A TCP-Aware Call Admission Control Scheme for Packet-Switched Wireless Networks

Xinbing Wang, Do Young Eun, Wenye Wang
Department of Electrical and Computer Engineering
North Carolina State University
Email: {xwang8, dyeun, wwang}@ncsu.edu

Abstract

Traditional Call Admission Control (CAC) schemes only consider call-level performance and are believed to be sufficient for the circuit-switched wireless network. Since the future wireless network will become packet-switched, the packet-level performance should not be ignored. This is especially true when the TCP-type of applications are running over such packet-switched wireless networks, because TCP congestion control algorithm will exhaust all the available resource until packet loss occurs. In order to regulate the TCP applications to friendly coexist with other types of services, we propose a TCP-aware CAC scheme. We analyze the system performance under the scenario that call holding time is independent of the system state. Simulations results for different performance metrics are presented to show that the proposed scheme can effectively improve the system performance in terms of call blocking probability, call-level throughput (call/min) and the link utilization.

1 Introduction

Currently, TCP [1] is responsible for carrying more than 90% of data and 80% applications generated in the Internet. Due to the enormous success of TCP in the wired network [2, 3, 4], it is assumed to be responsible for carryiong data transmission over a wireless environment and extensive research has been conducted to make the TCP transmission over wireless networks feasible. Examples include ATCP [5], M-TCP [6], I-TCP [7], among others.

A key feature of the TCP congestion control is so called the Additive-Increase-Multiple-Decrease (AIMD) algorithm. If there is no packet loss or ECN mark within the current round trip time (RTT), the window size is increased by one during the next RTT; Otherwise, the window size is halved. Even though the existing literature regarding TCP over wireless varies in different aspects, they share a common TCP congestion control algorithm, i.e., AIMD, which determines the packet-level TCP dynamics and the associated performance for a packet-based network.

Since the packet-switched technology yields more efficient utilization of the scarce wireless resource than that of circuit-switched networks, the future 3G and 4G wireless network will be packet-based networks [8, 9] and this is especially true for 4G system. (We note that 3G is a circuit and packet switched network, but 4G will be a pure packet-switched network.) In a packet-switched wireless network, the packet-level dynamics will affect the call-level performance as we will illustrate later on, and thus both the packet-level and the call-level have to be considered by Call Admission Control (CAC).

Traditionally, CAC schemes in wireless networks deal only with call-level performance metrics: call blocking probability and handoff dropping probability [10, 11, 12, 13]. Recently, CAC schemes in [14, 15, 16] take packet-level performance into account. The authors propose an efficient CAC scheme for heterogeneous services in wireless ATM networks in [15, 16]. The packet-level constraints such as delay and jitter are considered to guarantee predefined QoS. A higher priority is assigned to handoff calls to improve the handoff dropping probability by reserving certain amount of bandwidth for potential handoff calls, and such guard-channel type of idea is widely applied to the CAC field as in [13]. Their simulation results indicate that the proposed CAC scheme can achieve both packet-level and call-level performance in terms of link utilization and handoff dropping probability.

However, none of these schemes has considered the effect of TCP applications on the packet-level performance. As TCP will be responsible to carry data over the future packet-switched wireless network, it would be important to investigate the effect of TCP algorithm on CAC. Intuitively, the AIMD algorithm makes the TCP call keep increasing the rate injecting to the

1-4244-0198-4/06/$20.00

network until there is a packet loss due to the limited capacity. There are two implications by such an AIMD algorithm. First, TCP call based on AIMD algorithm will exhaust all the available system capacity and thus the system will block/reject any subsequent call requests coming in, degrading the call-level performance (call blocking probability and handoff dropping probability). Second, once there is a packet loss, the TCP call will decrease the sending rate by half, which could be far below the system capacity, and hence lead to poor packet-level performance (i.e., low link utilization).

In order to overcome these problems for TCP algorithm over a packet-switched wireless network, in this paper, we first layout a two-level framework to allow us investigate CAC over the future packet-switched wireless networks. Based on the framework, we then propose a TCP-aware CAC scheme that takes the effect of AIMD algorithm into consideration. The idea is that the system will intentionally control the packet loss rate of TCP calls based on the current system status. In particular, if the system is under light load (i.e., the number of calls in the system is small), the system will choose smaller packet loss probability of TCP calls to achieve higher average TCP throughput. (TCP throughput is roughly inversely proportional to the square root of the packet loss probability.) In contrast, if the system is under heavy load (i.e., the number of calls in the system is large and the system capacity is nearly fully-utilized), the system will impose higher packet loss probability to reduce the throughput of the ongoing TCP calls, and this allows the system to accommodate more call requests to reduce the blocking probability. Our results show that by dynamically controlling the packet loss probability of the TCP calls, we can effectively regulate the system to yield the desirable performance metrics of both the call level (e.g., smaller call blocking probability and higher call-level throughput (call/min)), and the packet level (e.g., higher link utilization).

