

TCP-Taichung: A RTT-based Predictive Bandwidth Based with Optimal Shrink Factor for TCP Congestion Control in Heterogeneous Wired and Wireless Networks

Ben-Jye Chang¹, Shu-Yu Lin¹, and Ying-Hsin Liang²

¹ Department of Computer Science and Information Engineering
Chaoyang University of Technology, Taiwan, ROC.
{changb,s9227623}@mail.cyut.edu.tw

² Department of Computer Science and Information Engineering
Nan-Kai Institute of Technology, Taiwan, ROC.
t136@nkc.edu.tw

Abstract. TCP congestion control works well in wired Internet network but it is difficult to determine a good congestion window in a heterogeneous wireless network that consists of the wired Internet and various types of wireless networks. Therefore, we propose herein a novel adaptive window congestion control, namely TCP-Taichung, for TCP connections in heterogeneous wireless networks. The proposed RTT-based congestion control is designed for the Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery phases to increase throughput while supporting high fairness. In addition, an optimal shrink factor is proposed to determine the optimal *cwnd* and *ssthresh* for the cases of network congestion or wireless-link error. Numerical results demonstrate that TCP-Taichung outperforms other approaches in goodput and fairness under various wireless network topologies. Especially, in the case of 10% packet loss rate in wireless link, the proposed approach increases goodput up to 200% as compared with NewReno.

Keyword:TCP, RTT, congestion control, error wireless link, heterogeneous wireless networks

1 Introduction

ALL IP based access for Internet rich resources anytime anywhere makes different wireless network standards and technologies have been developed progressively. Several successful IP-based wireless standards include IEEE 802.11 family [1] and 802.16 [2] for wireless networks, and 3G [3]/4G [4] for mobile communications. The important protocols of all IP based wireless networks are IP [5] and TCP [6] that provides reliable end-to-end connections in transport layer and connectionless-oriented hop-by-hop routing in network layer, respectively. Although TCP works well in the wired Internet, it is hard to determine a good

congestion window ($cwnd$) in wireless networks. In addition, TCP congestion control [7] can be divided into four phases, including the slow start, Congestion Avoidance, Fast Retransmit and fast recovery phases. In [8], Hoe proposed a bandwidth delay product based computation for determining *initial ssthresh*, in which the first three ACKs are required for sender to determine it. The main ideas of these mechanisms are based on sliding window and additive increase multiplicative decrease (AIMD) [9] algorithms. We can classify these TCP congestion control mechanisms into two types: the AIMD-based mechanism and the estimated available bandwidth mechanism.

First, several AIMD-based algorithms [10–13] have been proposed. TCP-Tahoe [12] TCP-Reno [13] and TCP-NewReno [10] are typical AIMD-based algorithms. [14, 15] proposed an additive increase adaptive decrease (AIADD) mechanism to improve such a problem by estimating available bandwidth. Second, several approaches of estimated available bandwidth have been proposed [14–19]. TCP-Westwood [14, 15] is proposed to estimate available network bandwidth by using the ACK reception rate at sender. Although TCP-Westwood improves throughput, it suffers from inaccurate estimation of available bandwidth. Then, [17, 18] proposed TCP-Westwood+ for estimating network available bandwidth more accurate and then increases throughput. Based on TCP-Westwood, TCP-Jersey [16] was proposed to provide the Explicit Congestion Notification (ECN) in a router’s Active Queue Management (QAM). [20] proposed TCP-Vegas to determine whether a network occurs congestion or not by using the difference of the Expected (i.e., $cwnd/RTTmin$) and Actual (i.e., $cwnd/RTT$) rates. Unfortunately, the optimal lower and upper bounds of α and β are difficult to determine in TCP-Vegas.

Although many approaches have been proposed to support or improve congestion control in TCP, it is difficult to provide high throughput while providing good fairness and friendliness in heterogeneous wireless networks. Therefore, in this paper, we propose an adaptive RTT-based $cwnd$ control with a precise shrink factor approach to achieve high throughput while providing good fairness.

