Receiver Based Stable and Minimum Queueing Policy for TCP in CDMA2000-1X

Jaegyu Jung*,† , Banghun Chun*, Youngjoo Kim*, Chuck Yoo†
*Software Center, Samsung Electronics
Shinsadong, Kangnamgu, Seoul, Korea
Email: {pang, benchun, yjkim}@samsung.com
Telephone: +82–2–3416–0592
Fax: +82–2–3416–0589
†Dept. of Computer Science & Engineering, Korea University
Anamdong, Sungbukgu, Seoul, Korea
Email: hxy@os.korea.ac.kr
Telephone: +82–2–3290–3639
Fax: +82–2–922–6341

Abstract— **There have been many researches to improve the performance of TCP operating in wireless link. A lot of prior work in this direction, however, has focused on avoiding the case of a TCP sender misinterpreting packet losses in the wireless link as congestion signals. In CDMA2000, however, a reliable RLP ensures that packets are delivered in order and help to recover form packets received in error. In this paper, we have measured the performance of transport layer protocols on a nationally deployed commercial CDMA2000-1x network. We used a cellular phone itself to measure the performance through modifications of Qualcomm TCP/UDP stack. We mainly focus on the large delay variation and excessive queueing in CDMA2000-1x. And it is evitable for traditional TCP to accumulate segment at the buffer of bottleneck link, because TCP's congestion control algorithm would fill the entire router buffer before incurring packet loss. The primary goal of our work is to analyze the exact characteristics of CDMA2000-1x, and maintains the queue length at bottleneck link can be stable without changing the TCP sender.**

I. Introduction

There has been an increasing growth in cellular telephony, which has also lead to an increased demand for wireless data services such as GPRS, CDMA2000, and UMTS [1][2]. In the United States and South Korea, third generation (3G) wide-area wireless networks are currently being deployed in the form of CDMA2000-1x technology with speeds up to 144Kbps [3]. CDMA2000 is part of the International Mobile Telecommunications 2000(IMT-2000) specification suite of access platforms that comprise what is known collectively as 3G. CDMA2000 enables a logical migration from the existing 2G platforms to 3G without plowing the legacy system [2].

As these services provide pervasive internet access, providing efficient and reliable connectivity over wireless links is thus becoming a critical issue. There have been many researches to improve the performance of TCP, the most widely accepted reliable protocol in traditional wired networks and one that is likely to be similarly important for mobile users too, operating in wireless link. A lot of prior work in this direction, however, has focused on avoiding the case of a TCP sender misinterpreting packet losses in the wireless link as congestion signals. In 3G wireless services, a reliable Radio Link Protocol (RLP) ensures that packets are delivered in order, and the Automatic Repeat Request (ARQ) in RLP combined with Forward Error Correction (FEC) helps to recover from packets received in error. These mechanisms ensure packet loss probability of less than 1% in the wireless link [4]. Actually, we could not experience any packet loss during our experiments with UDP on a commercial CDMA2000-1x network. Instead, we underwent periodic high delays even though there is no movement of mobile terminal and observed excessive queueing at the Packet Data Service Node (PDSN) that interconnects the wireless network and wired network.

We used a cellular phone itself to measure the performance through modifications of Qualcomm TCP/UDP stack. With the result of experiment on real world, we analyze the characteristics of CDMA2000-1x by a viewpoint of transport layer protocol and define the problems of TCP under such environment.

The primary goal of our work is to analyze the exact characteristics of CDMA2000-1X, and maintains the queue length at bottleneck link can be stable while achieving fully utilization of the link bandwidth without changing the TCP sender. It is impractical to change the protocol stacks of all stationary hosts merely to accommodate mobile hosts. The fundamental idea behind our solution is to control the usable window size of sender for each connection by manipulating the receiver's advertise window size. We applied TCP Vegas [16] similar mechanisms at the receiver.

The rest of this paper is organized as follow. We discuss some related works in Section 2, followed by some results of our measurement and the problems of TCP in the real CDMA2000-1X networks in section 3. Section 4 presents an overview of our approach. Section 5 will show the results of experiments with our proposed scheme. Finally we conclude this paper in Section 6.

II. Related Works

In this section, we review prior work on improving TCP performance in wireless link. A lot of prior work has focused on the impact of packet loss in the wireless link, which may be misinterpreted as congestion.