2 Preliminary: TCP AIMD Algorithm

The main feature of TCP congestion control algorithm is characterized by TCP AIMD [1]. Specifically, assume that the window size of a TCP source in the current round trip time (RTT) is W. If all the W packets are transmitted successfully to the destination, the TCP source will inject $W+1$ packets in the next RTT. If at least one of the W packets is lost, the TCP source will reduce the window size to $W/2$. In this paper we assume that packet loss due to the wireless link is handled by physical or MAC layer such that TCP source can distinguish whether the loss is due to congestion (i.e., buffer overflow) or wireless link. Without congestion signal, the TCP sources will keep increasing the rate injecting to the network until packet loss occurs. In other words, the capacity of a base station (BS) will be fulfilled sooner or later and packet arrivals when the buffer is full will be dropped by the BS and become the congestion signal for the TCP sources to reduce their sending rates. In this situation, the average throughput of a TCP call can be expressed by the well-known formula [17]

$$\mathbb{E}\{TCP\} \approx \frac{1}{RTT}\sqrt{\frac{3}{2p(t)}}, \tag{1}$$

where $p(t)$ is the packet loss probability.

3 System Model

We study a packet-switched wireless network supporting multiple classes of services as in [18, 19]. Unlike traditional CAC scheme for wireless networks that only takes call-level performance metrics into account, we look at a system with two-level considerations: call-level performance and packet-level performance.

At call-level, the BS will allocate radio resource to the call request–either new call or handoff call request. Each class of calls is characterized by call arrival rate, and service holding time, etc., while the packet-level features are characterized by the QoS profile that describes the packet arrival rate, packet service rate and packet loss requirement. Different classes may have different packet loss ratio. For example, the voice traffic requires much smaller loss ratio than data traffic does, and in practice, the real time traffic has more stringent loss requirement than the non-realtime traffic.

The interaction between call level and packet level is the following. If a BS accommodates more call requests into the system and each request has certain amount of data to be transmitted, the overall system load in terms of packets (average packet arrival rate) will increase, which becomes the 'action' of the call level on the packet level. Hence, at the packet level, the system with fixed capacity (bits/sec) is more likely to be overwhelmed by the increase in the number of calls. In other words, the probability of packets dropped due to buffer overflow is increased. Since each type of service has certain QoS constraints (e.g. the packet loss probability being less than some predefined value), in order not to violate the QoS agreement with the ongoing users, the system would control the admission of the arrival request and this becomes the 'reaction' of the packet level to the call level.

In order to quantify the system behavior, we summarize system parameters in Table 1 that will be used in the rest of the paper.

Table 1: System parameters: call level and packet level.

Call Level		
	K	number of classes, $k = 1, 2, ..., K$
	Λ_k	call arrival rate of class k (call/min)
	Ψ_k	call service rate of class k (call/min)
	P_b^k	call blocking probability of class k
Packet Level		
	C	system capacity (bits/sec)
	$\tilde{C}$	QoS guaranteed capacity
	$\overline{p}$	QoS guaranteed packet loss prob.
	p_T^u	upper bound of TCP packet loss prob.
	p_T^l	lower bound of TCP packet loss prob.
	$p(t)$	packet loss prob. of TCP call
	B	buffer size (pkt)
	λ_k	packet arrival rate of class k (pkt/sec)
	μ_k	packet service rate of class k (pkt/sec)