The remainder of the paper is organized as follows. Section II describes the network model of TCP connections and performance metrics for evaluation. Section III then describes the TCP-Taichung approach for controlling congestion window in TCP under heterogeneous wireless networks. Section IV evaluates the performance of the proposed RTT-based approach. Section V draws several important conclusions.

2 Network Model

We model a TCP/IP network as a logical graph, $G = (V, E)$, which consists of a set of nodes, V , and a set of logical end-to-end connections, E . Two end types are considered in a TCP/IP connection, including the TCP connection server (V_s) and client (V_c). Fig. 1 demonstrates an example of four TCP connections, including $E_{V_s^2, V_c^7}^1$, $E_{V_s^1, V_c^5}^2$, $E_{V_s^4, V_c^3}^3$, and $E_{V_s^8, V_c^6}^4$ in heterogeneous wireless net-

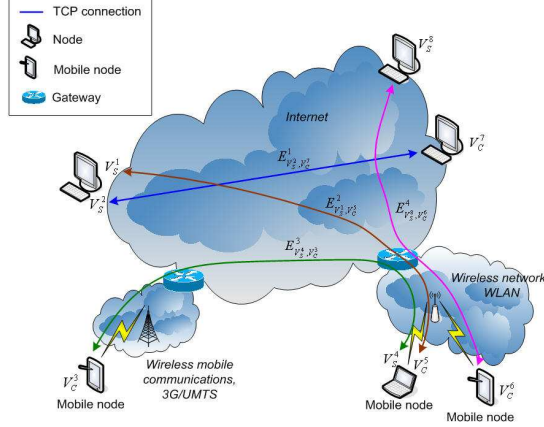


Fig. 1. An example of TCP connections in heterogeneous wireless networks

works, in which connections $E_{V_s^1, V_c^5}^2$ and $E_{V_s^8, V_c^6}^4$ have a single wireless link and connection $E_{V_s^4, V_c^3}^3$ has two wireless links.

Various performance metrics, including goodput and fairness, are adopted to evaluate the TCP-Taichung approach and other approaches under various network topologies with different error rate of wireless links. First, since packets transmission from sender may be lost or discarded due to network congestion and error wireless links, not all transmitted packets will arrive at receiver. Therefore, we adopt average goodput to evaluate the congestion control mechanisms. Average goodput is defined by

$$Average\ Goodput = \frac{\sum_{n=1}^N \frac{MaxACKSeqNo_n \times MSS_n}{TotalTransmissionTime}}{N}, \quad (1)$$

where $MaxACKSeqNo_n$ denotes the maximum ACK number of connection n , MSS_n is the maximum segment size of connection n , and N is the total number of connections. Second, fairness is another important metric for evaluating whether all connections fairly utilize the bandwidth of the network or not. A good TCP congestion control approach should achieve a fair usage of bandwidth among all connections. The fairness metric is defined by

$$Fairness = \frac{\left(\sum_{n=1}^N S_n \right)^2}{N \times \sum_{n=1}^N (S_n)^2}, \quad (2)$$

where S_n is the throughput of connection n . The range of *Fairness* is between one and $1/N$, where *Fairness* = 1 is the most fair result.

3 Adaptive Network Bandwidth Approach

This section first briefly describes the problems of congestion control in TCP connections under heterogeneous wireless networks. Then, we detail the motivations of the proposed TCP-Taichung approach. Finally, the optimal congestion control is described in two aspects of the network bandwidth is sufficient or not. First aspect proposes two adaptive control mechanisms to determine the *optimal cwnd*, $cwnd_{opt}$, and the ideal slow start threshold, $ssthresh_{ideal}$, in the increasing *cwnd* phase while the expected bandwidth is enough. Second aspect provides the determinations of the *optimal cwnd*, $cwnd_{opt}$, and the *network ssthresh*, $ssthresh_{net}$, in the decreasing *cwnd* phase while the expected network bandwidth is insufficient.

