

Channel Capacity Fair Queueing in Wireless Networks: Issues and A New Algorithm

Li Wang, Yu-Kwong Kwok, Wing-Cheong Lau, and Vincent K. N. Lau

Department of Electrical and Electronic Engineering, The University of Hong Kong

Abstract -- Wireless fair queueing algorithms have been extensively studied recently. However, a major drawback in existing approaches is that the channel model is overly simplified—a two states (good or bad) channel is assumed. While it is relatively easy to analyze the system using such a simple model, the algorithms so designed are of a limited applicability in a practical environment, in which the level of burst errors are time-varying and can be exploited by using channel adaptive coding and modulation techniques. In this paper, we first argue that the existing algorithms cannot cater for a more realistic channel model and the traditional notion of fairness is not suitable. We then propose a new notion of fairness, which bounds the actual throughput normalized by channel capacity of any two sessions. Using the new fairness definition, we propose a new fair queueing algorithm called CAFQ (Channel Adaptive Fair Queueing), which, as indicated in our numerical studies, outperforms other algorithms in terms of overall system throughput and fairness among error prone sessions.

Keywords: wireless networks, fair queueing, fairness, performance guarantees, quality of service.

I. INTRODUCTION

Fair queueing algorithms have been extensively studied in wireline networks for providing QoS (Quality of Service) guarantees to connections among end hosts. In recent years, as wireless networks proliferate, researchers have also put much efforts in extending the fair queueing techniques for applications in a wireless environment [1], [8], [14]. However, a major drawback in these wireless fair scheduling techniques is that the channel model is rather unrealistic: the channel is either in a “good” state (or perfect state) in which a session can transmit using full bandwidth, or in a “bad” state in which a session cannot transmit any data. In reality, using state-of-the-art channel adaptive techniques [7], the transmitter/receiver in a wireless network can exploit the time-varying nature of the channel and accordingly adjust the effective throughput by choosing an appropriate level of FEC (forward error correction). Simply put, in techniques such as ABICM (Adaptive Bit-by-Bit Interleaved Channel Modulation) proposed in [7], when the channel condition is not good (by checking the pilot symbols in a feedback channel or reverse link), the amount of protection can be re-adjusted by choosing a different channel coding and modulation mode [7]. Thus, even in a so-called “bad” channel state, a mobile terminal can in fact transmit data and realize a possibly lower effective throughput, instead of totally unable to transmit.

With such a realistic channel adaptive transmission method, intuitively the overall system throughput will be enhanced. However, a question remains is that what a scheduler should do in order to maintain the fairness among the sessions in the system, which, more often than not, are in a “not so good” channel states. In previous algorithms, the answer is simple because

those algorithms simply regard a session as “dormant” (cannot transmit) if it is in a “not so good” channel state—only a session with a “perfect” (the best) channel state can transmit.

In view of the fact that existing algorithms cannot cater for the situation where multiple channel quality levels exist, in this paper we propose a new notion of fairness, which then induces our proposed algorithm called CAFQ (Channel Adaptive Fair Queueing). As indicated by our numerical studies, the CAFQ algorithm outperforms other existing state-of-the-art algorithms in that CAFQ produces a higher overall system throughput and maintains fairness even among the sessions without perfect channel conditions.

The balance of the paper is as follows. In the next section, we first provide a discussion on the different fairness notions in wireline and wireless networks, and then demonstrate that a new fairness notion is needed in order to cater for the multi-level channel qualities. We then describe our new notion of fairness and the CAFQ algorithm in detail in Section III. Due to space limitations, we only present one numerical example in Section IV. Finally, we provide some concluding remarks in Section V.

II. FAIRNESS NOTIONS

A. Effort Fair and Outcome Fair

In a broad sense, fairness can be defined with respect to two aspects: effort and outcome [2]. Intuitively, a policy is called *effort fair* if the allocation of services to different sessions is fair, without regard to the actual amount of data successfully delivered by the sessions using the allocated services. Informally, “fair” means a session gets the service amount that it deserves to get. On the other hand, a policy is called *outcome fair* if the actual realized data throughput among the sessions is fair.