4 TCP-Aware CAC Scheme

In this section, we present our TCP-aware CAC scheme and describe how the packet-level dynamics affects the admission control at call level. Generally, the QoS profile for each class of calls is fixed. For example, the packet arrival rate of voice traffic is fixed and the packet loss probability of voice traffic should be less than a certain fixed threshold. If the system serves only for those classes of calls with fixed QoS profile, the system does not have any flexibility to deal with (either accept it with guaranteed QoS, or reject it). However, for a TCP call, the QoS requirement is not fixed, but falls into a wide range, because TCP is a self-adaptation protocol. If there is more bandwidth available, TCP will increase the transmission rate. If there is little bandwidth left in the system and congestion happens, TCP will reduce the transmission rate to fit the available bandwidth. Thus, TCP itself is capable of coping with a wide range of situations and this feature allows the BS to have the flexibility to control the TCP calls. For example, if there is a new call request and the system has no enough resource (i.e., the total packet arrival rate is near the threshold) to admit this call directly, instead of rejecting the call immediately, the system will try to reduce the transmission rate of TCP call by controlling the packet loss probability, $p(t)$ (cf. (1)) of TCP calls to make some room for the new call request. This increases the chance of accommodating the new call request and decreases the call blocking probability. Next, we will illustrate how our TCP-aware CAC works in detail.

Algorithm 1 TCP Aware CAC: New Call Admission.

Require: Base Station (BS) status
A new call request 'm' arrives
if BS has enough resource **then**
  Accept 'm'
else
  "TCP Call Adjustment" /* adjust TCP calls */
  if BS has enough resource after TCP adjustment **then**
    Accept 'm'
  else
    Reject 'm'
  end if
end if

Algorithm 2 TCP Aware CAC: TCP Call Adjustment.

Require: $p(t)$, p_T^U, $\tilde{C}$, QoS profile of call 'm', E_m is the average transmission rate of call 'm'.
$n_1 \leftarrow$ the number of TCP call in the system
$\lambda \leftarrow$ the total packet arrival rate
for i = 1 to n_1 **do**
  /* reduce TCP call's transmission rate */ $\Delta\lambda \leftarrow \frac{1}{RTT}\sqrt{\frac{3}{2p(t)}} - \frac{1}{RTT}\sqrt{\frac{3}{2p_T^U}}$
  $\lambda \leftarrow \lambda - \Delta\lambda$
  if $\lambda + E_m \leq \tilde{C}$ **then**
    Accept 'm'
    Return "True"
  end if
end for
Reject 'm'
Return "False"

When there is a call request arrival at the system, if it is a handoff call, the BS will accept the call regardless of what class it belongs to. The system will not suffer the handoff dropping problem at call level at the expense of increasing likelihood of packet loss at packet level [1]. If the call request is a new call, the BS will examine whether there are enough resource or not. That is, the total packet arrival rate is smaller

[1] An alternative approach is not to admit the handoff call unconditionally. Rather, the base station will accept the handoff request based on the packet-level constraint. There is a tradeoff between handoff dropping probability and packet loss probability.

than the QoS guaranteed capacity $\tilde{C}$ or not ($\tilde{C}$ is obtained in Section 5.1). If true, the BS will accept the request; If not, the BS will proceed TCP call adjustment (cf. Algorithm 2). After TCP call adjustment, if the system has enough resource, the call request will be accepted, otherwise, the call will be rejected. This is the algorithm for new call admission displayed in Algorithm 1. We note that Algorithm 1 mainly deals with call-level decision, and the packet-level consideration is reflected in Algorithm 2: TCP call adjustment. The idea of Algorithm 2 is that the BS will enumerate all the ongoing TCP calls and reduce the packet-level transmission rate of TCP calls. Hence, the total transmission will be decreased until there is enough room to accommodate the new request. If there is still no enough capacity to accept this call after this enumeration, the call request will be rejected.

5 Performance Analysis

In this section, we analyze the system performance under the assumption that the service holding time is independent of the transmission rate as commonly assumed in most of the existing CAC literature [13, 12];

Before we analyze the call-level performance, we first present the effect of TCP flows on the packet level. Next, we describe a general form for the call blocking probability, and then we present the analysis for Cases I and II as a special case of the general form, respectively.

