Theoretical Analysis of an Ideal Startup Scheme in Multihomed SCTP

Johan Eklund[1], Karl-Johan Grinnemo[2], and Anna Brunstrom[1]

[1] Department of Computer Science, Karlstad University, SWEDEN
[2] School of Information and Communication Technology, KTH, SWEDEN

Abstract. SCTP congestion control includes the slow start mechanism to probe the network for available bandwidth. In case of path swap in a multihomed association, this mechanism may cause a sudden drop in throughput and increased message delays. By estimating the available bandwidth on the alternate path it is possible to utilize a more efficient startup scheme. In this paper, we analytically compare and quantify the degrading impact of slow start in relation to an ideal startup scenario. We identify three different scenarios, where a path swap could occur. Further, we identify relevant traffic for these scenarios. Our results point out that the most prominent performance gain is seen for applications generating high traffic loads, like video-conferencing. Moreover, the results show an increasing impact of an improved startup mechanism with increasing RTTs.

1 Introduction

New applications with a diversity of requirements regarding timeliness and robustness continuously appear in computer networks. This is challenging for the traditional communication protocols and the Stream Control Transmission Protocol (SCTP) [16], with its multihoming ability has become an attractive alternative to address these challenges. The protocol is not a totally new invention; it has in fact inherited some crucial features from the most commonly used transport protocol in computer networks, TCP [13]. One of these features is the congestion control, including the slow-start mechanism, which sets the initial congestion window to a small value, and the sender then successively increases the window as acknowledgements of successful transfers are returned. This startup mechanism was introduced to preserve network stability, and to respect fairness between competing connections over the same path. On the other hand, if there is spare bandwidth in the network, slow start increases the time it takes before the traffic may utilize the bandwidth.

The multihoming feature of SCTP implies that a connection, a so-called association, may consist of more than one path, all available from the start of a session. This mechanism was included in the protocol to increase robustness in case of failure, as the transfer may proceed utilizing one of the alternate paths, without having to establish a new association. In this case the path swap is called a failover.

After standardization, new mechanisms have been added to SCTP. One of these additional mechanisms is the so-called Dynamic Address Reconfiguration, (DAR) [17] mechanism, whereby IP addresses may be added or deleted to/from an existing association dynamically. This feature was added to enable maintenance of the network without disturbing the association, a so-called hotswap. Furthermore, this feature enables SCTP to be an alternative for seamless mobility. Particularly, the DAR mechanism can be used to support handover for wirelessly connected terminals moving between different networks.

One aspect that may have an impact in all the above mentioned scenarios, failover, hotswap and handover between different paths, is that the transfer has to go through the slow-start phase every time the association is switched to a new path. The small initial congestion window on the new path may result in a sudden reduction of throughput and increased message transmission delay, which may severely impede the current transfer.

Today, several techniques exist to estimate the available bandwidth of a network path. These estimates are based either on data from traffic sent over the network, for example, the active packet-pair technique [12] or on passive network monitoring [5]. In a multihomed association, this estimate could be performed before an expected path swap, since alternate paths are available prior to the path swap.

Some research on improving the startup of the transfer on the alternate path has earlier been conducted. In a study by Zheng et al. [11], they estimate the bandwidth on the alternate path by utilizing a packet-pair technique [12]. Based on the estimate, they enlarge the initial congestion window of the alternate path at the time of handover. Further, they regard the latency of the new path in relation to the former path to reduce the risk of packet reordering. Another study, proposing a handover scheme called ECHO, was conducted by Fitzpatrick et al. [8], where they aim at improving the handover scheme SIGMA [4] for traffic in a wireless environment. The improvement in ECHO is that this scheme regards the quality of service by evaluating the Mean Opinion Score(MOS) [7] before handing over to the new network. The above mentioned studies are performed with mobile clients in mind and the focus is on transfer of Voice over IP (VoIP) traffic.