3.1 In the increasing *cwnd* phase while the expected network bandwidth is enough

The algorithm of determining the optimal congestion window ($cwnd_{opt}$)

Assume that MSS is S bits and the data rate of is R bit per second and the minimum round-trip time of TCP connections is RTT_{min} when the network traffic is not saturation. The total time of transmitting a packet in such a light-traffic network is thus $(S/R) + RTT_{min}$ (sec). In addition, in such a light-traffic network, the increment of congestion window (*cwnd*) of sender will not affect the round trip time and the total time of transmitting *cwnd* packets should be less than $(S/R) + RTT_{min}$, as shown below

$$cwnd \cdot (S/R) < (S/R) + RTT_{min}, \quad (3)$$

that is,

$$cwnd < \frac{RTT_{min}}{S/R} + 1. \quad (4)$$

Based on the analysis, we obtain the round trip time (*RTT*) of a connection, which is close to RTT_{min} , if *cwnd* is less than $\frac{RTT_{min}}{S/R} + 1$. On the other hand, if $cwnd \geq \frac{RTT_{min}}{S/R} + 1 = k$, the network traffic becomes saturation and *RTT* will be increased based on $(cwnd - k) \cdot S/R$ till congestion. As a result, the value of $\frac{RTT_{min}}{S/R} + 1$ is the optimal window size of the connection, which is denoted as $cwnd_{opt}$. In other words, when *cwnd* is larger than $cwnd_{opt}$, the network will enter into the saturation state.

The algorithm of determining the ideal ssthresh ($ssthresh_{ideal}$) In TCP, the Slow Start threshold, *ssthresh*, is an important parameter for a sender to determine when to enter into the Congestion Avoidance phase. If *ssthresh* is

set too small, the sender will enter into the Congestion Avoidance phase early that brings two disadvantages: low goodput and unfairness. On the other hand, if $ssthresh$ is set too large, the sender will send too many packets in a congested network, which will cause the congested network more serious. Since the congested intermediate router may discard the over-sent packets, this causes too many duplicate ACKs or timeout. In the worst case, sender will enter into the Slow Start phase. $Cwnd$ will be set to one and result in low goodput. Therefore, the determination of an *ideal ssthresh*, i.e., $ssthresh_{ideal}$, is proposed in this paper for achieving high goodput and high fairness.

Since $ssthresh$ is affected by traffic load, an ACK-based mechanism is proposed to determine the $ssthresh_{ideal}$ adaptively. We first define the received inter-ACK period of two continuous ACKs at sender as $InterACK$ that is a parameter represented current traffic load. Then, the new expected smooth $InterACK$ is denoted as $InterACK_{new}$, which is computed on line based on the Exponential Weight Moving Average (EWMA) model,

$$InterACK_{new} = \alpha \cdot InterACK_{avg} + (1 - \alpha) \cdot InterACK_{old}, \quad (5)$$

where $InterACK_{avg}$ denotes the average inter-ACK time of the connection, $InterACK_{old}$ is the inter-ACK time between current and previous ACKs, and α is a constant value ranging of $(0, 1)$.

Based on receiving each new ACK packet, sender can adaptively determines the expected smooth $InterACK$. Consequently, the *ideal ssthresh* of the Slow Start phase can be formulated by

$$ssthresh_{ideal} = \frac{RTT_{min}}{InterACK} + 1, \quad (6)$$

which is also the *optimal ssthresh* of the first $ssthresh$ in the Slow Start phase.

The determination of adaptive network bandwidth and network ssthresh

After obtaining the *ideal ssthresh*, the adaptive network bandwidth can be estimated by

$$Adaptive\ network\ bandwidth = \frac{ssthresh_{ideal}}{RTT}, \quad (7)$$

where RTT is the round-trip time of the last received ACK. Under the estimated bandwidth, $cwnd$ will increase exponentially until $cwnd$ exceeds the *network ssthresh*, $ssthresh_{net}$, in the Slow Start phase and then enters into the Congestion Avoidance phase. As the *ideal ssthresh* analysis in previous subsection, the *network ssthresh* of the expected available bandwidth is an important threshold that should be addressed for achieving high goodput and fairness. Therefore, we first determine the *network ssthresh* by