5.1 The Effect of TCP Flows on the Packet Level

Define $x = (x_1, ..., x_K)$, where x_k is the number of calls of class k. For convenience, we assume the 1^{st} class traffic is TCP call and the average throughput of TCP call $\mathbb{E}\{TCP\}$ is given by (1). Let $\mathbb{E}\{X_k\}$ be the average packet-level throughput of class k ($k = 2, ..., K$), which is determined from the predefined QoS profile. Let Ω be the set of all the admissible/possible states in the system and $\Omega_k \subset \Omega$, $(k = 1, ..., K)$ be the set of all the 'boundary states'. We formulate Ω and Ω_k as follows.

$$\Omega = \left\{ x \in \mathbb{Z}^{+K} \middle| x_1\mathbb{E}\{TCP\} + \sum_{k=2}^{K} x_k\mathbb{E}\{X_k\} \le \tilde{C} \right\},$$

and

$$\Omega_k = \left\{ x \in \Omega \middle| x_1\mathbb{E}\{TCP\} + \sum_{k=2}^{K} (x_k+1)\mathbb{E}\{X_k\} > \tilde{C} \right\},$$

for $k = 2, ..., K$, and

$$\Omega_1 = \left\{ x \in \Omega \middle| (x_1+1)\mathbb{E}\{TCP\} + \sum_{k=2}^{K} x_k\mathbb{E}\{X_k\} > \tilde{C} \right\}.$$

Here, $\tilde{C}$ (QoS guaranteed capacity in Table 1) is the maximum total packet arrival rate to the system packet-level with QoS guarantee. In other words, if the total packet arrival rate is greater than $\tilde{C}$, then the system cannot guarantee the predefined QoS requirement such as packet loss probability.

The packet loss probability and the induced $\tilde{C}$ can be obtained by many existing stochastic loss models such as: $M/M/1$ model [20]; effective bandwidth model [21]; Gaussian approximation model [22]. For example, an $M/M/1$ model assumes that the total arrival rate of all the classes follows Poisson process with mean λ and the service time of each packet is exponentially distributed with mean $1/\mu$. In this model, the service rates for different class are assumed to be the same and the total arrival rate λ is simply the summation of the arrival rate of each class. If we denote by $\rho = \frac{\lambda}{\mu}$ the traffic intensity, and the packet loss probability is given by $P\{loss\} = \frac{1-\rho}{1-\rho^{B+1}}\rho^B$, where B is the base station's buffer size in Table 1. For any given packet-levle QoS requirement $\overline{p}$, i.e., $P\{loss\} \le \overline{p}$, we set $\tilde{C}$ as the maximum packet arrival rate under this constraint, i.e., $\tilde{C} = \max\left\{\lambda \middle| \frac{1-\rho}{1-\rho^{B+1}}\rho^B \le \overline{p}\right\}$. Finally, we know that by changing the packet loss probability $p(t)$, we can control $E\{TCP\}$ according to (1) and thus also control the induced state space Ω and boundary states Ω_k.

5.2 Call Blocking Probability

Suppose in general that the call arrival process for class k is state-dependent Poisson process with mean $\Lambda_k(x)$ and the service holding time is exponential with mean $1/\Psi_k(x)$, which means both the arrival process and the service process are state dependent processes. We borrow some notations from [23] and define the $K-$dimensional base vector e_i, for $i = 1, ..., K$ as all the elements are zeros except for i^{th} element is 1, and the function $T_i^j(x)$ as $T_i^j(x) = x - e_i + e_j$, i.e., a transition where a single flow from i^{th} class departs and a single flow from j^{th} class arrives to the system. In particular, $T_0^k(x)$ ($T_k^0(x)$) denotes an arrival (a departure) of a flow from class k given that the current state is x.

Consider a given state $x \in \Omega$ with stationary probability $P(x)$. The transition rate out of the state x, denoted as $\mathcal{A}(x, \rightarrow)$, and the transition rate into the state x, denoted as $\mathcal{B}(\rightarrow, x)$, can be written as

$$\mathcal{A}(x, \rightarrow) = \begin{cases} \Lambda_k(x) & \text{if } T_0^k(x) \in \Omega \\ x_k \Psi_k(x) & \text{if } T_k^0(x) \in \Omega \end{cases}$$

$$\mathcal{B}(\rightarrow, x) = \begin{cases} \Lambda_k(T_k^0(x)) & \text{if } T_k^0(x) \in \Omega \\ (x_k+1)\Psi_k(T_0^k(x)) & \text{if } T_0^k(x) \in \Omega \end{cases}$$

where $k = 1, ..., K$. Therefore, by combining $\mathcal{A}(x, \rightarrow)$ and $\mathcal{B}(\rightarrow, x)$ with the stationary probability distribution $P(x)$, we can obtain the general form of steady-state balance equations for all x as follows:

$$\left[\sum_{k=1}^{K} \Lambda_k(\omega) 1_{\{T_0^k(\omega)\in\Omega\}} + \sum_{k=1}^{K} n_k \Psi_k(\omega) 1_{\{T_k^0(\omega)\in\Omega\}}\right] P(\omega)$$
$$= \sum_{k=1}^{K} \Lambda_k(T_k^0(\omega)) P(T_k^0(\omega)) 1_{\{T_k^0(\omega)\in\Omega\}}$$
$$+ \textstyle\sum_{k=1}^{K} (n_k+1)\Psi_k(T_0^k(\omega)) P(T_0^k(\omega)) 1_{\{T_0^k(\omega)\in\Omega\}},$$

where $\Lambda_k(x)$ and $\Psi_k(x)$ are state-dependent arrival and departure rate, respectively, and $\sum_{x\in\Omega} P(x) = 1$. Then, the blocking probability of class k, P_b^k, can be written as

$$P_b^k = \sum_{x\in\Omega_k} P(x), \tag{2}$$

Remark: From (2), we can clearly see the effect of packet-level dynamics on the call-level performance reflected by the definition of Ω and Ω_k. Different packet loss probabilities $p(t)$ result in different Ω and Ω_k, and hence the induced call-level performance will be different.

Throughout this paper we assume that the call arrival rate of class k is independent of the current system state, i.e., $\Lambda_k(x) = \Lambda_k$. In Case I, since we assume that the service holding time is independent of the transmission rate, it is independent of the system state, i.e., $\Psi_k(x) = \Psi_k$ for all x. So, by solving (2) with Λ_k and Ψ_k, the $P(x)$ will be of a product form as follows:

$$P(x) = \frac{\Phi(x)}{\Phi(\Omega)}, \quad x \in \Omega, \tag{3}$$

where $\Phi(x)$ is given by

$$\Phi(x) = \prod_{k=1}^{K} \frac{\Lambda_k^{x_k}}{\Psi_k^{x_k} x_k!}$$

and the normalizing factor $\Phi(\Omega)$ is $\Phi(\Omega) = \sum_{x\in\Omega}\Phi(x)$. Then, the blocking probability P_b^k for Case I can be calculated from (2).

Note that we analyze the performance under the another assumption as well. We consider the service holding time of a TCP call depends on its transmission rate (i.e., Ψ_1 is state dependent). Due to the space limit, we refer to our technical report for the detail [24].

6 Simulation Results

In this section, we present extensive simulation results of our TCP-aware CAC scheme and investigate its performance under the assumption described in Section 5. Due to the space limit, we refer to our technical report for the more simulation results [24].

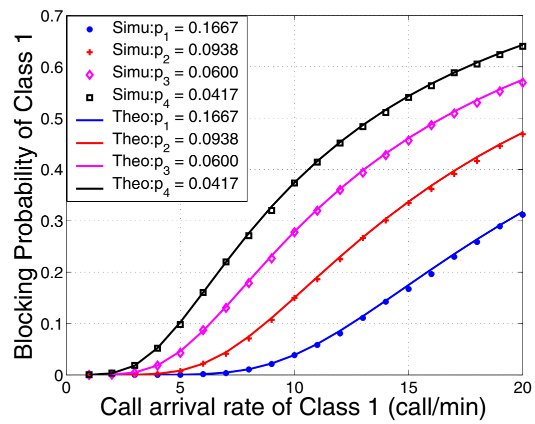

Figure 1: The blocking probability of Class 1 for different TCP AIMD control: $p_1 = 0.1667, p_2 = 0.0938, p_3 = 0.0600, p_4 = 0.0417$ corresponds to $E\{\text{TCP}\}_1 = 3, E\{\text{TCP}\}_2 = 4, E\{\text{TCP}\}_3 = 5, E\{\text{TCP}\}_4 = 6$ pkt/sec, respectively.

The system parameters used in the simulation are as follows. We consider two classes of calls and note that our scheme are quite general and could be effectively applied to the system with more classes. class 1 is TCP call with the transmission rate controlled by the packet loss probability $p(t)$ (the value of $p(t)$ is set according to different scenarios) and class 2 is the voice-type call with average transmission rate of $E\{\text{voice}\} = 2$ (pkt/sec), which could correspond to the voice transmission rate ranging from 2 kbps and 8 kbps by choosing different packet sizes [25]. The call arrival processes for both classes are characterized by Poisson process with mean rate λ_1 and λ_2, respectively, and the transmission time is exponentially distributed with mean $1/\mu_1$ and $1/\mu_2$, respectively. The performance metrics consist of call blocking probability [2] and throughput (call/min) for call level, and link

[2]In order not to distract from our focus, we do not consider

utilization for packet level.