Fracchia et al. present a modification, WiSE [14], to make SCTP more suitable for wireless networks. WiSE aimed at always choosing the best path for the transfer by estimating the bandwidth of all existing paths in case of a timeout. Further, WiSE possesses a mechanism to utilize the bandwidth estimate to differentiate losses due to congestion from losses due to radio channel errors to properly adjust the congestion window. An extension of WiSE is presented by Casetti et al., called AISLE [2], which aims to optimally distribute traffic across overlapping WLANs. Unlike WiSE, AISLE triggers a potential path change also on fast retransmits and not only on timeouts. Both these proposals have been focusing on bulk transfer.

In this paper, we analytically examine the potential performance gains of using a bandwidth-aware startup scheme for multihomed SCTP, where the initial

congestion window on the alternate path is set to an estimate of the available bandwidth. We identify three scenarios, where a path swap may occur. Further, we present relevant traffic patterns for these scenarios. The main contribution of the study is that we quantify the theoretically feasible performance gains for the mentioned scenarios, by setting the initial congestion window to an ideal size. We do not consider a particular bandwidth estimation technique, but rather make the assumption that the estimated bandwidth is correct and stable. Although, this assumption does not match a real scenario, it allows us to analytically derive and quantify the upper bounds for the performance gains, in a range of relevant traffic scenarios.

Intuitively, applications with real-time requirements may be affected by extra delay after a path swap. Our results indicate that the benefit from an improved startup scheme is most considerable for applications of this category generating high traffic loads and the impact increases with increasing RTTs. Furthermore, the results show that the traffic pattern affects the impact in two ways; packets generated at high frequency may have to wait a long time to be transmitted, while large messages may increase the congestion window fast.

The rest of this paper is structured as follows. In Section 2, different use cases and traffic characteristics are discussed, while Section 3 presents analytical results from comparing an ideal startup to slow start for different traffic scenarios. Finally, in Section 4, the paper ends up with a discussion and some conclusions.

2 Scenarios and Traffic Patterns

To evaluate the benefit of an ideal startup mechanism, we match the three path swap scenarios; failover, hotswap and handover, to relevant traffic types. To be able to perform an appropriate evaluation it is indeed important to extract the characteristics for the different traffic types.

2.1 Scenarios

The three scenarios where a path swap may occur have different characteristics:

- *Failover.* This scenario may occur as a consequence of path failure. In many cases, the failover is preceded by a failure detection period. During this period no data reaches the destination without retransmissions, which degrades performance. In situations where the failure happens to the network interface or to the first or last hop link, the failure is usually detected immediately and a path swap is not preceded by the aforementioned failure detection period.
- *Hotswap.* This is a scenario where a network path is swapped intentionally, due to maintenance or upgrading of a network component. This event is a planned swap, why it is not preceded by any failure detection period. In this scenario, the only throughput degradation is due to the slow-start mechanism after startup on the alternate path.

– *Handover.* This scenario occurs in a mobile scenario, where a wirelessly connected terminal moves and swaps from one network to another. In this scenario, the path swap is usually conducted as the performance, according to application specific aspects, on the new path, exceeds the performance of the current path. In the handover scenario, throughput degradation could occur before handover as well as during startup on the alternate path.

Not all traffic patterns are applicable for all path-swap scenarios. The handover scenario is applicable for user data, while the other scenarios are more applicable for control traffic. A view of the connection between scenario and applicable traffic is shown in Table 1.

Table 1. Traffic types representative of different scenarios

	Failover	**Hotswap**	**Handover**
Traffic Type	signaling	signaling	real time/bulk

2.2 Characteristics for Different Traffic Types

– *Signaling Traffic.* The traffic generated by signaling applications normally consists of small messages with slightly varying size. The messages are generated at irregular intervals, usually in bursts [1, 15]. Each signaling message carries its own piece of information, why subsequent messages are generally independent. The requirement on signaling traffic is that the message should reach the destination within a specific time, so both reliability and timing issues are relevant.
– *Real Time Traffic.* Some major applications generating real-time traffic are VoIP and video-conference applications. The crucial part for this type of traffic is timeliness, i.e., that the data reaches the destination within specified time, while robustness is less critical since a single lost data message only imposes marginal impact on user experience. Applications for VoIP traffic falls into two categories: those with silence detection and those without. If silence detection is used, the data to be transferred is generated only during so-called talk spurts, while applications without this feature generate traffic continuously. In this work, we focus on the latter case, which more or less generate traffic at constant bit rate (CBR).
  The traffic generated during a video conference is also usually transferred at a constant frequency. Data from these applications, called frames, carry information blocks from the video. The frame size varies according to the amount of information sent. This traffic is more complex to model as the generated bit rate is variable (VBR).
– *Bulk Traffic.* Bulk traffic is characterized by the availability of all data at session start. The data is sent on the link as soon as possible. Traditionally, timeliness was not an issue for bulk data. However, in recent years, short

