions (Phones, PCs, Palms, …). On the other hand, SIP servers include the following main types:

- Proxy server: The most important element in the SIP architecture, as it constitutes an intermediate device receiving SIP requests from clients and forwarding the requests on clients' behalf. Typically, proxy servers forward SIP messages to other SIP servers in the network. Proxy server can play several roles. Besides, it provide functions such as authentication, authorization, network access control, routing, reliable request retransmission, and security.
- Redirect server: It takes care of directing the client to the next hop until the client reaches the destination server and contacts UAS directly.
- Registrar server: It handles client registration request for its current location. Generally, Registrar server is co-located in the same physical entity hosting the redirect or proxy server.

SIP works in tandem with the Session Description Protocol (SDP) that describes multimedia sessions. Session description serves for session announcement, session invitation and other session initiation functionalities. SDP is completely independent of transport protocol. It concerns mainly the format of session description and is designed to work with any transport protocol.

Many of the SDP messages are sent using Session Announcement Protocol (SAP). These messages are UDP packets with a SAP header and a text payload. The text payload is the SDP session description. Messages can also be sent using email or the World Wide Web.
Fig. 2, depicts the position of SIP and SDP in the multimedia protocol stack.

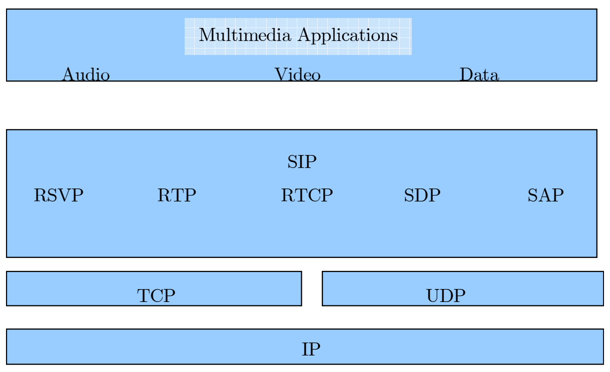

Figure 2. Multimedia protocol stack.

III. SIP OVER DIFFSERV

A. Background

Few researches [11], [12] and [13] have addressed the architecture of SIP over DiffServ architecture in IP networks. Zhang and Guy [13] proposed an extension to the Proxy server in the SIP architecture to include Traffic Engineering (TE) capabilities, where the SIP proxy server uses the messages exchanged during an SIP session to provide TE requests.

The proxy server may use the messages exchanged during an SIP session to exchange traffic engineering requests. These requests will be exchanged between the SIP proxy server and the Label Edge Router (LER) by Common Open Policy Service (COPS) protocol messages

[12]. Indeed, we need to transfer information related to the request of resource by QoS clients and for the allocation of resources by resource allocation servers (e.g., bandwidth broker) in a DiffServ network. Hence, this resource allocation functionality can be added in the COPS framework. Fig. 3, depicts the proposed architecture for SIP over DiffServ.

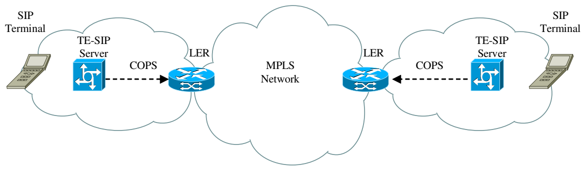

Figure 3. SIP over DiffServ architecture.

The SIP proxy server could only negotiate TE sessions with another TE enabled SIP proxy server, otherwise normal SIP session (without TE extensions) is initiated. The flow of SIP messages is resumed on Fig. 4.

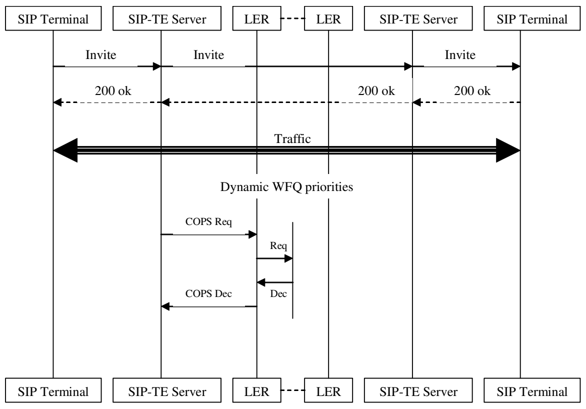

Figure 4. SIP over DiffServ flow of messages.

Assuming that communication issues related to resources management and reservation at the LER is achieved by the COPS framework; SIP can play the role of an application control layer protocol to guarantee QoS requirements of user sessions.

B. Session based QoS

Generally, we speak of QoS per type of service. Thus, for real time applications we are concerned about end to end time constraints like delay and jitter to guarantee the reconstitution of multimedia signals (voice or video) [17]. On the other hand, non-real time applications (or data applications) are less sensitive to time constraints, while they require guaranteed nominal bit rates and loss free transmission. Hence, these applications (e.g Web, FTP and email) use Transmission Control Protocol (TCP) for reliable and in-order transmission. However, the elastic behavior of TCP makes the transmission completely dependent of the network congestion state.

In both cases, real time and non-real time applications, the QoS requirements are grouped together to express an SLA required for one application. This could be considered as a microscopic view of QoS. On the other hand, a macroscopic view of QoS may include user behavior during session. The global view per session is important in our context because SIP handles the session establishment and considers the user activity during a session as a whole. The extension of SIP to offer QoS provisioning mechanisms requires a macroscopic treatment for sessions. Consequently, we define a session based QoS in which we consider user activity during the session and the type of the application used.