Effort Fair: A scheduler is fair if the bandwidth (e.g., the amount of time slots) the system allocates to different sessions is proportional to the different service shares. Mathematically, that means the difference between the normalized services the system allocates to any two sessions i and j is bounded as follows:

$$\left| \frac{S_i(t_1, t_2)}{r_i} - \frac{S_j(t_1, t_2)}{r_j} \right| < \epsilon \quad (1)$$

where $S_i(t_1, t_2)$ denotes the allocated service of a certain session i during time interval $[t_1, t_2]$, r_i is the requested service share, and ϵ is a finite constant. Such a fair scheduler can be considered as effort fair [3] in that the scheduler only guarantees the effort expended on the sessions is fair, without regard to the actual throughput achieved by the different sessions.

Outcome Fair: A scheduler is fair if the difference between

the normalized amount of realized throughput of any two sessions i and j is bounded as follows:

$$\left| \frac{T_i(t_1, t_2)}{r_i} - \frac{T_j(t_1, t_2)}{r_j} \right| < \varepsilon \quad (2)$$

where $T_i(t_1, t_2)$ denotes the actual throughput a session i achieves during the time interval $[t_1, t_2]$. Such a fair scheduler can be considered as outcome fair [9] in that the scheduler tries to provide a fair actual performance achieved by the sessions (rather than the “nominal” performance as in the effort fair definition discussed above).

In a TDMA system, “effort” is the number of time-slots allocated, while “outcome” is the actual data throughput using the allocated time-slots. Note that a “variable actual throughput” is manifested by the fact that some data may be lost due to poor channel conditions and thus, inducing retransmissions; or, in adaptive FEC schemes such as ABICM, the amount of data protection varies according to the channel conditions.

B. Fairness Notions for Wireless Networks

Recently, much research has been done on devising new algorithms for fair queueing in wireless networks. A myriad of algorithms have been proposed [1], [3], [5], [6], [8], [9], [10], [11], [12], [13], [14]. The general idea of wireless scheduling algorithms is as follow: the scheduler simulates an error-free system running a wireline packet scheduling algorithm when the sessions have good (or perfect) channel states (effective throughput is maximum). When the session that is scheduled to transmit data encounters a bad channel state, it will give up the transmit opportunity to other error-free sessions (i.e., with a good channel state), then these error-free sessions will give their transmit rights back to the error session to compensate when it escapes from a bad channel state. Thus, in fact, the scheduler tries to swap the allocated time slots between error-free sessions and error prone sessions when sessions encountering error. The goal is to hide the short term channel error burst from the end users. The system maintains long term fairness at the expense of instantaneous fairness between sessions.

In our study, we have considered the following existing scheduling algorithms for wireless networks: WPS (Wireless Packet Scheduling) [9], IWFQ (Idealized Wireless Fair Queueing Algorithm) [9], CIF-Q (Channel-Condition Independent Fair Queueing) [12], SBFA (Server Based Fairness Algorithm) [13], CS-WFQ (Channel State Independent Wireless Fair Queueing) [8], ELF (Effort Limited Fairness) [3], Proportional Fairness (PF) [4], and WFS (Wireless Fair Service) [11]. A scrutiny of these current scheduling algorithms for wireless networks reveals that in most of these algorithms, there are two common major deficiencies:

1. The channel model is too simple and not realistic. Only a two state (good or bad) model is used.
2. There is few analysis for sessions which have bad channel states.

On the surface, these previous algorithms work well in that they schedule the error-free sessions to transmit data while leaving the error sessions (in a bad channel state) waiting until

they recover. Thus, to guarantee fairness, it suffices to guarantee that the error sessions can catch up (i.e., get back the missing service) within a bounded period of time. However, nothing is said about the behavior and the time bound of the error period. Furthermore, the key assumption, which, we believe, is the major drawback, is that a session in bad channel state can transmit nothing. This is undeniably an over-simplification in view of the fact that channel adaptive and variable rate MAC protocols are commonly sought to combat the time-varying nature of wireless channels. Algorithms that use such a simplified assumption include: CIF-Q, IWFQ, SBFA, and WFQ.

On the other hand, more practical algorithms, such as the ELF, CS-WFQ (uses a similar principle as in ELF), proportional fair, and our proposed algorithm, allow sessions to transmit packets even though the sessions are in a non-perfect channel state (hence, “effort” is very likely not equal to “outcome”).