We present simulation results when the service holding time is independent of transmission rate, and hence is irrelevant to packet loss probability $p(t)$. Then, we fix $p(t)$ during the simulation as p and investigate the system performance under different p denoted as p_i. We show that different packet loss probability results in different system performance and thus by controlling p_i of TCP flows, we can make the system yield the desirable performance.

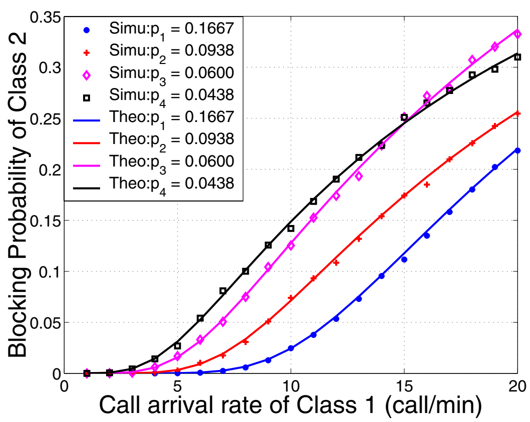

Figure 2: The blocking probability of Class 2 for different TCP AIMD control: $p_1 = 0.1667, p_2 = 0.0938, p_3 = 0.0600, p_4 = 0.0417$ corresponds to $E\{\text{TCP}\}_1 = 3, E\{\text{TCP}\}_2 = 4, E\{\text{TCP}\}_3 = 5, E\{\text{TCP}\}_4 = 6$ pkt/sec, respectively.

We know that different p_i corresponds to different average throughput of TCP calls denoted by $E\{\text{TCP}\}_i$. Figures 1 and 2 display the blocking probabilities with the increase of arrival rate of class 1, while keep the arrival rate of class 2 fixed i.e., $\lambda_2 = 2$. We plot both the simulation results (the legends with markers but no lines) and the theoretical results, i.e., P_b^k derived based on (3), (the legends with lines but no markers) for the purpose of comparison. We can see the simulation results are in good agreement with the theoretical results for both Figures.

Moreover, Figure 1 shows a clear trend that with the decrease of p (i.e., the increase of average TCP throughput), the blocking probability of class 1 increases. However, this trend is violated in Figure 2 when the arrival rate is larger than 15 (call/min) (cf. p_3 and p_4). There is a trade off between p_3 and p_4, i.e., the blocking probability of class 1 of p_3 is smaller than that of class 1 of p_4 in Figure 1, but the blocking probability of class 2 of p_3 is larger than that of class 2 of p_4 in Figure 2. Even though the underlining reason is not quite clear yet, the results indicate that there may exist tradeoff among different classes under some situations. However the overall trend is that the blocking probabilities increase with the decrease of p.

the handoff call specially.

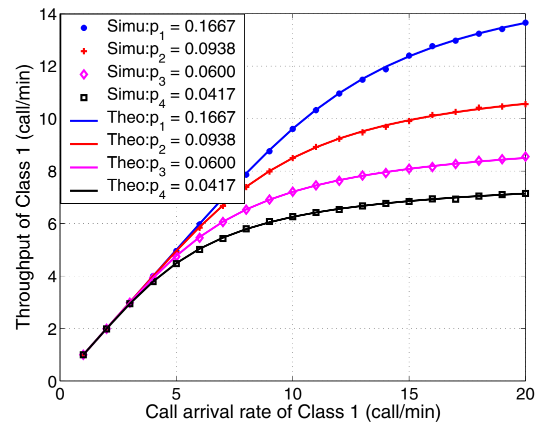

Figure 3: The throughput of Class 1 for different TCP AIMD control: $p_1 = 0.1667, p_2 = 0.0938, p_3 = 0.0600, p_4 = 0.0417$ corresponds to $E\{\text{TCP}\}_1 = 3, E\{\text{TCP}\}_2 = 4, E\{\text{TCP}\}_3 = 5, E\{\text{TCP}\}_4 = 6$ pkt/sec, respectively.

