On the Throughput and Fairness Performance of TCP over Ethernet Passive Optical Networks

Kai-Chien Chang, *Member, IEEE,* and Wanjiun Liao, *Senior Member, IEEE*

***Abstract*— The Ethernet Passive Optical Network (EPON) is a promising solution for metropolitan optical networks. In EPON, the downstream channel is a point-to-multipoint broadcast network and the upstream channel is a multipoint-to-point network based on a polling mechanism. In this paper, we model the performance of TCP over EPON with respect to the aggregate throughput and fairness. We calculate the maximum aggregate throughput of TCP flows on each of the upstream and downstream channels, and identify the fairness problem with existing polling schemes for EPON. We then derive two tight bounds of the fairness index for downloading users, and propose two polling mechanisms to improve unfair resource sharing among multiple TCP flows. The performance of the proposed mechanisms is evaluated via ns-2 simulations. The results show that our mechanisms can improve fairness among downloading TCP flows without degrading the aggregate throughput and fairness of uploading TCP flows.**

***Index Terms*— EPON, TCP, polling, fairness.**

I. Introduction

ETHERNET Passive Optical Networks (EPONs) [1-2] have been a popular choice for broadband access to the Internet. In EPON, the topology used is tree-and-branch. An EPON consists of an optical line terminal (OLT), a splitter, and some optical network units (ONU). The OLT is often installed in the central office, and the ONUs can be located at the curb or in the home, depending on where the optical line ends. An OLT is required to support at least 16 ONUs. As a result, the splitter should be able to split a signal into 16 signals.

In EPON, the downstream channel (i.e., from the OLT to ONUs) is a point-to-multipoint network, in which the only transmitter is the OLT and all ONUs are receivers; the upstream channel (i.e., from ONUs to the OLT) is a multipoint-to-point network, in which all ONUs may be transmitters and the OLT is the only receiver. The downstream channel is a broadcast channel, and thus all ONUs can hear transmissions from the OLT. The upstream channel is a multiple access channel shared by all ONUs, and thus a media access control mechanism is needed to arbitrate the access to the upstream channel.

The multiple access technique adopted by the 802.3ah group for EPON is a polling-based mechanism. The OLT polls each ONU in turn, and each ONU sends a REPORT message in reply and attaches its current queue size to the REPORT message. From these REPORT messages, the OLT obtains the transmission requests of active ONUs at the moment, and schedules the requests it has received based on a scheduling algorithm. The OLT informs each ONU of the start time and the end time it can use to transmit data by a GATE message. Each ONU, upon receiving a GATE message, knows when to turn on its laser and the period of time the laser has to remain on.

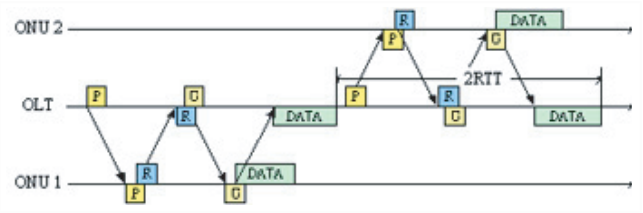


Fig. 1. Simple polling.

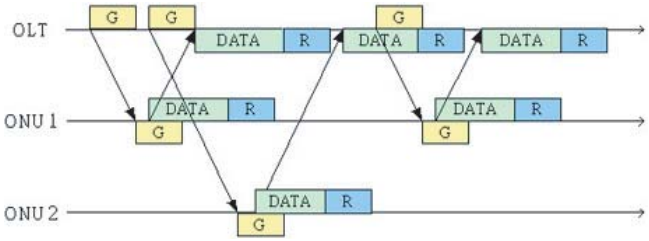


Fig. 2. IPACT polling.