To mitigate this problem, Snoop protocol [5] introduces a snoop agent at the base station. The agent monitors every packet that passes through TCP connections in both directions and maintains a cache of TCP segments sent across the link that have not yet been acknowledged by receivers. When a packet loss is detected either by the arrival of acknowledgements (dupacks) from the receiver or by a local timeout, the snoop agent retransmits the lost packet to the receiver if it has the packet cached. During the phase of local recovery, moreover, the snoop agent hides the packet loss from the sender by not propagating dupacks.

Link layer enhancements for reducing wireless link losses including ARQ (Automatic Repeat Request) and FEC (Forward Error Correction) have been proposed in [6]. Link layer approaches are now part of the CDMA2000 standards [3].

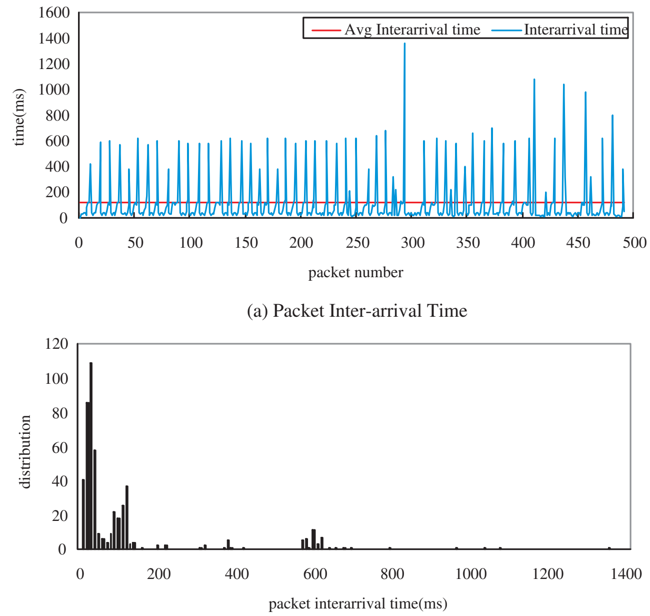
There are approaches that inform the transport layer sender about specific events in wireless link so that it can adapt accordingly. Explicit congestion notification (ECN) [7] and explicit loss notification (ELN) [8] are representative techniques classified as these approaches.

Indirect TCP (I-TCP) [9] breaks the TCP connection at the base station and maintains two separate connections, one over the fixed network and the second over the wireless link. This allows wireless losses to be completely shielded from the wired ones and can recover from losses in wireless link more quickly, resulting in better throughput. However, the end-to-end semantics of TCP is destroyed. M-TCP [10] is similar to I-TCP except it better preserves end-to-end semantics. M-TCP uses a simple zero window advertisement to throttle transmission of data from the wired sender.

WTCP [11] is a pure end-to-end transport layer approach. This protocol tries to distinguish random losses from congestion related losses by measuring the packet inter arrival time with the packet inter departure time. The basic idea behind this protocol is that TCP should not half its transmission rate for just a packet loss which happens more frequently in wireless networks.

Note that these approaches specifically deal with the influence of random packet losses in wireless link. In 3G, however, the problems of a packet loss are mostly settled by a local retransmission mechanism at link layer. Actually we could not observe any incidence of packet loss or packet corruption during our experiment.

Another issue of TCP in wireless link is large delay variation. It may lead to spurious timeouts where TCP sender unnecessarily retransmits a packet and lowers its congestion window to a minimum, when the packet is merely delayed. Eifel Algorithm [12] eliminates the retransmission ambiguity by using TCP timestamp option, thereby solving the problems caused by the spurious timeouts. With timestamp the TCP sender can distinguish whether the acknowledgement is for the original segment or the retransmitted segment.