flows of traffic with similarities to bulk data, for example, interactive web transfers and streaming media, where timeliness is usually an issue, have become more and more common. These recent forms of traffic are generated by applications with timing requirements and the traffic has characteristics in common with bulk traffic. Thus, these types of traffic are in this study categorized as bulk traffic.

3 Analysis of Performance Improvements by an Ideal Startup Mechanism

In this section, we compare the impact of the the traditional slow-start mechanism with that of an ideal one. For the ideal mechanism the initial congestion window is set to a size that utilizes all available bandwidth.

3.1 Assumptions

To enable an analytical approach and a clear presentation, we have made a few simplifying assumptions:

- Lossless links are assumed. In fact, losses only occur after failure on the primary path.
- The available bandwidth is constant, and for non-bulk traffic bandwidth is not a limiting factor.
- A symmetric network is assumed, i e., all available paths have the same bandwidth.
- Transmission time for packets is negligible and does not vary with packet size.
- Overhead for headers are not regarded in the calculations.
- In the situations where the slow-start mechanism is used, the initial congestion window is set to 4500 bytes[3][16].

These assumptions somewhat estrange the situation from a real scenario, but despite these assumptions, we believe the results will be relevant to identify the scenarios where a more efficient startup is most beneficial, and to point out the magnitude of the performance improvement.

3.2 Parameters and Metrics

As mentioned before, different applications generate different types of traffic which results in different message sizes generated at different frequency. Signaling traffic is in this study modeled as messages resulting in packets of 250 bytes,

[3] According to RFC 4960, the initial congestion window is MTU-dependent. To simplify the expression of the calculations we have chosen 4500 as the size of the initial congestion window, since 4500 bytes is an even multiple of the assumed path MTU (1500 bytes) and of the packet size for signaling messages (set to 250 bytes in our study). This, to exclude the plausible impact of congestion window overgrowth.

a plausible average size of a signaling message. Signaling traffic in the access network is usually not very intense. In the core network, on the other hand, traffic from several signaling endpoints may be aggregated into a common association where the traffic intensity may be quite high. In this study, we model signaling traffic inter arrival times between 1 ms and 0.5 s to cover both situations. Since the aim of the study is to compare the slow-start mechanism to an ideal situation, the delay of an arriving message will depend on the queuing delay created by previously arrived messages. Thus, the quantity of sent data and not the distribution of data between the different messages is the most important. Therefore, we can simplify the modeling by approximating VBR traffic to CBR using the average message size. In this study, we approximate the signaling traffic with CBR traffic with messages of uniform size.

Concerning VoIP traffic, the frequency and the size of the packets vary depending on codec used. One commonly used codec is G.711, which typically packetizes data into messages of 80 bytes, transmitted every 20 ms [6]. We have used this traffic pattern to model VoIP traffic in the study.

As mentioned in Section 2, video traffic is sent as VBR traffic, where the frame sizes vary according to several parameters. A recent codec is H.264 [19], which generates different types of frames with great variability between different frames. Depending on how the codec is configured, different traffic patterns are generated. In this study, we will utilize a high definition trace taken from the "horizon" talk show, provided by Reisslein et al. [3] for network evaluation purposes[4]. With the same motivation as for signaling traffic, we approximate the video traffic to CBR. From the trace, we have calculated the mean frame size to be about 6000 bytes and on average 30 frames are generated per second. This will be the input representing video traffic in this study. All the above mentioned combinations have been calculated for a range of RTTs varying from 5 to 250 ms. A complete view of the parameters used in this study is found in Table 2.