Based on the above presented architecture we want to implement QoS management algorithms at the session level. The proposed algorithms are based on traffic engineering techniques and will be hosted in the SIP proxy server. Thus, the SIP proxy server implements QoS mechanisms on multimedia sessions, based on measurements and a priori estimation of equivalent bandwidth. Indeed, the SIP proxy server measures the duration of sessions and the data volume exchanged during sessions (functions that are generally used for billing purposes), then a session scheduling can be achieved. Furthermore, the SIP proxy server can perform flow based equivalent bandwidth estimation, based on information collected about session parameters before initiating it. However, this requires equivalent bandwidth estimation techniques that we will present in the following section.

IV. EQUIVALENT BANDWIDTH

A. Related Work

Traffic control is generally used to optimize the allocation of network resources in order to sustain an acceptable QoS for network connections. Many traffic control strategies including congestion control and call admission policies, rely on the notion of the equivalent (or effective) bandwidth or the resulting connection load on network links. For example, access control uses this information to decide whether to accept or not incoming requests for new connections. The admission criteria depend on the impact of new added connections on both the resource utilization and the QoS offered for accepted and already existing traffic.

The notion of equivalent bandwidth has been used in the literature and two broad categories of equivalent bandwidth estimation approaches are generally used. The first category is based on Kelly's [7] mathematical definition of equivalent bandwidth for different kinds of traffic. The second category refers to the analytical methods based on traditional queuing theory. The mathematical framework proposed by Kelly relies on large deviation theory to estimate the equivalent bandwidth of a stationary arrival process. On the other hand, analytical approaches hypothesize the traffic models in order to give an approximate expression for the equivalent bandwidths in some cases such as Markov processes (e.g. [2,10]). In this paper, we suggest an analytical approach to estimate equivalent bandwidth based on a renewal process approximation.

B. Equivalent bandwidth estimation by renewal process approach

In order to characterize the effective bit rate or equivalent bandwidth of a traffic source, we need to select an appropriate model to specify its characteristics in terms of known parameters or metrics. For the purposes of our research, we adopt a two-state model ($ON\text{-}OFF$) that captures the basic behavior of the data source associated with a connection. The rationale for such a model is that a source is either in an "idle state," (OFF) transmitting at zero bit rate, or in a "burst state" (ON) and transmitting at its peak rate. Such a source model has the advantage of being both simple and flexible as it can be used to either represent connections ranging from bursty to continuous bit streams. Let the following values be associated with one connection:

R_M : Average rate of one connection (Kbps),

T_{ON} : Average duration of ON period (Sec),

T_{OFF} : Average duration of OFF period (Sec),

R_{ON} : Average rate in ON period (Kbps),

Q_{ON} : Average file size in ON period (Kb) (to be used only with TCP based models),

X : Buffer size,

E : Target packet loss probability.

The renewal process approach is an approximation of the superposition of $ON\text{-}OFF$ processes. In this method, we study the superposition of N $ON\text{-}OFF$ processes as a $GI/D/1/K$ system. In order to evaluate the equivalent bandwidth of N $ON\text{-}OFF$ processes we study the packet loss in $GI/D/1/K$ system. The packet loss probability is calculated as a function of the following parameters:

K : The buffer size in packets,

ρ: System load with $\rho = \lambda/\mu$, λ is the aggregate arrival rate of input $ON\text{-}OFF$ processes and μ is the average service time.

c_a^2: The squared coefficient of variation of the input arrival process.

c_s^2: The squared coefficient of variation of the service time.

Our approach is based on packet loss approximation formulas for the $GI/D/1/\infty$ queue presented in [19,20]. Let the number of clients in the queue including the one being serviced, be denoted by M. Whitt [19] expresses the average and the second moment of the number of clients M as:

$$\begin{aligned} E(M) &= \rho + \frac{\rho^2(c_a^2 + c_s^2)g}{2(1-\rho)} \\ E(M^2) &= E^2(M)(c_M^2 + 1) \end{aligned} \tag{1}$$

g is a weighing factor depending on the value of c_a^2:

$$g = \begin{cases} e^{\left[-\frac{2(1-\rho)(1-c_a^2)^2}{3\rho(c_a^2+c_s^2)}\right]} & c_a^2 < 1 \\ 1 & c_a^2 \geq 1 \end{cases} \tag{2}$$

The parameter c_N^2 is defined as:

$$c_M^2 = \frac{Y_1 Y_2}{Y_3} \tag{3}$$

Y_1 is the value of $Var(M)$ given by:

$$Var(M) = \lambda E(W) + \rho + \rho^2 c_s^2 + \lambda^2 Var(W) \tag{4}$$

W denotes the steady state waiting time before beginning service. The average and the variance of W are given by:

$$\begin{aligned} E(W) &= \frac{\tau\rho(c_a^2 + c_s^2)g}{2(1-\rho)} \\ Var(W) &= E^2(W)c_w^2 \end{aligned} \tag{5}$$

τ is the mean service time and c_w^2 is the squared coefficient of variation of the waiting time expressed as:

$$c_w^2 = \frac{c_D^2 + 1 - \sigma}{\sigma} \tag{6}$$

σ is the delay probability whose value is:

$$\sigma = P(W > 0) = \rho + (c_a^2 - 1)\rho(1-\rho)h \tag{7}$$

$$h = \begin{cases} \dfrac{1 + c_a^2 + \rho c_s^2}{1 + \rho(c_s^2 - 1) + \rho^2(4c_a^2 + c_s^2)} & c_a^2 \leq 1 \\ \dfrac{4\rho}{c_a^2 + \rho^2(4c_a^2 + c_s^2)} & c_a^2 \geq 1 \end{cases} \tag{8}$$

c_D^2 is the squared coefficient of variation of the conditional delay given that the server is busy. Its value when service time is deterministic is given by:

$$c_D^2 = \frac{2\rho - 1 + 4(1-\rho)}{3(c_s^2 + 1)^2} \tag{9}$$

Finally Y_2 and Y_3 are given by:

$$\begin{aligned} Y_2 &= \frac{1 - \rho + \sigma}{\max\{1 - \sigma + \rho, 0.000001\}} \\ Y_2 &= \max\{\rho + \lambda E(W), 0.000001\} \end{aligned} \tag{10}$$