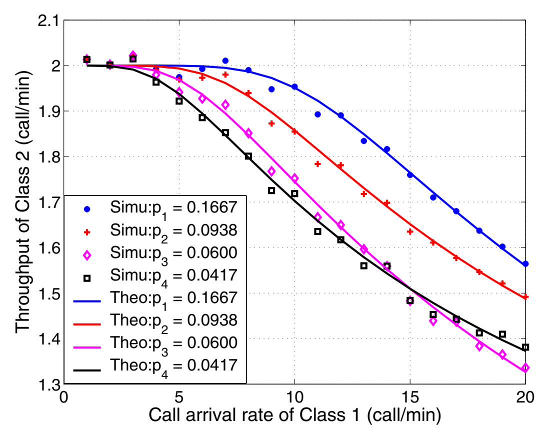

Figure 4: The throughput of Class 2 for different TCP AIMD control: $p_1 = 0.1667, p_2 = 0.0938, p_3 = 0.0600, p_4 = 0.0417$ corresponds to $E\{\text{TCP}\}_1 = 3, E\{\text{TCP}\}_2 = 4, E\{\text{TCP}\}_3 = 5, E\{\text{TCP}\}_4 = 6$ pkt/sec, respectively.

Figures 3 and 4 plot the call-level throughput (call/min) of class 1 and class 2 as the arrival rate of class 1 varies. Similarly to the blocking probability in the previous two figures, the simulation results and theoretical results are provided. Again, we can see the simulation results match the theoretical results. The reason is that since the call-level throughput is equal to the arrival rate times the non-blocking probability, i.e., $\lambda \times (1 - P_b^k)$, Figures 3 and 4 are the consequence of Figures 1 and 2, where the same reasoning is applied.

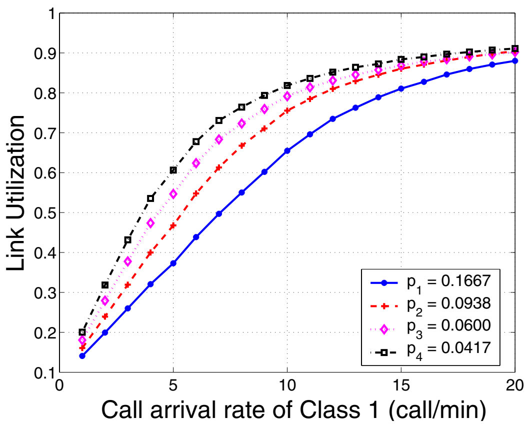

Figure 5: The link utilization for different TCP AIMD control: $p_1 = 0.1667, p_2 = 0.0938, p_3 = 0.0600, p_4 = 0.0417$ corresponds to $E\{\text{TCP}\}_1 = 3, E\{\text{TCP}\}_2 = 4, E\{\text{TCP}\}_3 = 5, E\{\text{TCP}\}_4 = 6$ pkt/sec, respectively.

The simulation results of link utilization under different p are plotted in Figure 5 indicating that the link utilization is increased as the decrease of p. The reason is that because the channel holding time is not related to the transmission rate, the decrease of p implies higher transmission rate and thus contributes to higher link utilization.

Overall, Figures 1 to 5 show that by increasing the average packet loss probability of TCP call, p, the system will have reduced call blocking probability, increased throughput at the slight expense of link utilization.

In addition, we simulate the service holding time of a TCP call depends on its transmission rate (i.e., Ψ_1 is state dependent), that shows that our TCP-aware CAC improves system performance at packet level and call level [24].

7 Conclusion

Traditional Call Admission Control (CAC) schemes only consider call-level performance and are believed to be sufficient for the circuit-switched wireless network. Since the future wireless network will become packet-switched, the packet-level performance cannot be ignored. This is especially true when the TCP-type of applications are running over such packet-switched wireless networks, because TCP congestion control algorithm will exhaust all the available resource until packet loss occurs. In order to regulate the TCP applications to friendly coexist with other types of services, we have proposed a TCP-AIMD aware CAC scheme based on a two-level framework that takes both the call-level and the packet-level performance into account. The performance of our scheme is analyzed at two levels, respectively, and the interaction between two levels is discussed. Our simulation results show that the proposed scheme can effectively improve the system performance such as call blocking probability (call-level metric), throughput (calls/min, call-level metric), and link utilization (packet-level metric), which are in good agreement with our analysis.

References

[1] V. Jacobson, "Congestion Avoidance and Control," *ACM Computer Communications Review*, vol. 8, pp. 314–329, Aug 1988.

[2] S. Ravot, "TCP Transfers Over High Latency Bandwidth Networks Grid DT," in *The First International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2003)*, Feb 2003.