III. CHANNEL ADAPTIVE FAIR QUEUEING

A. Overview

Our proposed algorithm is called Channel Adaptive Fair Queueing (CAFQ) which has the following distinctive features:

- a new notion of fairness is employed;
- contrary to CIF-Q, graceful degradation is not ensured to help the lagging session more efficiently;
- a punish factor is used to decide how seriously the scheduler punishes a non-perfect channel state session that transmit packets; and
- virtual compensation session is incorporated to help the lagging sessions to catch up.

From the user’s viewpoint, fairness should be maintained in that so long as a session can transmit something, it should be provided some chance to transmit. At the same time, QoS should also be met. However, from the system manager’s viewpoint, its hard to meet these two sometimes conflicting goals with a limited bandwidth and channels that have time-varying quality. Because whenever a session without a perfect channel state is allowed to transmit, there will be part of the bandwidth wasted, and the wasted bandwidth can never be replenished. It should be noted that this is very different from the idea of swapping sessions that are error-free and error prone, as in existing scheduling algorithms such as CIF-Q. When a error-free session takes the opportunity of a error prone sessions, it will relinquish the service when the error prone one recovers.

Of course, if the maximum available bandwidth is very large or the channel state is most likely to be perfect, we should maintain the graceful degradation, and prevent the leading ones from starving. But in a realistic system in which the channel is not so good, we cannot expect to achieve perfect allocations, but rather we should meet the sessions QoS first. Thus, in our proposed CAFQ algorithm, graceful degradation is not implemented and the rationale is to compensate the lagging sessions as soon as possible.

B. Channel Adaptive Fairness

We propose a new notion of fairness to be maintained in the short term, called *channel adaptive fairness* (CAF). To achieve CAF, a scheduler is fair if in the short term the difference between the normalized throughput (normalized with respect to the channel capacity) of any two sessions i and j is bounded as follows:

$$\left| \frac{T_i(t_1, t_2)}{r_i f(\Phi_i)} - \frac{T_j(t_1, t_2)}{r_j f(\Phi_j)} \right| < \varepsilon \quad (3)$$

where Φ_i denotes the channel state (e.g., one of the five classes A, B, C, D, and E), and $f(\Phi_i) = M(\Phi_i)^a$ in which $M(\Phi_i)$ is the *effective throughput factor* ($0 \leq M(\Phi_i) \leq 1$). The effective throughput factor is channel state dependent: $M(\Phi_i) = 0.75$ if Φ_i is channel state B, and so on. Here, a is a *punish factor* which is a positive number. Thus, in our definition of fairness, the throughput a session receives will be proportional to its channel quality. And, in the long term, outcome fair is maintained among all sessions

Our proposed fairness is more reasonable in the wireless environment because it considers explicitly the different channel states. Unlike the CIF that prevents the sessions without perfect channel state from transmitting and unlike the ELF that distributes the normalized amount of service inversely proportional to their channel states, it allows transmission to all sessions that do not suffer from the worst channel state in the short term, and at the same time, it punishes the sessions without good channel states to different extent. Furthermore, unlike the proportional fair scheduler, using CAF the scheduler does not necessarily schedule the session with the best channel condition to transmit. With the channel adaptive fairness, we devise a new fair queueing algorithm, which is explicated in detailed in the following section.

The punish factor a can help to decide between to make use of the bandwidth more efficiently and to treat every session more fairly. When a larger value of punish factor is used, we punish the non-perfect channel state session that transmit packets more seriously, and prevent them from wasting too much bandwidth. In effect, the bandwidth is used more efficiently, and the average delay of the total system is decreased and the throughput is increased. But if there is a session that is more unlucky than the others and have a higher probability of having a bad channel state, its average delay and throughput may be very bad, because it is punished seriously and prevented from occupying the bandwidth. When a smaller punish factor is used, this kind of unlucky sessions will be punished only moderately, so the average delay and throughput of these sessions are reduced. But as they have more chance to access the bandwidth and hence incur a larger wastage of bandwidth, the total throughput and average delay of the system will be adversely affected. Thus, the punish factor can be used to tune the utilization of system resources.

C. Comparison with other Fairness Notions

Having defined our proposed fairness notion, it is useful to compare it with other existing fairness notions, as shown in Table 1.

TABLE 1: Qualitative comparison of fairness notions.