The maximum is used to avoid division by zero. In order to compute the packet loss when finite buffer is considered the two first moments of the packet loss distribution are not sufficient. The distribution itself is needed, which can be obtained by a continuous distribution fit as shown in [19]. Thus $\Pr(M > x)$ is expressed as a function of c_M^2 value as follows:

Case $c_M^2 > 1.01$

$$\Pr(M > x) = p(e^{-\gamma_1 x} - e^{-\gamma_2 x})$$

$$\text{where } p = \frac{\left(1 + \sqrt{\dfrac{c_M^2 - 1}{c_M^2 + 1}}\right)}{2}$$

$$\text{And } \gamma_1 = \frac{2p}{E(M)}, \gamma_2 = \frac{2(1-p)}{E(M)}$$

Case $0.99 < c_M^2 < 1.01$

$$\Pr(M > x) = e^{-\frac{x}{E(M)}}$$

Case $0.501 < c_M^2 < 0.99$

$$\Pr(M > x) = \frac{(\gamma_1 e^{-\gamma_2 x} - \gamma_2 e^{-\gamma_1 x})}{(\gamma_1 - \gamma_2)}$$

where $\gamma_1 = \dfrac{\gamma_2}{\gamma_2 E(M) - 1}$

And $\gamma_2 = \dfrac{2}{E(M) + \sqrt{2Var(M) - E^2(M)}}$

Case $c_M^2 < 0.501$

$$\Pr(M > x) = e^{-\gamma x}(1 + \gamma x)$$

Where $\gamma = \dfrac{2}{E(M)}$

Note that the suggested heuristic estimates the packet loss probability as a function of buffer size, average input rate, average output rate and the squared coefficient of variation for both arrival and service processes (c_a^2 and c_s^2). Specifically, the last two parameters play an important role in estimating the equivalent bandwidth of input traffic.

C. Erlang blocking probability

The equivalent bandwidth estimated by the previous technique considers a constant number N of ON-OFF connections. In the general case we model the flow arrival process at the call level as Poisson process. Each flow is defined by its call arrival rate λ, ON period average rate R_{ON}, mean rate R_M and average ON duration T_{ON}. Thus, for N connections with $R_{M,i}$ equivalent bandwidth, $i = 1,...,N$. The overall equivalent bandwidth BW_{eq} is given by:

$$BW_{eq} = \sum_{i=1}^{N} R_{M,i} \tag{11}$$

In the last equation we consider N constant sessions. When Poisson arrival process is considered, one can estimate the equivalent number of connections (circuits) for one blocking probability, using Erlang B formula:

$$P(N) = \frac{A^N / N!}{\sum_{i=0}^{N} A^i / i!} \tag{12}$$

With $A = \lambda / \mu$.

Hence for one flow with session Poisson arrivals, we estimate the equivalent number of sessions N for one blocking probability B_p and the equivalent bandwidth BW_{eq} is obtained easily by multiplying the equivalent rate of one connection by N.

D. Equivalent bandwidth of multimedia flows

Two types of multimedia flows are considered: VoIP, and Data. VoIP and Data sessions are usually modeled as ON-OFF processes. However, packet transmission is achieved by UDP for VoIP, and by TCP for Data sessions. Recall that TCP algorithm reacts to packet losses. Hence, the equivalent bandwidth of TCP flows can only be achieved under the hypothesis of loss free transmission. However, it is sufficient in our case as the goal is to achieve loss free transmission.

VoIP applications have a common characteristic which is the constant packet size and constant packet inter-arrival time during ON periods (see [25] for more details). The squared coefficient of variation of service time process in a deterministic service queue for VoIP packets is null ($c_s^2 = 0$) because packet sizes are constant, while the squared coefficient of variation of packet arrival process for N ON-OFF connections is given by [24]:

$$c_a^2 = wc_1^2 + 1 - w \tag{13}$$

c_1^2 is the squared coefficient of variation of single ON-OFF connection and it is calculated as function of the packet transmission probability p, constant packet inter-arrival T, T_{OFF} duration:

$$c_1^2 = \frac{1 - p^2}{(T / T_{OFF} + 1 - p)^2} \tag{14}$$

$$w = \frac{1}{1 + 4(1 - \rho)^2 (N - 1)} \tag{15}$$

The average rate is $R_M = P_{ON} * R_{ON}$ and the maximum rate is $R_{ON} = \dfrac{1}{T}$.

Although Web sessions share the same ON-OFF structure with VoIP applications; the packet arrival process is more complicated as it depends on the TCP algorithm. By consequence, the squared coefficient of variation of packet arrival process can not be estimated analytically. Two solutions to this problem may be proposed: first, we can measure the value of c_a^2 directly on the generated trace. This requires having the generated traffic before evaluating the equivalent bandwidth, which may not be useful when used in a QoS management server (the SIP proxy server). That is why we suggest a second heuristic based on the approximation of packet arrival process during ON periods by a constant process of the same average. Thus, we need to estimate the average rate during the ON period when only the file size in known.