$$ssthresh_{net} = (Adaptive\ network\ bandwidth) \cdot (RTT_{min}). \quad (8)$$

That means the *idela ssthresh* of the expected adaptive network bandwidth will occur at the situation of RTT_{min} . In consequence, after applying (7) to (8), we

thus have

$$\begin{aligned} ssthresh_{net} &= (\text{Adaptive network bandwidth}) \cdot (RTT_{min}) \\ &= \frac{ssthresh_{ideal}}{RTT} \cdot RTT_{min}, \end{aligned} \quad (9)$$

or

$$ssthresh_{net} = \left(\frac{RTT_{min}}{InterACK} + 1 \right) \cdot \frac{RTT_{min}}{RTT}, \quad (10)$$

where the term $\frac{RTT_{min}}{RTT}$ is called a shrink factor of the *ideal ssthresh*. Since network bandwidth is shared among all connections, RTT of an adaptive network bandwidth should be less than RTT_{min} . The *network ssthresh* can be formulated by shrinking the *ideal ssthresh* based on the shrink factor. Consequently, when $cwnd \geq ssthresh_{net}$, sender will enter into the Congestion Avoidance phase. In addition, we consider the value of *ssthresh* in the Congestion Avoidance phase, which is also determined based on the shrink factor of *ssthresh* when three duplicate ACKs received or timeout occurs.

Based on the analysis of $cwnd_{opt}$, $ssthresh_{ideal}$ and $ssthresh_{net}$, the algorithm of controlling congestion window while receiving ACKs at sender is demonstrated in Fig. 2. There are two significant contributions. First, the expected smooth *InterACK* is computed based on the EWMA model for achieving accurate estimation of *InterACK*. Second, the condition of switching from the Slow Start phase to the Congestion Avoidance phase is based on two thresholds of $ssthresh_{net}$ and $ssthresh_{ideal}$ for accurately predicting when will occur network saturation. The variation of congestion window of receiving ACK in Slow Start of TCP-Taichung is compared with NewReno. In Fig. 3, we can observe that TCP-Taichung avoids occurring timeout in the first Slow Start phase and results in higher goodput than that of NewReno, in which the goodput of the proposed approach is 8779980 bps but that of NewReno is 8177664 bps. The accurate analysis of *ideal ssthresh* and the optimal congestion window in Slow Start results in high goodput.

3.2 In the decreasing *cwnd* phase while the expected network bandwidth is insufficient

When $cwnd$ exceeds $ssthresh_{net}$ and $ssthresh_{ideal}$, sender enters into the Congestion Avoidance phase and $cwnd$ increases linearly while receiving each ACK. In this phase, the available network bandwidth becomes worse and buffers of intermediate routers will stuff with unsent packets and increase the RTT. If these delayed packets exceed the receiving timeout, receiver will send a duplicate ACK request back to sender. When sender receives three duplicate ACKs that means network congestion becomes worse, $cwnd$ should be reduced for routers to process these queued packets, i.e., the flight packets. In Congestion Avoidance, most of TCP congestion control approaches significantly reduce $cwnd$, if sender receives three duplicate ACKs. For instance, $cwnd$ is reduced to one in Tahoe and

```

ReceiveACK()
{
  if  $\left( \frac{InterACK}{InterACK_{min}} < cwnd \parallel InterACK_{org} > InterACK \right)$  {
    // Filter InterACKdiff
     $InterACK = (\alpha) * InterACK_{org} + (1 - \alpha)InterACK$ 
  }
   $ssthresh_{net} = \left( \frac{RTT_{min}}{InterACK} + 1 \right) + \left( \frac{RTT_{min}}{RTT} \right)$ 
  if  $(cwnd < ssthresh_{net} \ \&\& \ cwnd < ssthresh_{ideal})$  {
     $cwnd++$  //exponential increasing phase
  }
  else {
     $cwnd+ = \frac{1}{cwnd}$  //linear increasing phase
  }
}

```

Fig. 2. The algorithm of controlling congestion window while receiving ACKs at sender