Fairness	Short-term	Long-term
CIF	short term fairness is maintained only among sessions with perfect channel states; neither outcome fair nor effort fair is considered for sessions with "not so good" channel states	outcome fair
ELF	outcome fair is maintained among sessions with channel states better than a predefined threshold	not precisely defined
PF	short term fairness is not precisely maintained	outcome fair
CAF	short term fairness (normalized by channel states) is maintained among all the sessions unless the session has the worst channel state; a compromise is achieved in attaining outcome fair and in attaining efficient bandwidth usage	outcome fair

D. Detailed Description of CAFQ

As in existing algorithms, we associate the scheduling system with an error-free system to account for the service lost or gained by a session due to errors. A session is classified as leading or non-leading depending on the difference of the service it received between the error-free system and the real one. A session is leading if it has received more service in the real system than in the error-free one, while it is non-leading if it has received less or the same amount.

We simulate SFQ (Start-Time Fair Queueing) [15] in the error-free system in our study for the reason of simplicity because it is hard to schedule according to the finish times of the packet in the wireless environment. In the SFQ, when packet k of session i arrives, it is stamped with a virtual start time $S(P_{i_k})$, computed as:

$$S(P_{i_k}) \leftarrow \max\{V((A(P_{i_k})), F(P_{i_{k-1}}))\} \quad (4)$$

$$F(P_{i_k}) \leftarrow \left(S(P_{i_k}) + \frac{l_{i_k}}{r_i} \right) \quad (5)$$

where P_{i_k} is the k th packet of session i , $F(P_{i_k})$ is the virtual finish time of packet P_{i_k} , $V(A(P_{i_k}))$ is the virtual clock of the system at the arrival time $A(P_{i_k})$ of the packet, r_i is the pre-allocated service share of session i , and l_{i_k} is the length of the packet. The virtual time of the packets are initialized to zero. In

the error-free system, a session i is selected in the increasing order of the sessions virtual starting times among sessions that are backlogged. Since it is possible that the packet of another session instead of session i will be transmitted in the real system, a session's virtual time only keep track of the normalized service received by the session in the error-free system.

Another parameter, Δ , is used to keep track of the difference of the service a session received in the real system and in the error-free one. The Δ of a session is initialized to 0 when it becomes backlogged. A session is non-leading if Δ is greater than or equal to 0, while it is leading if Δ is less than 0.

In CAFQ, fairness is maintained in two aspects: in the short term, CAF is maintained among the leading sessions and non-leading sessions separately unless the sessions have the worst channel state (cannot transmit). In the long term, outcome fair is ensured with the help of virtual compensation session.

1) Short Term Fairness

We introduce two parameters N and L to implement the channel adaptive fairness in the short term. N_i keeps track of the normalized amount of services received by session i which is proportional to its channel state function when it is non-leading. When a session i becomes both non-leading and not suffering from the worst channel state, N_i will get initialized as follows:

$$N_i \leftarrow \max\{N_i, \min_{k \in \Psi} \{N_k \mid \text{lag}_k \geq 0\}\} \quad (6)$$

where Ψ denotes the set of sessions that are backlogged and for a non-leading session chosen to transmit packets in the real system, the N_i is updated as follows:

$$N_i \leftarrow \left(N_i + \frac{l_i}{r_i f(\Phi_i)} \right) \quad (7)$$

and L_i is defined similarly. Here, L_i keeps track of the normalized amount of services received by session i which is proportional to its channel state function when it is leading. When a session i becomes both leading and not suffering from the worst channel state, L_i will get initialized in a way analogous to (6).

In the real system, selection is made among the non-leading ones first. The session with the minimum N_i will be selected, and the packet at the head of the waiting queue of this session will be transmitted and N_i will be updated accordingly. If there is no such kind of session which is non-leading and backlogged, the system will select from the leading ones in the increasing order of the sessions' L_i and L_i will be updated. If all sessions are not backlogged, dummy packets will be sent. If the session j selected in the real system is not the one chosen in the error-free one and it is i that is selected in the error-free system, the Δ of i and j will both be updated: $\Delta_i \leftarrow (\Delta_i + l_i)$, $\Delta_j \leftarrow (\Delta_j - l_j)$; otherwise, the Δ will not be changed.

When a session with a comparatively bad channel state transmits packet, the N_i or L_i will increase more rapidly than a session with a better channel state. As the punish factor changes, we can decide how serious we should punish a session which does not have a perfect channel and transmits packets. The larger the punish factor is, the more serious we punish the unlucky sessions.