Table 2. Parameters

Traffic	Message size (Bytes)	Message interval (ms)	RTT (ms)
Signaling	250	1-500	5-250
VoIP	80	20	5-250
Video	6000	33	5-250
Bulk	Bulk	Bulk	5-250

Different applications do not only generate different traffic patterns, they value different traffic properties. Thus different metrics should be used for different applications. For bulk traffic, the total transfer time is important, while for signaling traffic, the Message Transfer Time (MTT) of single messages is crucial. Real-time applications, like VoIP or video-conference, on the other hand, expect smooth and timely delivery. For this reason, the results for the different

[4] http://trace.eas.asu.edu/h264/horizon/

scenarios will be presented in different forms related to the specific application requirements.

The network parameters used in the scenarios are an MTU of 1500 bytes. Further, for signaling traffic and real time traffic, Nagle's algorithm [9] and the delayed SACK [18], may be an issue. In this study, these mechanisms are assumed to be disabled.

3.3 Impact on Signaling Traffic

For signaling messages, the MTT for a single message is of importance. Further, as discussed in Section 2, two scenarios are relevant for signaling messages: the failover scenario and the hotswap scenario. In the failover scenario, several packets may, dependent on the traffic pattern, be queued at the sender during the failure detection period. Thus, at the startup, after failover, the traffic may consist of several packets waiting to be transferred. These messages will be bundled before transfer and several packets may be transferred in connection to each other. Therefore, the startup behavior for a failover scenario may be comparable to the startup for bulk traffic, described in Subsection 3.5. In the hotswap scenario, it is possible to model the MTT for a packet. When utilizing the ideal startup behavior, as there is no queuing delay, the MTT will be equal to the propagation delay, RTT/2, for all messages. In case the slow-start mechanism is used, it is possible to model the MTT under the conditions that all messages have uniform size, and that the congestion window is not decreased due to long idle periods.

In Eq.(1) it is seen that the MTT for a message[5] with number n, m_n, is at least RTT/2, which is the delay for transfer over the link. Further, it is seen that there may be an extra delay, D_n, which occurs when the message is queued at message arrival as a result of the congestion window being full. In these cases, the message will have to wait at the sender before transmission.

$$MTT(m_n) = \frac{RTT}{2} + max(0, D_n) \tag{1}$$

The above mentioned delay D_n consists of three components, found in Eq.(2). These components are:

- The number of RTT's before the message may be sent, p_n, which is calculated according to Eq.(3). p_n is dependant on the message number n, the message size s (bytes) and on the size of the initial congestion window w_i (bytes).
- The arrival time of the message, α_n, (since all data is not available initially).
- An offset, Δ_n, which is given in Eq.(4) and may occur if the message does not fit in the initial congestion window. Δ_n depends on the arrival times of messages sent in the initial congestion window, which makes it possible to calculate Δ_n by transposing the current message n to the matching message

[5] The first message in the transmission on the alternate path is given the ordinal number one.

Φ_n in the first window, according to Eq.(5). Thus it is possible to extract the arrival time of message Φ_n, α_{Φ_n}.

$$D_n = pRTT - \alpha_n + \Delta_n \tag{2}$$

$$p_n = \left\lfloor log_2 \left(\frac{ns - 1 + w_i}{w_i} \right) \right\rfloor \tag{3}$$

$$\Delta_n = \begin{cases} 0, & n \leq w_i \\ \alpha_{\Phi_n}, & n > w_i \end{cases} \tag{4}$$

$$\Phi_n = \left\lfloor \frac{n - w_i \sum_{j=0}^{p-1} 2^j}{2^p} \right\rfloor + 1 \tag{5}$$

An illustration of the formulas is seen in Fig. 1. In the illustrated scenario we assume that four messages fit into the initial congestion window, w_i, while messages 5-12 will have to wait until the first RTT is completed to be transferred. Taking message 7 as an example, it has to wait until one RTT is completed, thus p_7=1. Moreover, message 7 is sent when the acknowledgement for message 2 arrives and Φ_7 is equal to 2. The Δ_7 shown in the figure depends on the arrival time of message 2.