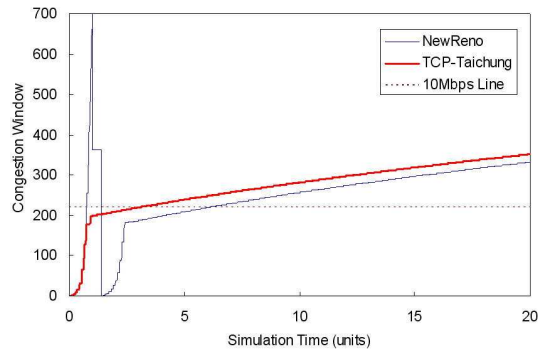


Fig. 3. The variation of congestion window in Slow Start

becomes half of original $cwnd$ in Reno and NewReno. Since these $cwnd$ reducing mechanisms are unsystematic and are independent on the available network bandwidth, which results in large variation of $cwnd$ and low throughput.

Therefore, an expected network bandwidth based algorithm is proposed herein for reducing $cwnd$ adaptively. In the case of receiving three duplicate ACKs in Congestion Avoidance, the network will become saturation and cause congestion seriously. For changing the network state from saturation to non-saturation, $cwnd$ is set to the optimal $cwnd$ that is the window size of the non-saturation state while with the same expected actual rate. In other words, the optimal $cwnd$ is at the threshold between the saturation and non-saturation states. The

optimal $cwnd$ is computed by

$$Expected\ actural\ rate = \frac{cwnd}{RTT} = \frac{cwnd_{opt}}{RTT_{min}}, \quad (11)$$

that is,

$$Optimal\ cwnd(or\ cwnd_{opt}) = cwnd \cdot \frac{RTT_{min}}{RTT}. \quad (12)$$

Finally, the algorithm of reducing congestion window by the shrink factor while receiving three duplicate ACKs at sender is shown in Fig. 4.

```

if(three DUPACKs are received) {
    ssthreshnet = cwnd · (  $\frac{RTT_{min}}{RTT}$  )
    cwndopt = ssthreshnet
}
if(timeout expired) {
    ssthreshnet = cwnd · (  $\frac{RTT_{min}}{RTT}$  )
    cwnd = 1
}

```

Fig. 4. The variation of congestion window in Slow Start

4 Numerical Results

This section evaluates the proposed TCP-Taichung approach for TCP congestion control in heterogeneous wireless networks by comparing various performance metrics, including goodput and fairness. Several compared approaches includes NewReno [10] and Westwood+ [17, 18]. The Network Simulator (NS-2) [21] is adopted for all simulations with the following network parameters. The packet size is 512 bits. Router buffer is determined by Bandwidth-Delay Product (BDP) rule-of-thumb [22]. Impatient variant is adopted for responding partial ACKs for all TCP connections. In wireless networks, the uniform packet loss model is adopted at the sender and receiver nodes.

4.1 Scenario 1: N TCP connections in a wired network

In scenario 1, total n (i.e., $n = 1, 2$ and 10) TCP connections are operated in a wired network with a bottleneck link, which bandwidth is 10 Mbps and delay is 25 ms. The other links except the bottleneck link are with the bandwidth of 100 Mbps and delay of 10ms as shown in Fig. 5. Two-type of simulations are evaluated including the cases without CBR and with CBR background traffic.

Without CBR background traffic, the average goodput and fairness of TCP-Taichung, NewReno, and Westwood+ are demonstrated in Tables 1-2. In Table

1, TCP-Taichung yields the highest goodput under the number of connections is 1, 2, or 10. Meanwhile, Westwood+ yields the worst goodput among all compared approaches under different number of connections. In Table 2, all approaches yield very competitive fairness, in which the results are very close to one. $Fairness = 1$ means the fairest result. In Table 3, TCP-Taichung outperforms NewReno and Westwood+ in average goodput, in which Westwood+ yields the worst goodput. In addition, the goodput of all approaches decreases as the number of connections increases.