Authors in [21] present a formula to calculate the transfer time when TCP is used on short-lived connections (one ACK per two packets b=2).

$$T(Nb) = RTT \begin{pmatrix} \log_{1.57}(Nb) \\ +(f(p, RTT)Nb \\ +4p\log_{1.57}(Nb) + 20p) \end{pmatrix} + \frac{(10 + 3RTT)Nb}{4(1 - p)W_{\max}\sqrt{W_{\max}}} \tag{16}$$

Assuming that:

Nb is the file size in packets,

$W_{\max}$ The maximum reception window,

p packet loss probability,

And

$$f(p, RTT) = \frac{2.32(2p + 4p^2 + 16p^3)}{(1 + RTT)^3} + \frac{1 + p}{10^3 RTT}$$

As we are concerned with the ON periods of Web sessions, the file transfer activity is very short compared to the idle period. Thus, this formula is appropriate to our study case. Meanwhile, a major simplification can be done when estimating the equivalent bandwidth with small loss probabilities. Indeed, the contribution of the first term $RTT \log_{1.57}(Nb)$ is dominant, and the equation can be used in its simpler form:

$$T(Nb) = RTT \log_{1.57}(Nb) \qquad (17)$$

Using the estimated transmission time during the ON period we can estimate the average transmission rate as $\lambda_{ON} = \frac{Nb}{T(Nb)}$ with $Nb = \frac{Q_{ON}}{Ps}$.

Having the average transmission rate λ_{ON} we can use the same formulas as for VoIP while considering constant packet inter-arrivals during the ON period (this is an approximation).

On the other hand, we consider a maximum segment size in the TCP algorithm of 984 bytes. Thus, we generate packets of (MSS+Header= 984+40 =1024 Bytes). As a result we generate 1024 bytes for all packets except the last one (the residual value). Consequently, the squared coefficient of variation of service time in a deterministic service queue for Web packets can be supposed null ($c_s^2 \approx 0$). As a result, the equivalent bandwidth estimation procedure is very similar to the VoIP case.

E. *Performance validation*

We validate the estimated equivalent bandwidth values for G711 sessions (refer to Table II) and W3 Sessions (refer to Table II) in network environment. For this purpose we inject the traffic generated by the two types of sessions into a queuing system of deterministic service while the service rate is chosen as a function of equivalent bandwidth estimation values. We compare values of observed packet loss rate of both estimation methods for the equivalent bandwidth. We show results as function of the number of connections. Note that we evaluate the equivalent bandwidth with 30 packet size buffer and 1% packet loss rate.

TABLE I. VALIDATION OF THE EQUIVALENT BANDWIDTH

N	EB VoIP Kbps	Loss rate %	EB Web Kbps	Loss rate %
100	3164	1.45	9494	0.9
200	6330	1.37	11531	1.5
300	9494	1.24	16142	1.38
500	15824	1.11	25490	1.09
1000	31648	1.03	50980	1.07

The estimated equivalent bandwidth guarantees the average packet loss rate (1%) especially when the number of connections is important. Notice that constant packet inter-arrivals approximation during ON periods for Web sessions lead to acceptable results. Although this is not the real behavior of inter-arrivals when TCP is used, the equivalent bandwidth estimated with this method guarantees an acceptable loss rate. This approximation allows analytical estimation of the equivalent bandwidth for Web sessions directly.

V. QoS MECHANISMS WITH SIP

The goal of this section is to introduce novel mechanisms for QoS management on a per-session basis. The SIP proxy server will be delegated for the implementation of these mechanisms. The first mechanism relies on the scheduling of TCP sessions based on session duration and data volume exchanged during a session. The second mechanism uses the equivalent bandwidth estimation methods to allocate bandwidth per flow. The SIP proxy server is supposed to achieve measures on session durations and data volume exchanged, as well as the equivalent bandwidth estimation based on session parameters exchanged during session initiation phase.

A. *Scheduling of TCP sessions*

Numerous studies show that 80% of internet flows are carried by TCP. It is also shown that 80-90% of the traffic is carried by only 10-20 % of the flows (big file transfers) while 80-90 % of the flows carry only 10-20% of the traffic. It is obvious that TCP requires special attention and particularly the interaction between big and small data transfers must be considered in any QoS provisioning mechanisms. Indeed, several researches dealing with the efficiency of TCP congestion control mechanism in congested networks have been undertaken. However, results show that losses have dramatic consequences on short TCP connections. It was suggested that according higher priority to short TCP connections constitutes a good solution to this problem [18]. The question of differentiating long from short TCP connections requires modifications in TCP headers to perform measures on TCP connections (Duration or data volume exchanged) reader can refer to [22,23] for some other proposals. In all cases this issue was always addressed at the connection level. On the other hand, an application level solution to this problem is more appropriate and easier to implement. Indeed, we can consider the user behavior during all the session as a whole and instead of differentiating short from long connection we distinguish small from big sessions.

Using SIP we can manage user communications at the session level with session scheduling based on session level criterion. This is achieved by supervising mechanisms implemented in the SIP Proxy server in the extended SIP architecture. The main advantage of our approach is that supervising mechanisms relies on measures that are performed usually for billing purposes. By consequence, no modification on TCP headers are required and no extra transmission overhead is supported.

1) The Concept

The main idea behind session scheduling is to change traffic priority dynamically during communication based on real time measurements. The goal is to minimize the impact of long TCP sessions on both short TCP sessions and real time traffic. Indeed, the SIP proxy server measures in real time the duration of TCP sessions and data volume exchanged during one session. All sessions initiated by the SIP server has the same priority at first. Sessions lasting more than average session duration T_{rs}, or exchanging more than average data volume V_{rs} are automatically declassed into a lower priority class of traffic.