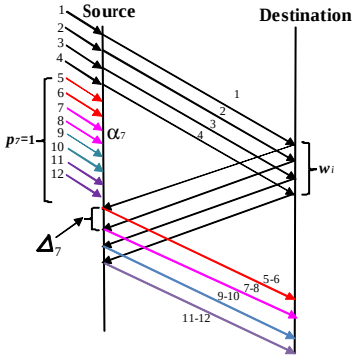


Fig. 1. Slow start scenario.

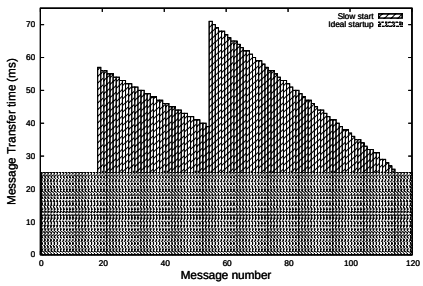


Fig. 2. MTT versus message sequence number in a hotswap scenario. Message interval 1 ms, RTT 50 ms.

An initial congestion window of 4500 bytes restricts the number of outstanding packets containing signaling messages of size 250 bytes to 18 when slow start is used. In case the RTT is greater than 17 times the message interval, messages are delayed at the sender until the first message is acknowledged according to Eq.(2). Then, the first queued message is transferred together with the next waiting message, since the congestion window is increased by the size of one message per acknowledgement.

Figure 2 shows the MTT as a function of message number in a hotswap scenario, when utilizing slow-start as well as for an ideal startup mechanism. The

figure represents messages sent at high intensity, with an average message interval of 1 ms and an RTT of 50 ms, plausible for an intra-continental association. In the figure, it is seen that all messages representing an ideal startup mechanism will have an MTT of exactly RTT/2. Further, in case slow start is used, the messages that fits in the initial congestion window will have an MTT of RTT/2. Moreover, it is seen that the first message outside this window has an extreme increase in MTT, from waiting for the first acknowledgement, while for the following messages the MTT will decrease, due to later arrival, until the congestion window is filled again. At this moment, a new peak in MTT will appear. After three round trips the congestion window has opened up enough to transfer all messages immediately.

For signaling messages it is important to reach the destination within a certain time bound. From this aspect the Maximum Message Transfer Time (MMTT), for a sequence of messages is important. In the upper part of Table 3, some MMTTs for signaling traffic are seen in relation to message interval and round-trip times. From the table, it is evident that only when RTTs are quite long, and message intervals are small, like in the core network, the slow-start mechanism has a significant impact on the MMTT for signaling traffic.

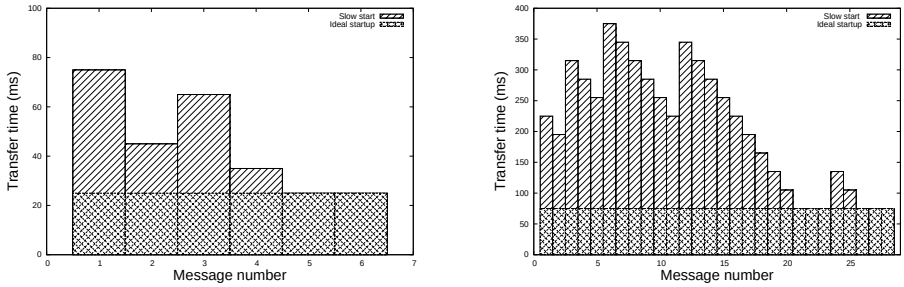
Table 3. Maximum message transfer times (ms)

		RTT (ms)		
Traffic	**Message Interval (ms)**	**20**	**50**	**100**
Signaling slow start	1	12	71	224
Signaling slow start	3	10	25	96
Signaling slow start	5	10	25	60
Signaling ideal startup	1-500	10	25	50
VoIP slow start	20	10	25	50
VoIP ideal startup	20	10	25	50
Video slow start	30	30	65	190
Video ideal startup	30	10	25	50