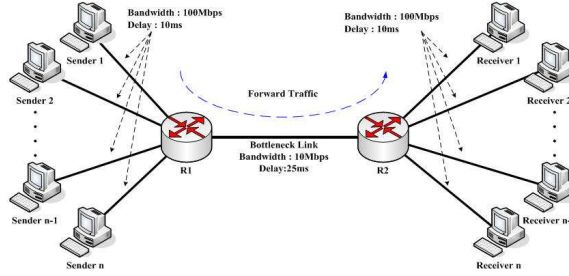


Fig. 5. Wired network with a single bottleneck link

Table 1. Goodput of compared approaches without CBR traffic (Wired network)

Number of connections	Goodput (bps)		
	TCP-Taichung	NewReno	Westwood+
1	9043681	8885084	8575426
2	4527145	4454523	4371722
10	901849	896581	893390

Table 2. Fairness of compared approaches without CBR traffic (Wired network)

Number of connections	Fairness		
	TCP-Taichung	NewReno	Westwood+
2	0.999965	0.99993674	0.99992643
10	0.999403	0.99953054	0.99726527

Table 3. Goodput of compared approaches with CBR traffic (Wired network)

Number of connections	Goodput (bps)		
	TCP-Taichung	NewReno	Westwood+
1	4286341	3978199	4164608
2	2174054	2120765	2171043
10	451027	450265	447422

4.2 Scenario 2: N TCP connections in a wired network with a single wireless link

For evaluating goodput and fairness affected by network congestion and wireless error link, scenario 2 is simulated for all compared approaches. In Fig. 6, total 20 TCP connections are operated in a wired network with a single wireless network, in which bandwidth and delay of the wired bottleneck link is 10 Mbps and 25 ms, respectively. The packet loss rate of wireless links is between 0 and 0.1.

In Fig. 7, goodput of all approaches decreases as the wireless packet loss rate increases. The proposed approach yields the best goodput under all different wireless packet loss rate, but NewReno yields the worst goodput; especially, the goodput of the proposed approach is 210% higher than that of NewReno and 175% higher than that of Westwood+. Furthermore, fairness of all approaches under different wireless packet loss rate are evaluated as shown in Table 4. The proposed approach yields very competitive fairness to that of NewReno. Fairnesses of the proposed approach of different wireless packet loss rates are all above 0.9883.

In summary, the proposed approach results in the highest goodput not only in the wired network but also in the heterogeneous wired and wireless network. However, goodput of NewReno is better than that of Westwood+ under a wired network with CBR background traffic, but Westwood+ outperforms NewReno in goodput. Furthermore, all approaches yield very competitive fairness. In the

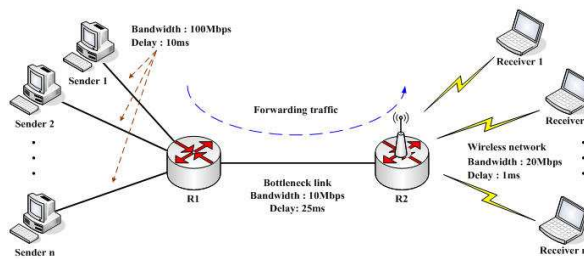


Fig. 6. 20 TCP connections in a wired network with a single wireless link

wireless network, the *cwnd* variation of the proposed approach is more stable than that of NewReno.

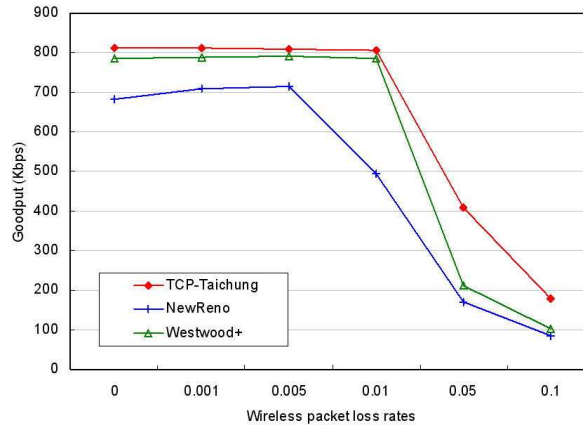


Fig. 7. Goodput of compared approaches (with a single wireless network)

Table 4. Fairness of compared approaches (with a single wireless network)