(a) Packet Inter-arrival Time

(b) Distribution of Packet Inter-arrival Time

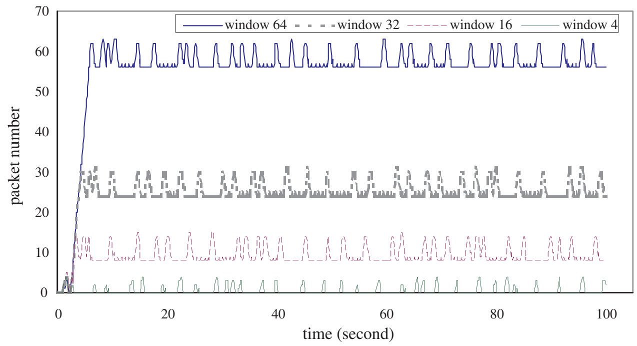
Fig. 1. Packet Inter-arrival Time

Explicit Bad State Notification (EBSN) [13] proposes a mechanism to update the TCP timer at the source to prevent spurious timeouts. EBSN are sent to the source after every unsuccessful transmission of packet over the wireless link by base station. EBSN would cause the previous timeout timer to be cancelled and new timer put in place.

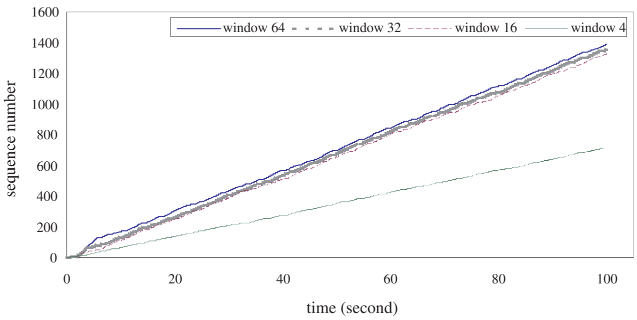
Due to its low bandwidth, the wireless link is almost always the bottleneck of any TCP connection. So, the final issue is the excessive queueing at the gateway, which connect the wired section and wireless section. Higher queueing delays cause the inflation of round trip time (RTT), which result in an inflated retransmit timeout (RTO) value that degrades TCP performance [14]. With large RTO, TCP stalls several seconds if timeout and retransmission are needed due to multiple packet losses in a window. Moreover, because the bottleneck queue is filled with many segments of first TCP connection, any second TCP connection established over the same link is likely to have its initial connection request timeout [15].

III. Measurement results of CDMA2000-1X

Fig. 1 presents a histograms of packet inter arrival time and distribution for five hundred 1000 bytes UPD packets sent with 80,000 bps sending rate. The transmission delays of packets are highly variable. While over 80% of packet inter arrival time is lower than average, some packets experienced very large delay periodically even though there is no movement of mobile



(a) Packet numbers queued at the bottleneck link



(b) Progress of sequence number for each window size

Fig. 2. Characteristics of TCP for several window size

terminal. The main reasons of periodic jitters are periodic checking for incoming call during data communication. Recovery time suspended by RLP with automatic retransmission also contributes this result.

Fig. 2 shows the characteristics of TCP for each window size during FTP file transmission from a fixed host to a mobile host. We measured the sequence number and packets queued at the bottleneck link by subtracting the arrived packet numbers at the mobile host and the packet numbers expected to being transferred from the outstanding packet numbers at a given time. We performed various experiment with varying the window size of the sender, such as 4, 16, 32, and 64. As we can observe, almost of available sending buffer are filled with packets after 5 seconds, and TCP accumulates these packets at the buffer of bottleneck link. Therefore, the TCP has 64 window size could not achieve more throughput than that has smaller window size, such as 16, 32. TCP sender that has larger window size just increases the number of packets queued at the bottleneck link and incurs excessive queueing.

The TCP sender that permitted to send small segments at a time due to the small advertised window experienced many spurious timeout and retransmission because of periodic high jitters. Spurious timeouts degrade the TCP performance by unnecessarily reducing its congestion window to initial value. With bigger advertised window, TCP can overcome these spurious timeout and retransmission because queueing delay at the bottleneck link increases retransmission timeout (RTO) and large RTO alleviates these high jitters. But, large advertised window induces excessive queueing at the bottleneck link,

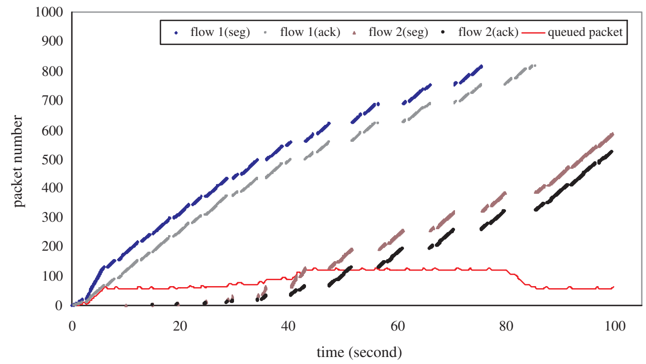


Fig. 3. Time sequence plots of two concurrent flows

such as PDSN.