In order to calculate the values of T_{rs} and V_{rs} we introduce the notion of the reference session RS. The reference session RS represents the threshold session activity under which the user session is considered as small. The notion of small session may refer to the duration or the data volume of a session. This is quiet different from the connection notion in which long connections are synonym of big file transfers. In fact, TCP sessions may contain long idle periods and the notion of duration may lead to some wrong differentiation between sessions. Once the reference session RS is defined, the theoretical duration and data volume exchanged during reference session RS can be calculated. Let:

Nbr : Average number of ON periods in a session,

T_{ON} : Average duration of ON period (Sec),

T_{OFF} : Average duration of OFF period (Sec),

R_{ON} : Average rate in ON period (Kbps),

Q_{ON} : Average file size in ON period (Kb) (to be used only with TCP based models)

In order to calculate the T_{ON} value, we consider the TCP session transmission in the free loss case (the simplified form of TCP connection duration equation (17)):

$$T\left(\frac{Q_{ON}}{Ps}\right) = RTT\log_{1.57}\left(\frac{Q_{ON}}{Ps}\right) \text{ with } Ps \text{ packet size}$$

The average session duration is given by:

$$T_{rs} = Nbr(T_{ON} + T_{OFF}) \qquad (18)$$

And the average session data volume is given by:

$$V_{rs} = Nbr.Q_{ON} \qquad (19)$$

2) The Algorithms

Here we develop the two algorithms based on reference session duration and data volume exchanged. Those algorithms are implemented by the SIP proxy server and are applied to all incoming TCP sessions.

Algorithm 1. Service class differentiation based on session duration/volume

```
Calculate reference session duration $T_{rs}$ and data volume $V_{rs}$
Define two service classes High and Low
Accept all incoming TCP sessions with the High service class
For all sessions
    If session duration > $T_{rs}$ (or session data volume > $V_{rs}$ )
        Declass the session service to the Low class
    End
End
```

B. Resource allocation

The scheduling of TCP sessions is a posteriori solution to network congestion. Indeed, it minimizes interaction between big and small TCP session after some threshold. Although it requires no information about TCP sessions, the choice of its threshold may problematic. Actually, it influences the performance of the algorithm and the overall gain in terms of QoS. On the other hand, the role played by the SIP proxy server can be enhanced to suggest a priori solutions that prevent interaction between big and small TCP sessions. Hence, instead of detecting big TCP sessions after some threshold, users may declare their sessions previously. According to the session type required by the user a different class of service may be assigned and consequently an appropriate QoS is obtained.

1) The Concept

Resource allocation requires equivalent bandwidth estimation per session type. The idea is delegate the SIP proxy server to evaluate the equivalent bandwidth of TCP flows per type of session. In order to achieve this estimation the SIP proxy server needs some specific description of initiated sessions. This will be achieved by the SDP protocol associated with SIP.

Assuming a Poisson arrival distribution of client sessions, an equivalent number of sessions N can be estimated by the Erlang B formula for a determined blocking probability. Then an equivalent bandwidth estimation procedure is launched based on session information exchanged during the SDP communication phase. Once the equivalent bandwidth is determined, a bandwidth sharing process is undertaken by Weighted Faire Queuing (WFQ) system at the Edge router. Particularly, weights are chosen as function of equivalent bandwidth and available bandwidth.

The goal is to assign the required bandwidth to small TCP sessions, while big TCP sessions share the residual bandwidth. In the following section we will present the algorithm that will be implemented in the SIP proxy server while considering only two flows types. Of course this approach may be extended to several flows of different types (not only TCP sessions).

2) The Algorithm

Consider two types of TCP sessions: small and big. The goal is to allocate resources for small TCP sessions to eliminate interaction between the two types. The SIP proxy server handles the following parameters:

Bp The blocking probability
D Poisson session arrival rate (Session/sec)
B Available bandwidth
And the following session parameters:
Nbr : Average number of ON periods in a session
T_{OFF} : Average duration of OFF period (Sec)
Q_{ON} : Average file size in ON period (Kb)

Algorithm 2: Flow based bandwidth reservation

```
For small TCP sessions calculate the equivalent number of sessions $N$
$B_p(N) = \frac{A^N / N!}{\sum_{i=0}^{N} A^i / i!}$    With $A = \frac{\lambda}{\mu}$
Estimate the equivalent bandwidth $B_{eq}$ for $N$ sessions
Define two classes of service $S$ for small and $B$ for Big
Adjust the WFQ weights so that:
$W_S = \frac{B_{eq}}{B}$ and $W_B = \frac{B - B_{eq}}{B}$
Assign small TCP sessions to the $S$ class of service
Assign big TCP session to the $B$ class of service
```

C. Call Admission Control for TCP sessions

So far only the issue of minimizing interaction between flows was addressed, first by scheduling of sessions and then by resource allocation. Indeed, a more natural role can also be assigned to the SIP proxy server, which is Call Admission Control (CAC). Whether it concerns small or big sessions, the overhead that may be induced by the initiation of new sessions may be considerable when network fall into congestion.

Thus, the scheduling of sessions will have no effect if the number of small TCP sessions exceeds system capacity. Moreover, big sessions will endure extremely long time of service due to the system overhead. Indeed, this issue is of particular interest for TCP sessions as the session duration is tightly linked with the loss rate observed on network links. If the number of accepted sessions is larger than system capacity, higher packet loss rate may be observed and longer transmission times are needed (because of TCP retransmission mechanisms). Let users being connecting to the system according to Poisson process with D session arrival rate (Session/sec) and let T be the session duration, then the average number of users N present in the system may be obtained by the $Little$ formula:

$$N = D * T \qquad (20)$$

Notice that $N \underset{T\to\infty}{\to} \infty$.