This simple polling mechanism, however, suffers from long polling latency. As shown in Fig. 1, the upstream channel is idle for at least twice the round trip time (RTT) before a data transmission starts, where P, R, and G represent polling, REPORT, and GATE, respectively. An RTT is defined as two times of the minimum propagation delay from the OLT to ONUs. To fully utilize the bandwidth, the Interleaved Polling with Adaptive Cycle Time (IPACT) protocol [3] was proposed. With IPACT, the OLT keeps a table to record all ONUs and their requests, and sends a GATE to each ONU in the table in a cyclic order so as to inform each of them about the starting time and transmission period for sending data. As show in Fig. 2, the GATE must be sent out one RTT before the respective data arrives, regardless of whether the data of the ONUs which have been polled before them have arrived at the OLT. For example, the GATE message for ONU_2 is sent before the data of ONU_1 arrives at the OLT in Fig. 2. Each ONU, upon receiving a GATE message, starts sending its data packets at the time specified by the GATE. At the end of the granted

Manuscript received June 15, 2003; revised December 20, 2004. This work was supported in part by National Science Council (NSC), Taiwan, under a Center Excellence Grant NSC95-2752-E-002-006-PAE, and in part by NSC under Grant Number NSC95-2221-E-002-066.

K.-C. Chang and W. Liao are with the Department of Electrical Engineering and the Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan (e-mail: wjliao@ntu.edu.tw).

Digital Object Identifier 10.1109/JSAC-OCN.2006.014904.

period, the ONU attaches a REPORT message to tell the OLT its current queue size, i.e., the size it requests for the next cycle. Upon receiving a REPORT, the OLT checks whether the requested size exceeds the maximum size each ONU can use in a cycle, denoted by W_{max}. If it does, the maximum size is granted to the ONU; otherwise, the OLT grants the size as requested by the ONU. If an ONU has no more packets in the buffer upon receiving a GATE, it sends a REPORT with zero request size in reply. In the next cycle, it will not be granted, but still can get an opportunity to send its REPORT.

In this paper, we evaluate the performance of TCP over EPON to see if EPON is a good solution for metropolitan optical networks. The study is based on IPACT due to its improved utilization over the simple polling mechanism. We examine how EPON affects the behavior of TCP via analysis and simulation. Specifically, we model the performance of TCP over EPON with respect to the aggregate throughput and fairness. We calculate the maximum aggregate throughput of TCP flows on each of the upstream and downstream channels, and identify the fairness problem with existing polling schemes for EPON. We then derive two tight bounds of the fairness index for downloading users, and propose two polling mechanisms to promote fair resource sharing among multiple TCP flows. The performance of the proposed mechanisms is evaluated via ns-2 [4] simulations. The results show that our mechanisms can improve fairness among downloading TCP flows without degrading the aggregate throughput and fairness of uploading TCP flows.

The rest of the paper is organized as follows. In Section II, the behavior of TCP over EPON is analyzed. The fairness problem is also discussed. In Section III, two polling mechanisms are proposed to ensure fairness among multiple TCP flows in EPON. Finally, the paper is concluded in Section IV.

II. Performance Analysis for TCP over EPON

In this section, we examine the effects of the EPON MAC protocol on the performance of TCP. Since IPACT has better performance than simple polling, we will focus on the throughput and fairness performance of TCP over IPACT in the rest of the paper.

A. Notation and Assumptions

The notation used in the analysis is listed in Table I. We make the following assumptions:

- All ONUs in the system are doing TCP transfers, either downloading or uploading, but not both. That is, $N_d + N_u = N$.
- All TCP data packets are of the same length.
- The bottleneck of the system is in EPON.
- The propagation delays between the OLT and each ONU is identical.

B. Throughput

Each ONU gets a transmission opportunity in a cyclic order. Denote a cycle time by T_{cycle}, as shown in Fig. 6[1]. We assume that uploading ONUs always use up their transmission windows. The expectation of a cycle time is given by

$$E[T_{cycle}] = N \times T_{IFS} + E[N_{ack}] \times T_{ack} + E[N_u] \times \frac{W_{max}}{C}, \quad (1)$$

where $E[N_{ack}]$ is the average number of ACKs that each downloading ONU accumulates in its upstream buffer in a cycle time, and T_{ack} is the time to transmit an ACK packet, i.e., $T_{ack} = L_{ack}/C$. Note that without the assumption that each uploading ONU always sends a full W_{max} worth of packets, the last term of (1) will become $E[N_u] \times \frac{E[W]}{C}$ to reflect the bursty nature of traffic.