Excessive queueing can lead to gross unfairness between competing flows as well as a competing TCP connection established over the same link is likely to have its initial connection request timeout. Fig. 3 shows a file transfer flow 2 initiated 10 seconds after transfer flow 1 with TCP has 64 window size, recommended for improving TCP performance in wireless environment by RFC 2757, long thin network. The few initial data packets of flow 2 are queued at the bottleneck link behind a large number of flow 1 packets. As a result, packets of flow 2 bear the full brunt of excess queueing delays due to flow 1. Until flow 1 cancels the operation at 80 seconds, flow 2 cannot achieve enough bandwidth.

IV. Receiver Based Stable and Minimum Queueing(RBSMQ)

There are two objectives in our proposed scheme. One is achieving fully utilization of the link bandwidth, and another is maintains the queue length of bottleneck link stable and minimal. These can support fairness to other concurrent flows. Since we do not assume any a priori knowledge of the receiver's link capacity, the full utilization of the link simply refers to the throughput achievable by the flows when operating under standard TCP. For this purpose, we leverage TCP's flow control mechanism by adjusting the TCP receiver's advertised window.

We measured packet inter-arrival time at the TCP receiver and defined two types of packet inter-arrival time such as *long_time* and *short_time*. We assumed *long_time* is the time suspended by transmission between a TCP sender and a TCP receiver and *short_time* is the time suspended by propagation between a base station and a TCP receiver. A packet arrived within *short_time* boundary can be considered as a queued packet at a base station. We divide *long_time* with *short_time* and use the quotient that we call β as an indicator that the queue of base station is filled sufficiently.

Whenever a new packet is arrived, the TCP receiver determines this packet inter-arrival time is included in *short-time_boundary* or not. If it is included in that boundary, a new *shorttime_base* and a new *shorttime_boundary* must be calculated with this equation,

$$shorttime\_base = \frac{7}{8} \times shorttime\_base + \frac{1}{8} \times interarrival\_time$$

$$s\_variance = \frac{3}{4} \times s\_variance + \frac{1}{4} \times \mid s\_variance - shorttime\_base \mid$$

$$shorttime\_boundary = [\ 0\ ,\ shorttime\_base + s\_variance\ )$$

If the inter-arrival time of a new packet is over *shorttime_boundary*, new *longtime_base* and new *longtime_boundary* can be calculated with similar equation,

$$longtime\_base = \frac{7}{8} \times longtime\_base + \frac{1}{8} \times interarrival\_time$$

$$l\_variance = \frac{3}{4} \times l\_variance + \frac{1}{4} \times \mid l\_variance - longtime\_base \mid$$

$$longtime\_boundary = [\ 0\ ,\ longtime\_base + l\_variance\ )$$

Packets that do not belong to any boundary should not be considered. These packets must be filterd out from new base calculation. When a new connection is established, the *shorttime_base* and the *longtime_base* must be recalculated due to packets for the new connection. The TCP receiver can easily detect about the new connection, because it request or receive the request of a new connection. So the receiver resets the *shorttime_base* and *longtime_base* to the initial value and recalculated these values suitable for new situation.

With the results of these factors, we can choose the value for advertised window. Before the number of continuous arrival of packets with the short time interval reaches predefined factor α, we increase the advertised window by one. If more than β times of packets with the short time interval are arrived continuously, we guess that enormous packets are queued at the base station and decrease the advertised window by one. If the number of continuously arrived packets are between α and β, we preserve the advertised window.

Since we can adjust buffer sizes and change the size of the advertised window in each acknowledgement, we can quickly respond to changes in workloads. We believe our scheme has fewer barriers for acceptance because it requires no modifications to the network, server or application software and requires no support from a service provider.