Hence, the increase of session duration results in increasing number of active sessions in the system. Therefore, the session management overhead for the SIP proxy server will increase rapidly affecting the performance of the server itself.

Call admission control could be associated with the previous proposed algorithms to guarantee normal functioning of the network. This requires the definition of the Call Admission Control Threshold (CAC_{th}) per type of sessions. When resource allocation is performed this threshold is simply defined by bandwidth reserved for one flow (denoted B_{eq}). Indeed, the estimation of B_{eq} rely on the number N of sessions, thus:

$$CAC_{th} \leq N \qquad (21)$$

On the other hand, when session scheduling is considered this is more complicated. In fact, the result of accepting new connection could not be evaluated if the equivalent bandwidth required by the incoming connection is not known. One solution may be to achieve call admission control based on real time measurements of resource requirements. However, in the scope of our study we will only consider integrating CAC with resource allocation algorithm based on bandwidth estimation.

1) CAC with bandwidth allocation

Basically, the session information used for equivalent bandwidth estimation and allocation is also used by CAC algorithm. Thus, for every new incoming connection the equivalent bandwidth necessary to allow the transmission of the flow is calculated. If the estimated value does not exceed the maximum bandwidth allocated for the flow the connection is accepted otherwise it is rejected.

Finally, we note that resource allocation procedure is based on session information exchanged before session initiation. The type of the session declared by the user is determinant for accepting or rejecting his demand. Meanwhile, in the case of wrong session type declaration, big sessions may be initiated as high priority ones causing the deterioration of the overall performance of the system. Therefore, it is possible to combine the scheduling of sessions algorithms to declass wrongly declared sessions to lower class of service as a posteriori validation mechanism.

Algorithm 3: CAC with bandwidth allocation

```
Bandwidth allocation algorithm (same steps as in Algorithm **2**)
For every incoming session
    Calculate the new equivalent session's number $N'$
    Estimate the equivalent bandwidth $B'_{eq}$ for $N'$ sessions
    If $B'_{eq} < B_{eq,\max}$
        Accept the connection
    Else
        Reject the connection
    End
End
```

VI. Simulations

The QoS session mechanisms proposed in previous sections are tested in a simple network of two nodes representing the two LER routers. The DiffServ domain is modeled by a bottleneck link between two LER routers. Bottleneck delay is of 10 ms (used for the RTT estimation). An SIP Proxy server is charged of initiating different kind sessions.

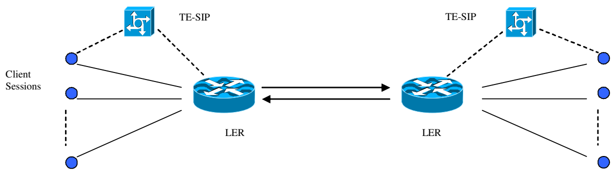

Figure 3. Testbed network

We consider three types of web sessions (W1, W2, W3), and a VoIP application session (G729). Four flows are generated (one for each type of session) in order to evaluate the suggested QoS mechanisms. Indeed, the G729 flow is only used for performance evaluation ends while data sessions are handled dynamically by the QoS mechanisms. Session parameters are listed in Table II.

TABLE II. SESSION PARAMETERS

Session	Q_{ON} Mean (Kb)	Q_{ON} Variance (Kb)	T_{OFF} Mean (Sec)	T_{OFF} Variance (Sec)
W1	20KB	40KB	10 sec	20 sec
W2	50KB	100KB	20 sec	40 sec
W3	100KB	200KB	30 sec	60 sec
VoIP	Packet Size (Bytes)	Packet Inter-arrival (ms)	T_{ON} (sec)	T_{OFF} (Sec)
G729	70	30	0.352	0.65
G711	136	12	0.352	0.65

Given the above session parameters we calculate the flows parameters that will be used when applying QoS mechanisms by the SIP proxy server. Blocking probability value is 1% for all flows (Table III).

TABLE III. FLOW PARAMETERS

Flow	D Session/sec	T Sec	N (equ)	B_{eq} Kbps
W1	0.1	101.3	21	283.5
W2	0.05	203.9	21	411.6
W3	0.0333	307.8	21	546
G729	0.0556	180	21	130.2

Recall that the estimation of TCP session duration is only possible in free loss transmission. This estimation is used to determine the reference duration (and volume) for scheduling of sessions. Packet transmission is achieved by TCP new Reno implemented from end to end in the simulator. Packet sizes are of 1024 bytes with 40 bytes ACKS.

A. *Test of Algorithm 1*

We consider only two classes of traffics High (Priority 1) and Low (Priority 2). All sessions start transmitting packets in the High traffic class. VoIP sessions stay always in the High traffic class and do not change their class. Declassing sessions concerns only TCP sessions. The goal of our test is to give short Web sessions represented by W1 sessions higher priority on other TCP sessions using scheduling of sessions. In this case, the W1 session is the reference session RS and its transmission time without losses is the reference time ($T_{rs}=101.3$ sec), and the data volume exchanged during the RS session is $V_{rs}=200$ Kbytes (those values are obtained based on W1 session parameters). We compare the results of proposed scheduling mechanisms with FIFO queue system without any priority classes. Results are shown on Table IV.