2) Long Term Fairness

Nonetheless, there is still one problem: although the sessions, which do not have perfect channel states but get packets transmitted, are punished, they are given some chance to transmit, and part of the bandwidth of the system is wasted and can never get compensated. Because the scheduler will not schedule a leading session to transmit if there is a lagging one which is backlogged and is not in the worst channel state (i.e., state E), the scheduler will not save the effort of the system as most of the other scheduling algorithms do. So, we assign a service share to a virtual compensation session to help in the long term. This pre-allocated service share is used to help the lagging ones with perfect channel state, because only when a session has a perfect channel state, it can get compensation most efficiently. When a lagging session exits from non-perfect channel states, its session ID will be queued in the virtual compensation session. Sessions that are queued in the virtual compensation session are in the decreasing order of their Δ . So we give bonus service to the lagging sessions if it has perfect channel state, and the session which lags most will get it first so that it can be helped to catch up, and thus, long term outcome fair can be maintained.

In the error-free system, we select a session i among all the backlogged sessions and the virtual compensation session in the increasing order of the virtual time. If it is the virtual compensation session that is selected and there is session ID waiting in the queue, the session with the ID at the head of the virtual compensation queue will be scheduled to transmit in the real system. The Δ of this session will be decreased as $\Delta_i \leftarrow (\Delta_i - l_i)$. If it is not the virtual compensation session that is selected or there is no session ID waiting in the queue, the system will select a session to transmit in the real system from the non-leading ones according to N_i , then from the leading ones according to L_i if there is no non-leading one to take the service as we have mentioned above.

IV. A NUMERICAL EXAMPLE

In this example, the simulation time is 500 seconds and we computed the average result over 10 simulation runs. We simulate CS-WFQ and our CAFQ algorithm under 5 kinds of error modes [7]. There are 3 sessions in the system. The pre-allocated service rates of them are: 0.25, 0.25, 0.5 in CS-WFQ. The virtual compensation session has the rate of 0.2, and the other sessions rates are: 0.2, 0.2, 0.4 in the CAFQ algorithm. The data source of the sessions are Poisson sources with the arrival rates as: 0.8Mbps, 0.8Mbps, and 1.6Mbps. All the ses-

sions in the system have the same kind of error mode in each simulation run. The punish factor in CAFQ is set to be 1. We calculate the average delay, max delay of all the session and the system throughput both in the CAFQ algorithm and in CS-WFQ when the error mode changes. The results are shown in Figure 1 and Figure 2.

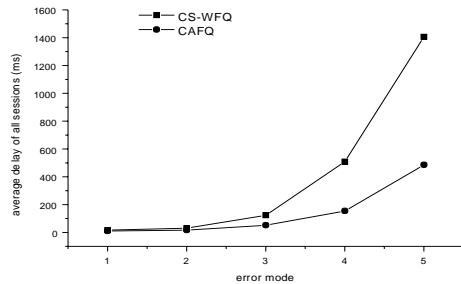


Figure 1: The average delays in CS-WFQ and CAFQ increase as overall channel state becomes worse

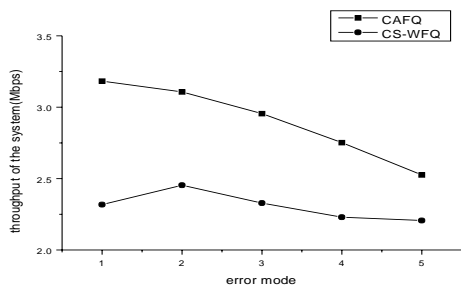


Figure 2: The throughput in both systems decrease as the overall channel state becomes worse.

As can be seen, the average delays increase and the throughputs decrease as the overall channel state becomes worse both in CS-WFQ and in the new CAFQ algorithm. But the rate of increase (decrease) is slower in the CAFQ algorithm, and the average delays are always smaller in the system with the CAFQ algorithm than in the system with CS-WFQ, while the throughputs in the CAFQ algorithm are always higher than in CS-WFQ. It is because CS-WFQ wastes the bandwidth seriously by maintaining outcome fair within the effort limit. Thus, fewer packets can be transmitted in a given time period, and packets have to wait for a longer time before they get transmitted so that the average delays grow. On the contrary, the CAFQ algorithm grants the session without perfect channel state to transmit at the same time of punishing them, so they have chance to transmit, but the chance is less if it has worse channel state.