Packet loss rate	Fairness		
	TCP-Taichung	NewReno	Westwood+
0	0.988328	0.999261	0.923017
0.001	0.993481	0.996917	0.973306
0.005	0.992687	0.996184	0.991795
0.01	0.996392	0.99666	0.995311
0.05	0.994997	0.99684	0.996087
0.1	0.988709	0.984752	0.996548

5 Conclusions

In this paper, we proposed a RTT-based predictive bandwidth approach, namely TCP-Taichung, to determine the optimal congestion window, network *ssthresh*, and ideal *ssthresh* for TCP congestion control in heterogeneous wireless networks. Numerical results show that TCP-Taichung outperforms NewReno and Westwood+ in goodput and fairness in both of wired network and heterogeneous wireless networks.

Acknowledgments

This work was funded in part by National Science Council, Taiwan, ROC, under Grant NSC 95-2221-E-324-002 for B.-J. Chang.

References

1. IEEE 802.11 WG, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE std. 802.11*, 1999.
2. "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems," *IEEE Std. 802.16*, Oct. 2004.
3. <http://www.3gpp.org>
4. K.R. Santhi, V.K. Srivastava, G. SenthilKumaran and A. Butare, "Goals of true broad band's wireless next wave (4G-5G)," *IEEE VTC 2003-Fall*, Vol. 4, pp. 2317-2321, Oct. 2003.
5. "Internet Protocol," *IETF RFC 791*, 1981.
6. "Transmission Control Protocol," *IETF RFC 793*, 1981.
7. "Transmission Control Protocol," *IETF RFC 2581*, 1999.
8. Janey C. Hoe, "Improving the start-up behavior of a congestion control scheme for TCP," *ACM SIGCOMM 1996*, pp. 270-280, Aug. 1996.
9. D. Chiu and R. Jain, "Analysis of the increase/decrease algorithms for congestion avoidance in computer networks," *Journal of Computer Networks and ISDN systems*, vol.17, no. 1, pp. 1-14, June 1989.
10. "The Newreno Modification to TCP's Fast Recovery Algorithm," *IETF RFC 2582*, 1999.
11. "The Newreno Modification to TCP's Fast Recovery Algorithm," *IETF RFC 3782*, 2004.
12. V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM 1988*, pp. 314-329, Aug. 1988.
13. W. Stevens, "TCP slow start, congestion avoidance, fast retransmit and fast recovery algorithms," *IETF RFC 2001*, 1997.
14. C. Casetti, M. Gerla, S. Mascolo, M.Y. Sanadidi and R. Wang, "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links," *ACM Mobicom 2001*, pp. 287-297, July 2001.
15. R. Wang, M. Valla, M.Y. Sanadidi, B.K.F. Ng and M. Gerla, "Efficiency/friendliness tradeoffs in TCP Westwood," *IEEE ISCC 2002*, pp. 304-311, July 2002.
16. Xu Kai, Tian Ye and N. Ansari, "TCP-Jersey for wireless IP communications," *IEEE JSAIC*, Vol. 22, Issue 4, pp. 747-756, May 2004.
17. L.A. Grieco and S. Mascolo, "Performance evaluation of Westwood+ TCP over WLANs with local error control," *IEEE LCN2003*, pp. 440-448, Oct. 2003.
18. <http://www-ictserv.poliba.it/mascolo/tcp%20westwood/modules.htm>
19. L.S. Brakmo, S.W. O'Malley and L.L. Peterson, "TCP Vegas: new techniques for congestion detection and avoidance," *ACM SIGCOMM*, pp. 24-35, 1994.
20. J. Mo, R.J. La, V. Anantharam and J. Walrand, "Analysis and comparison of TCP Reno and Vegas," *IEEE INFOCOM '99*, Vol. 3, pp. 1556-1563, Mar. 1999.
21. ns-2 Network Simulator, <http://www.isi.edu/nsnam/ns/>
22. C. Villamizar and C. Song, "High Performance TCP in the ANSNET," *ACM SIGCOMM Computer Communication Review*, v.24, no.5, pp.45-60, Nov. 1994.