TABLE IV. FLOWS STATISTICS WITH DIFFERENT PRIORITIES

Flow		Loss %	Delay ms	Average session duration sec
FIFO	W1	3%	155	299
	W2	2.9%	151	401
	W3	2.7%	156	497
	G729	3.5%	170	180
Time Priority	W1	2.9%	152	270
	W2	2.5%	149	395
	W3	2.3%	145	478
	G729	1.6%	165	180
Volume Priority	W1	0.9%	78	105
	W2	2.7%	150	393.9
	W3	3.1%	152	487.8
	G729	1.1%	86	180

Results show no big difference in performance when using time based session scheduling. Indeed, if we analyze the activity of W1 sessions we note that idle periods are very long compared to activity time during web sessions. Declassing sessions according to the time passed on the network is not profitable in this case. In fact, the communication duration criterion could only be used for FTP type like sessions where there are no idle times.

Conversely, we see that the performance of W1 sessions, G729 sessions has improved considerably in the case of session declassing based on data volume exchanged during a session. In fact, W1 session duration is closer to theoretical value without losses, while G729 sessions endure less packet loss rate.

B. *Test of algorithm 2*

Using the same previous example sessions, we calculate the weights of WFQ system based on the estimation of the equivalent bandwidth per flow. We consider two WFQ queues in the edge router with two corresponding traffic classes: High and Low. The weights are calculated to allocate the required bandwidth for the High traffic class. The Low traffic class takes the residual bandwidth. Table V depicts the calculated weights based on sessions parameters defined before.

TABLE V. WFQ WEIGHTS FOR BANDWIDTH SHARING

Flow	Beq Kbps	Traffic class	Weight	Bottleneck bandwidth kbps
W1+G729C	413.7	High	11	1200
W2+W3	957.6	Low	19	1200

TABLE VI. FLOWS STATISTICS WITH WFQ

Flow		Loss %	Delay ms	Average session duration sec
Bandwidth allocation	W1	0.3%	25	101.9
	W2	2%	145	375
	W3	2.1%	151	437
	G729	0.05%	170	180

Results are better than first mechanism. Loss rate and delay on the W1 flow and G729 flow are smaller than previous tests. Especially, W1 session average duration is very close to theoretical estimation. Even W2 and W3 sessions performs better compared to the FIFO system case. This is especially due to the better bandwidth utilization and the isolation factor resulting in less interaction between flows.

To illustrate the robustness of this approach we show on Fig. 5 the evolution of session durations while the link bandwidth is reduced from 1200 Kbps to 800 Kbps by 100 Kbps step. The SIP proxy server adjusts weights of the WFQ queuing system to guarantee the required bandwidth for W1 and G729 flows while the W2 and W3 flows gets always the residual bandwidth. The curves show that the measured W1 session duration is stable while the measured W2 and W3 session durations increases as the bandwidth is reduced.

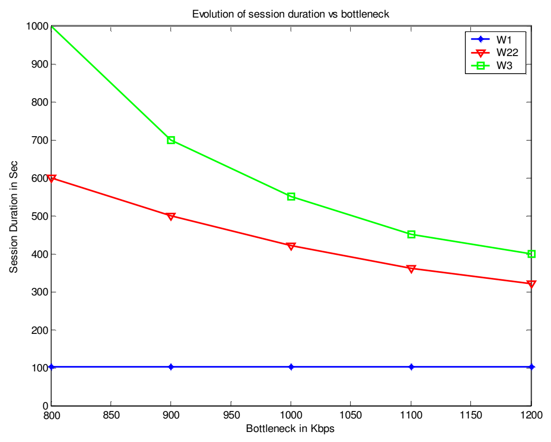

Figure 5. Evolution of session's durations vs bottleneck

The stability of the duration of small TCP session could be seen as a QoS parameter for Web sessions always served as best effort traffic. It has bigger impact in wireless networks where bandwidth is a precious resource, and guaranteed average throughput is an important parameter of the service.

VII. CONCLUSION

In this paper, we presented an SIP based framework for QoS at the session level. Using the SIP proxy server with extended architecture over DiffServ domain, we can implement session scheduling and bandwidth allocation mechanisms to minimize the interaction between small and big TCP sessions. The suggested mechanisms rely on session level real time measurements of session duration and data volume exchanged during a session as well as equivalent bandwidth estimation technique. The required measurements do not cause additional overhead as they are achieved for billing purposes. However, bandwidth allocation requires more specific session description that should be exchanged between the SIP proxy and the client using the SDP protocol. Bandwidth allocation is based on equivalent bandwidth estimation per flow. Indeed, we proposed a $GI/D/1/K$ queue system model to evaluate the equivalent bandwidth using the first and second order moment of packet inter-arrivals and packet service processes. However, some heuristics to measure the equivalent bandwidth in real time could be considered using iterative algorithms. This may be useful also to implement adaptive call admission control mechanisms.

During our study we only considered homogenous flows (per type of session) for the estimation of equivalent bandwidth. In fact, the estimated equivalent bandwidth depends on traffic mixes and may be influenced by the different packet sizes according to application types (and transport protocols). Particularly, in VoIP application packet sizes are very small comparing to Video and TCP sessions resulting in important covariance on the packet service process and the overall performance.

Finally, we would like to note the QoS mechanisms proposed are applied on a per-session basis, and this may result in an extra delay on session initiation process for users. Some aggregated reservation techniques [11] may be useful in this case to enhance the SIP proxy server response time.