In the short term, CAF is maintained and thus, the sessions' need for maintaining outcome fairness is handled. At the same time, efficient utilization of bandwidth is also achieved such that the precious bandwidth is not wasted to desparately maintain outcome fair. In the long run, virtual compensation session helps the session which lags most seriously and has perfect channel state. This helps to reduce the avearge delays.

V. CONCLUDING REMARKS

In this paper, we study qualitatively and quantitatively different fair queueing scheduling algorithms in wireless networks. Because of the time-varying nature of the wireless channel in a practical situation, burst errors are the norm rather than an exception and, thus, we believe that a good scheduling algorithm should take into consideration, or even exploit, the variations of channel conditions among the mobile terminals. We propose a new notion of fairness in which a scheduler is fair with respect to the throughput normalized by the channel capacity. Using this new fairness definition, we propose a new scheduling algorithm called Channel Adaptive Fair Queueing.

ACKNOWLEDGMENTS

This research is supported by HKU URC seed grants under contract numbers 10203010 and 10203413, and by a RGC research grant under contract number HKU7024/00E.

REFERENCES

- [1] V. Bharghavan, S. Lu, and T. Nandagopal, "Fair Queueing in Wireless Networks: Issues and Approaches," *IEEE Personal Communications*, Feb. 1999, pp. 44–53.
- [2] Y. Cao and V. O. K. Li, "Scheduling Algorithms in Broadband Wireless Networks," *Proceedings of the IEEE*, vol. 89, no. 1, Jan. 2001, pp. 76–87.
- [3] D. A. Eckhardt and P. Steenkiste, "Effort-Limited Fair (ELF) Scheduling for Wireless Networks," *Proc. INFOCOM'2000*, pp. 1097–1106, 2000.
- [4] A. Jalali, R. Padovani, and R. Pankaj, "Data Throughput of CDMA-HDR: A High Efficiency High Data Rate Personal Communication Wireless System," *Proc. VTC'2000*.
- [5] M. R. Jeong, H. Morikawa, and T. Aoyama, "Fair Scheduling Algorithm for Wireless Packet Networks," *Proc. ICPP'99*, pp. 280–285, 1999.
- [6] M. Kang and S. Wilbur, "A Fair Guaranteed Down-Link Sharing Scheme for Cellular Packet Switched Networks," *Proc. GLOBECOM'97*, pp. 1006–1010, 1997.
- [7] V. K. N. Lau, "Performance of Variable Rate Bit-Interleaved Coding for High Bandwidth Efficiency," *Proc. of VTC'2000*, vol. 3, pp. 2054–2058, May 2000.
- [8] P. Lin, B. Bensaou, Q. L. Ding, and K. C. Chua, "A Wireless Fair Scheduling Algorithm for Error-Prone Wireless Channels," *Proc. WoWMoM'2000*, pp. 11–20, 2000.
- [9] S. Lu, V. Bharghavan, and R. Srikant, "Fair Scheduling in Wireless Packet Networks," *IEEE/ACM Trans. Networking*, vol. 7, no. 4, pp. 473–489, Aug. 1999.
- [10] J. R. Moorman and J. W. Lockwood, "Multiclass Priority Fair Queueing for Hybrid Wired/Wireless Quality of Service Support," *Proc. WoWMoM'99*, pp. 43–50, 1999.
- [11] T. Nandagopal, S. Lu, and V. Bharghavan, "A Unified Architecture for the Design and Evaluation of Wireless Fair Queueing Algorithms," *Proc. MOBICOM'99*, pp. 132–142, 1999.
- [12] T. S. E. Ng, I. Stoica, and H. Zhang, "Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors," *Proc. INFOCOM'98*, pp. 1103–1111, 1998.
- [13] P. Ramanathan and P. Agrawal, "Adapting Packet Fair Queueing Algorithms to Wireless Networks," *Proc. MOBICOM'98*, pp. 1–9, 1998.
- [14] A. Stamoulis and G. B. Giannakis, "Packet Fair Queueing Scheduling Based on Multirate Multipath-Transparent CDMA for Wireless Networks," *Proc. INFOCOM'2000*, pp. 1067–1076, 2000.
- [15] H. Zhang, "Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks," *Proceedings of the IEEE*, vol. 83, no. 10, Oct. 1995, pp. 1374–1396.