3.4 Real-time traffic

When considering media traffic with real time requirements, timeliness demands have to be reached, and a smooth flow of data between the endpoints is desirable. Also for real time traffic the formulas in Eq. 1 to 5 are valid to model MTT. The situation for VoIP is essentially the same as the situation for signaling traffic. The major difference is that the VoIP messages are usually smaller than signaling messages. An initial congestion window of 4500 bytes enables 56 VoIP messages to be outstanding. Thus, queuing of messages due to the restricted initial congestion window will occur only if more than 56 messages are generated before the first acknowledgement arrives. Our calculations show that slow start does not imply a real problem for this traffic, since delays only occur in cases

where RTTs exceed 1100 ms. This scenario is not likely to occur, these RTTs by themselves will degrade the performance of the VoIP call to an unacceptable level. The marginal impact of slow start for VoIP is confirmed in Table 3, where it is seen that for the considered RTTs there is no increase in MMTT for this traffic, irrespective of startup mechanism.



(a) RTT 50 ms (b) RTT 150 ms

Fig. 3. Message transfer times, traffic representing Video-conference

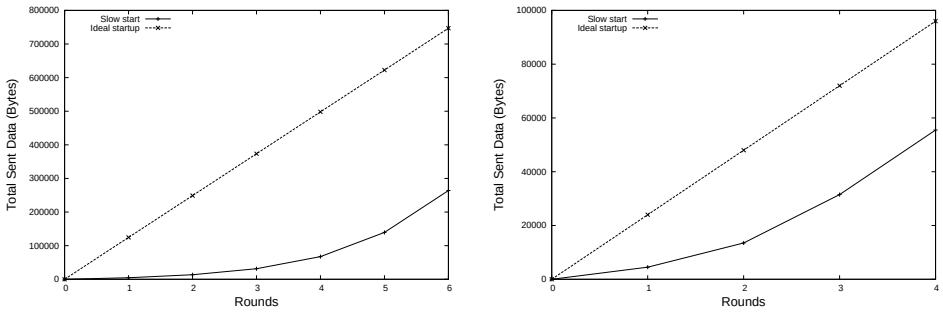
For the video-trace we have used [3], the average message size is about 6 Kbytes, which means that with a congestion window size of 4500 bytes the initial message will, in the slow-start case, not fit in the initial congestion window. Further, the second part of the message will not be transmitted until the first part is acknowledged. Intuitively, this means a dramatic impact on the transfer delay, especially if the RTT is long.

Figure 3 displays the transfer times for the first messages transferred after handover for two different RTTs, representing plausible intra- and inter continental transfers. As in the previous results, it is seen that when using an ideal startup mechanism, the delay stays constant at RTT/2. Utilizing the slow-start mechanism, on the other hand, brings significant extra delay to the transfer until the congestion window has opened up. In the figure, it is seen that MTT as well as the time it takes until data is transferred smoothly over the new path, is related to the RTT. In Figure 3(a), it takes about four messages before data is transferred without extra delay. If the RTT is longer, like in Figure 3(b), it takes about 20 messages to reach stable transfer times.

3.5 Bulk traffic

For bulk traffic, the latency for transfer of an entire file is of interest. To calculate the latency, we have used formulas for TCP presented in [10]. These formulas are applicable also for SCTP, since the protocols utilize a similar congestion control mechanism. The formulas in [10], include latency for the connection phase, which is not applicable for a failover scenario, since all available paths are established

before path swap in SCTP. Thus, we have removed the latency for the connection phase in our calculations.



(a) RTT 10 ms, Bandwidth 100 Mbps (b) RTT 100 ms, Bandwidth 2Mbps

Fig. 4. Data Bytes sent as a function of Rounds

The subfigures in Figure 4 present some results related to bulk transfer. The different subfigures represent different bandwidths and RTTs, and in both subfigures the number of sent bytes as a function of rounds are displayed for both slow start and for an ideal startup mechanism. It is seen that the difference between the different startup schemes emerges in the initial rounds, and becomes constant after some rounds. The constant difference implies that the congestion window has reached the bandwidth of the link in the slow start case.

As mentioned in Section 2.2, timing aspects are generally not crucial for bulk traffic, but recall that the impact on bulk traffic could be representative also for bulk-like traffic as well as for signaling traffic after a failover, where timing aspects may be of importance.