REFERENCES

[1] F. P. Kelly. "Effective bandwidths at multi-class queues". *Queueing Systems*, 9:5--16,1991.
[2] R. Geurin, H. Ahmadi, and M. Naghshineh. "Equivalent capacity and its application to bandwidth allocation in high-speed networks". *IEEE J. Select. Areas Commun*, 9(7):968-981, September 1991.
[3] C. Courcoubetis and R.Weber. "Effective bandwidths for stationary sources". *Prob. Eng. Inf. Sci.*, 9:285--296, 1995.
[4] A. Elwalid and D. Mitra. "Effective bandwidth of general Markovian traffic sources and admission control of high speed networks". *IEEE/ACM Trans. on Networking*, 1(3):329--343, October 1993.
[5] C. Courcoubetis and R. Weber. "Buffer overflow asymptotics for a switch handling many traffic sources". *Journal of Applied Probability*, 33:886-903, 1996.
[6] A. Simonian and J. Guibert. "Large deviations approximations for fluid queues fed by a large number of on-off sources". *IEEE J. Select. Areas Commun.*, 13(7):1017--1027, August 1995.
[7] F. P. Kelly. "Notes on effective bandwidths". In F. P. Kelly, S. Zachary, and I. Zeidins, editors, *Stochastic Networks: Theory and Applications,* pages 141--168. Oxford University Press, 1996.
[8] G. de Veciana and J. Walrand. "Effective bandwidths: Call admission, traffic policing and filtering in ATM networks". *Queuing Systems*, 20:37--59, 1995.
[9] N. Likhanov and R. R. Mazumdar. "Cell loss asymptotics for buffers fed with a large number of independent

stationary sources". *Proc. of IEEE INFOCOM'98*, San Francisco, USA, March 1998.
[10] W. Whitt, "Tail probabilities with statistical multiplexing and effective bandwidths in multi-class queues," *Telecommunication Systems 2*, pp.71-107, 1993.
[11] B. Rong, J. Lebeau, M. Bennani, M. Kadoch and A. Elhakeem, "Traffic Aggregation Based SIP over MPLS Network Architecture", *Proceedings of International Conference on Advanced Information Networking and Applications* (AINA 2005), 2005.
[12] S. Salsano and S. Veltri, "QoS Control by Means of COPS to Support SIP-Based Applications", *IEEE Network*, Vol. 16, Issue : 2, pages 27-33, 2002.
[13] C. Zhang and C.G Guy, "TE-SIP Server Design for a SIP-over-MPLS Based Network", *Proceedings of International Conference on Communication Technology* (ICCT 2003), Volume : 2, pages 1758-1761, 2003.
[14] http://www.ietf.org
[15] http://www.ietf.org/rfc/rfc3261.txt
[16] http://www.ietf.org/rfc/rfc822.txt
[17] La qualité de service de la voix sur IP, principes et assurance. http://www.accellent-group.com
[18] U. Ayesta, "Stochastic Scheduling and its Application to TCP/IP Networks", *Thesis of INRIA*, Nice-Sophia Antipolice, 2005.
[19] W. Whitt, "The queuing network analyzer". *Bell System Technical Journal*, pages 2779-2813, November 1983.
[20] R. Nagarajan, J. F. Kurose, D. Towsley, "Approximation Techniques for Computing Packet Loss in Finite-Buffered Voice Multiplexers", *IEEE Journal on Selected Areas in Communications* (1991).
[21] B Sikdar, S Kalyanaramana and K.S Vastola, "An Integrated Model for the Latency and Steady-State Throughput of TCP connections". *Performance Evaluation*, Vol 46, no. 2-3, pp. 139-154, 2001.
[22] C. Estan and G. Varghese, "New directions in traffic measurement and accounting: Focusing on the Elephants, ignoring the Mice" *ACM transactions on Computer Systems*, August 2003.
[23] C. Psounis A. Ghosh and B. Prabhakar. "SIFT: a simple algorithm for identifying large flows", *Presented at the Stochastic Networks Conference*, Montreal, Canada, 2004.
[24] K. Sriram and W. Whitt, "Characterizing Superposition Arrival Processes in Packet Multiplexers for Voice and Data", *IEEE Journal of Selected Areas in Communications*, pp.833-846, Sep, 1986.
[25] H. Hassan, J-M Garcia and C. Bockstal "Aggregate traffic models for VoIP applications", ICDT 2006, in press.

Hassan HASSAN. is a Ph.D. student at Laboratoire d'Architecture et d'Analyse de Systèmes of CNRS, Toulouse, France. He received his MS degree in Networks and Telecommunications from the ENSEEIHT, Toulouse, France, in 2003 and his B.E. degree in Electronics from ENSERG, Grenoble, France, in 1996. His research interests include Multimedia Traffic Modeling and Performance Analysis in Heterogeneous Networks.

Jean-Marie GARCIA. received his Ph.D. from the Université Paul Sabatier, Toulouse, France, in 1980. He received his B.E. degree from Institut National des Sciences Appliquées, Toulouse, France, in 1976, and the M. A. Sc degree from the Ecole Polytechnique, Montreal , P.Q., Canada, in 1978. He performed Research at the Institut of Hydro Quebec, Varennes, P.Q., Canada, in 1977. He spent one year as a postdoctoral fellow at the Electronics Research Laboratory, University of California, Berkeley, in 1981, and in 1982 he joind the Laboratoire d'Analyse et d'Architecture de Systèmes of CNRS, Toulouse, where he is presently directeur de recherches. His main interests are modeling and control of telecommunication networks, optimization and resource management aspects.

Olivier BRUN. received his Ph.D. from the Université Paul Sabatier, Toulouse, France, in 2000. He received his B.E. degree from Institut National des Telecommunications, Evry, France, in 1995. He spent one year as a research engineer in the Delta Partners company where he developed a modeling tool for large-scale telephone networks. In 2002, he joined the Laboratoire d'Analyse et d'Architecture de Systèmes of CNRS, Toulouse, where he is presently charge de recherches. His research deals with stochastic modeling and network planning.