V. Experiments and Performance Evaluation

A. Test Bed Setup and Experiment

The measurements were all performed over SK Telecom's CDMA2000-1X network in South Korea. We used a Samsung SPH-A500 phone. As we mentioned previously, we modified the TCP/UDP stack of Qualcomm to display the behavior of mobile terminal during data transmission with a host that is laid at a fixed network. The host is equipped with Pentium 4 1.4GHz processors, 256 Mbytes of memory, and 100Mbps ethernet card. It ran Windows XP professional version. We used windump to observe sending and receiving packets at the host. Measurements have been repeated at many different time zones. In all cases, the mobile terminal was stationary though a number of locations were used.

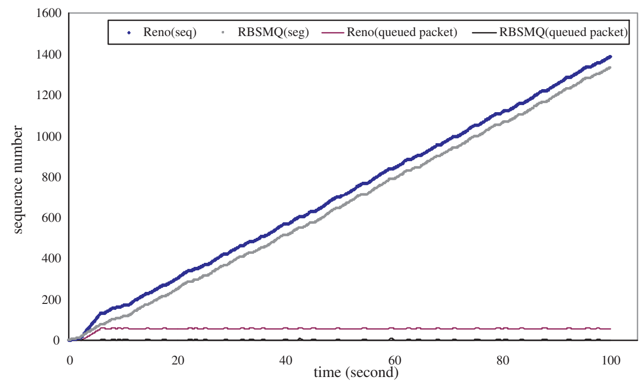


Fig. 4. Time sequence plots of TCP Reno and RBSMQ

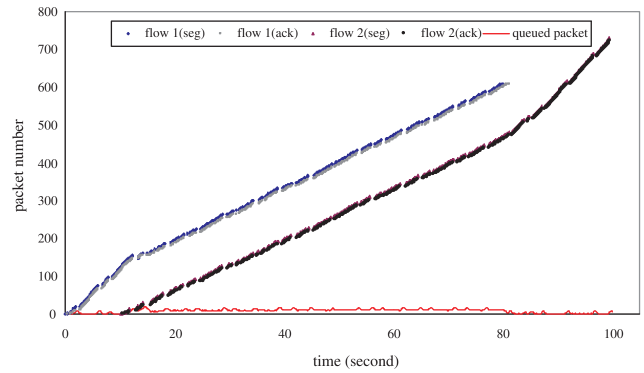


Fig. 5. Time sequence plots of two concurrent flows

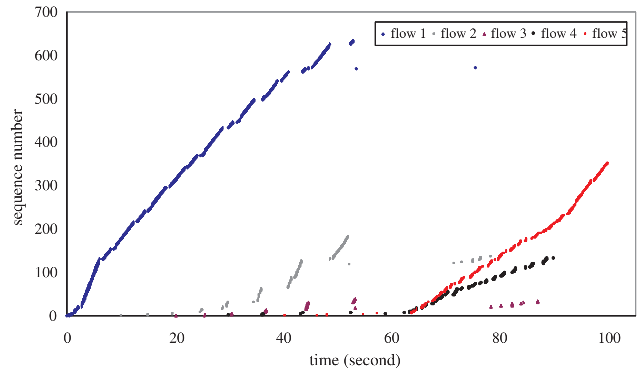
B. Performance Evaluation

Fig. 4 shows time sequence plots of TCP Reno and our modified solution, receiver based stable and minimum queueing(RBSMQ). Our primary goal is to maintain the queue length at bottleneck link can be stable while achieving fully utilization of the link bandwidth without changing the TCP sender. We achieved 96% of throughput compared with TCP Reno that has window size as 64 for one connection. For two concurrent flows, however, we achieve over 100% of throughput compared with TCP Reno and provide same bandwidth between competing flows. Fig. 5 shows the effect of our solution for two concurrent flows at a glance compared with Fig. 3.

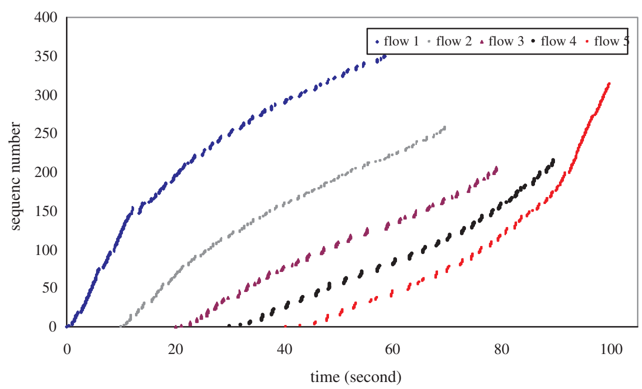
Fig. 6 shows the superiority of our proposed scheme with more concurrent flows. With five concurrent flows, TCP Reno could not support any operation except first connection due to overflow at bottleneck link. However, our receiver based stable and minimum queueing policy can provide fairness between flows and does not incur any packet loss since it maintains the queueing length stable and minimum.