4 Conclusions

In a multihomed association, awareness of the available spare bandwidth on the alternate path could serve as basis for a more efficient startup mechanism than slow start. The potential performance gain could reduce the throughput decline and the message transmission delay after path swap. In this paper, we have analytically studied the impact of the slow-start mechanism in comparison to an ideal startup mechanism, where the full path bandwidth is utilized immediately. We have identified typical scenarios where the negative impact of slow start may be crucial for the user. The results from our calculations show that the positive effect from an improved startup mechanism is most prominent for CBR or VBR traffic with large payload and small intervals between messages. For this traffic, the impact of a small initial congestion window grows drastically as RTT increases. This scenario may be representative for a mobile user taking part in a video conference moving between networks.

The calculations in this paper are made assuming ideal conditions, to point out and quantify the impact of not utilizing full network capacity after a path swap. For future work we intend to complement our analytic results with experimental data.

References

1. A. T. Andersen. *Modelling of Packet Traffic with Matrix Analytic Methods*. PhD thesis, Technical University of Denmark (DTU), 1995.
2. C.Casetti, C-F.Chiasserini, R.Fracchia, and M.Meo. AISLE: Autonomic interface selection for wireless users. In Proceedings of WOWMOM '06, pages 42–48, Washington, DC, USA, 2006. IEEE Computer Society.
3. G.van der Auwera, P.David, and M. Reisslien. Bit rate-variability of H.264/AVC FRExt. technical report, arizona state university, 2006.
4. F.Shaojian, Liran.Ma, Attiquzzaman.M, and Yong-Jin.L. Architecture and performance of SIGMA: a seamless mobility architecture for data networks. In Proc. of ICC, 2005, Seoul, Korea, May 2005.
5. H.Yihua and J.Brassil. NATHALIE An Adaptive Network-Aware Traffic Equalizer. In Proc. of The ICC2007, Glasgow, UK, June 2007.
6. ITU-T. Recommendation G.711 Pulse Code Modulation(PCM) of voice frequencies. *ITU-T*, 1972.
7. ITU-T. Methods for Subjective Determination of Transmission Quality, P.800. *ITU-T*, 1996.
8. J.Fitzpatrick, S.Murphy, M.Atiquzzaman, and murphy.J. ECHO A Quality of Service Based Endpoint Centric Handover Scheme for VoIP. In Proc. of the Wireless Communications and Networking Conference, 2008, Las Vegas, USA, April 2008.
9. J.Nagle. RFC 896 Congestion Control in IP/TCP Internetworks, January 1984.
10. J. Kurose and K. Ross. *Computer Networking -A Top Down Approach Featuring the Internet (3rd.ed)*. Addison Wesley, 2002.
11. K.Zheng, M.Liu, Z-C. Li, and G.Xu. SHOP: An Integrated Scheme for SCTP Handover Optimization in Multihomed Environments. In Proc. of the Global Telecommunication Conference 2008, New Orleans, LA, USA, December, 2008.
12. N.Hu and P.Steenkiste. Estimating Available Bandwidth Using Packet Pair Probing. Technical Report 2002:166, Division for Information Technology, September 2002.
13. J. Postel. RFC 793: Transmission Control Protocol, September 1981.
14. R.Fracchia, C.Cassetti, C-F.Chiasserini, and M.Meo. A WiSE extension of SCTP for wireless networks. In Proc. of ICC2005, Seoul, South Korea, May 2005.
15. F. J. Scholtz. Statistical Analysis of Common Channel Signaling System No. 7 Traffic. In 15th Internet Traffic Engineering and Traffic Management (ITC) *Specialist Seminar*, Wurzburg, Germany, July 2002.
16. R. Stewart. RFC 4960: Stream Control Transmission Protocol, September 2007.
17. R. Stewart, Q.Xie, M. Tuexen, S. Maruyama, and M. Kozuka. RFC 5061: Stream Control Transmission Protocol Dynamic Address Reconfiguration.
18. V.Jacobson and R.Braden. RFC 1072: TCP Extensions for Long-Delay Paths, October 1988.
19. T. Weigand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVS video coding standard. IEEE Transactions on Circuits and Systems for Video Technology, pages 560–576, July 2003.