[3] M. Jain, R. Prasad, and C. Dovrolis, "Socket Buffer Auto-Sizing for Maximum TCP Throughput," in *Journal of Grid Computing*, vol. 1, pp. 361–376, 2004.

[4] S. Floyd, "HighSpeed TCP for Large Congestion Windows," in *IETF, Draft, draft-floyd-tcp-highspeed-01.txt*, 2003.

[5] J. Liu and S. Singh, "ATCP: TCP for Mobile Ad-Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 1300–1315, July 2001.

[6] K. Brown and S. Singh, "M-TCP: TCP for Mobile Celluar Networks," *ACM Computer Communications Review*, vol. 27, no. 5, pp. 19–43, 1997.

[7] A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," in *15th International Conference on Distributed Computing Systems*, 1995.

[8] S. Y. Hui and K. Yeung, "Challenges in the Migration to 4G Mobile Systems," *IEEE Communications Magazine*, pp. 54–59, Dec 2003.

[9] H. Honkasalo, K. Pehkonen, M. T. Niemi, and A. T. Leino, "Analysis and Design of Controllers for AQM Routers Supporting TCP Flows," *IEEE Wireless Communications Magazine*, pp. 14–18, April 2002.

[10] W. Wang, X. Wang, and A. Nilsson, "A New Admission Control Scheme under Energy and QoS Constraints for Wireless Networks," in *Proceedings of IEEE Infocom'05, Miami, FL*, March 2005.

[11] R. G. Akl, M. V. Hegde, and M. Naghi-Pour, "Mobility-Based CAC Algorithm for Arbitrary Call-Arrival Rates in CDMA Celluar Systems," *IEEE Transactions on Vehicular Technology*, vol. 54, pp. 639–651, March 2005.

[12] C. W. Leong, W. Zhuang, Y. Cheng, and L. Wang, "Call Admission Control for Integrated On/Off Voice and Best-Effort Data Services in Mobile Celluar Communications," *IEEE Transactions on Communications*, vol. 52, pp. 778–790, May 2004.

[13] Y. Fang and Y. Zhang, "Call Admission Control Scheme and Performance Analysis in Wireless Mobile Networks," *IEEE Transaction on Vehic. Technology*, vol. 51, pp. 371–382, March 2002.

[14] M. Ghaderi, R. Boutaba, and G. W. Kenward., "Stochastic Admission Control for Quality of Service in Wireless Packet Networks," in *Proceedings of the 4th IFIP Networking Conference (Networking'2005)*, May 2005.

[15] D. Zhang, X. Shen, and J. W. Mark, "Radio Resource Management for Cellular CDMA Systems Supporting Heterogeneous Services," *IEEE Transactions on Mobile Computing*, vol. 11, pp. 32–39, April-June 2003.

[16] D. Zhang, X. Shen, and J. W. Mark, "Efficient Call Admission Control for Heterogeneous Services in Wireless Mobile ATM Networks," *IEEE Communications Magazine*, vol. 38, pp. 72–78, Oct 2000.

[17] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose, "Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation," *IEEE Transactions on Networking*, vol. 8, pp. 133–145, April 2000.

[18] M. H. Ahmed, "Call Admission Control in Wireless Networks: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 1, pp. 49–68, 2005.

[19] B. Li, L. Li, B. Li, K. M. Sivalingam, and X. Cao, "Call Admission Control for Voice/Data Integrated Cellular Networks: Performance Analysis and Comparative Study," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 706–718, May 2004.

[20] L. Kleinrock, "Queueing Systems, Volume I: Theory," *Wiley Interscience*, 1975.

[21] E. W. Knightly, "H-BIND: A New Approach to Providing Statitical Performance Guarantees to VBR Traffic," in *Proceedings of IEEE INFOCOM'96*, March 1996.

[22] S. M. Ross, "Introduction to Probability Models, Eighth Edition," *Academic Press, London*, 2003.

[23] R. F. Serfozo, "Introduction to Stochastic Networks," *Springer*, 1999.

[24] X. Wang, D. Y. Eun, and W. Wang, "A TCP-AIMD Aware Call Admission Control Scheme for Packet-Switched Wireless Networks: A Two-Level Approach," in *Technical Report, TR-2005-02, availabe at Http://www4.ncsu.edu/~xwang8/cac_tcp.pdf*, July 2005.

[25] "UMTS," in *Http://www.3GPP.org.*