VI. Conclusion and Future Works

We have measured the performance of transport layer protocols on a nationally deployed commercial CDMA2000-



(a) TCP Reno



(b) Receiver Based Stable and Minimum Queueing(RBSMQ)

Fig. 6. Time sequence plots of five concurrent flows

1x network. We used a cellular phone itself to measure the performance through modifications of Qualcomm TCP/UDP stack. With UDP experiments, we could achieve more accurate characteristics of CDMA2000-1x, such as periodic high jitter in transmission time, the buffer size per mobile terminal at PDSN etc. And we confirmed how these features of CDMA2000-1x influence the performance of TCP. Even though we did not experience any packet loss and packet corruption, the wireless link is almost always the bottleneck of any TCP connection and leads to excessive queueing at PDSN with large window size due to the low bandwidth of wireless link. Excessive queueing results in unfair bandwidth allocations between competing connections and induces longer drain time as well as longer recovery time after timeout.

We proposed receiver based stable and minimum queueing policy. Our mechanism maintains the queue length at bottleneck link can be stable while achieving fully utilization of the link bandwidth without changing the TCP sender. It is impractical to change the protocol stacks of all stationary hosts merely to accommodate mobile hosts. With our approach, we can achieve 96% of of throughput compared with TCP Reno that has window size as 64 for one connection. For more than two concurrent flows, moreover, we can achieve over 100% of throughput compared with TCP Reno and provide same bandwidth between competing flows.

References

[1] http://www.3gpp.org

[2] http://www.3gpp2.org

[3] TIA/EIA/cdma2000, "Mobile Station - Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular Systems", Washington: Telecommunication Industry Association, 1999

[4] TIA/EIA/IS-707-A-2.10, "Data Service Options for Spread Spectrum Systems: Radio Link Protocol Type 3," January 2000

[5] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *ACM Wireless Networks*, Vol. 1 December 1995

[6] S. Paul, S. Paul, E. Ayanoglu, T. F. LaPorta, K. H. Chen, K. K. Sabnani, and R. D. Gitlin, "An Asymmetric Link-Layer Protocol for Digital Cellular Communications," in *Proc. of INFOCOM* 1995

[7] R. Ramani and A. Karandikar, "Explicit congestion notification (ECN) in TCP over wireless networks", *Personal Wireless Communications*, 2000

[8] H. Balakrishnan, V. Padmanabhan, S. Seshan and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," in *Proc. of ACM SIGCOMM* 1996

[9] A.V. Bakre and B.R. Badrinath, "I-TCP: indirect TCP for mobile hosts," in *Proc. IEEE ICDCS*, 136-143, May 1995

[10] K. Brown and S. Singh, "M-TCP: TCP for Mobile Cellular Networks," *ACM Computer Communications Review*, Vol. 27 No. 5, October 1997.

[11] P. Sinha, T. Nandagopal, N. Venkitaraman, R. Sivakumar and V. Bharghavan, "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks." in *Proc. of ACM MOBICOM* 1999

[12] R. Ludwig and R. H. Katz, "The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions," *ACM Computer Communications Review*, Vol. 30, No. 1, 2000

[13] B. S. Bakshi, P. Krishna, N. H. Vaidya and D. K. Pradhan, "Improving Performance of TCP over Wireless Networks," *Technical Report* 96-014, Texas A & M University, 1996

[14] D. Dutta and Y. Zhang, "An Active Proxy Based Architecture for TCP in Heterogeneous Variable Bandwidth Networks", in *Proc. of IEEE GLOBECOM*, November, 2001

[15] R. Chakravorty and I. Pratt, "Performance Issues with General Packet Radio Service (GPRS)", *Journal of Communications and Networks (JCN)*, 266-281, Vol. 4, NO. 2, December 2002

[16] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas : New techniques for congestion detection and avoidance", in *Proc. of the ACM SIGCOMM'94*, 24-35, August 1994