To compute $E[N_{ack}]$, we assume the maximum bandwidth utilization, i.e., the whole downstream channel is filled by GATEs, ACKs, and TCP packets, and a 12-byte inter-packet gap between any two consecutive Ethernet packets is not considered. Since the size of a TCP packet is relatively large, the number of TCP data packets transmitted by an uploading ONU in a cycle time is generally limited by the maximum size that each ONU can send in a cycle, i.e., W_{max}. Thus, an uploading ONU can transmit W_{max}/L_{data} packets in a cycle. If the delayed ACK option is enabled with parameter k (i.e., sending one ACK packet to acknowledge the receipt of k data packets), there are $\frac{E[N_u] \times W_{max}}{k \times L_{data}}$ ACK packets transmitted on the downstream channel in a cycle time. Under the assumption that the whole downstream channel is filled, the time period used to transmit the data packets of downloading ONUs in a cycle is given by

$$E[T_{cycle}] - \frac{E[N_u] \times W_{max}}{kL_{data}} \times \frac{L_{ack}}{C}.$$

In each cycle, the average number of data packets transmit-

TABLE I
Notation used in the analysis

N	the total number of ONUs in the system
N_d	the number of ONUs downloading files
N_u	the number of ONUs uploading files
C	the capacity of the downstream/upstream channel
L_{data}	the size of a TCP data packet
L_{ack}	the size of an ACK packet
T_{IFS}	the time for an ONU to turn on/off its laser, adjust the receiver gain, and synchronize the clock
T_{cycle}	the time interval of a transmission cycle
W_{max}	the maximum size each ONU can send in a cycle
K	delayed ACK parameter

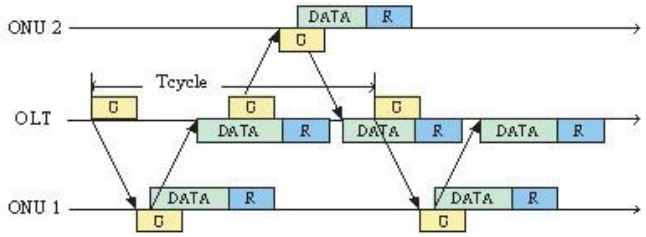


Fig. 3. An illustration of T_{cycle} for a 2-ONU system.

[1]The transmission times of REPORT and GATE messages are both negligible.

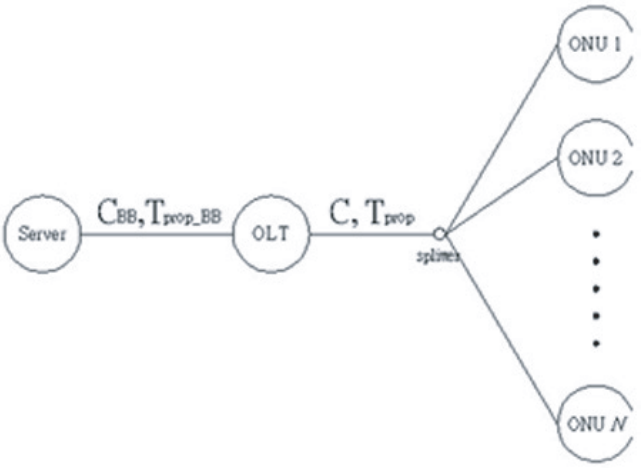


Fig. 4. Simulation model.

ted on the downstream channel is given by

$$\frac{E[T_{cycle}] - \frac{E[N_u] \times W_{max}}{kL_{data}} \times \frac{L_{ack}}{C}}{\frac{L_{data}}{C}}.$$

Considering the delayed ACK policy, the average number of ACK packets on the upstream channel in a cycle is

$$E[N_{ack}] = \frac{E[T_{cycle}] - \frac{E[N_u] \times W_{max}}{kL_{data}} \times \frac{L_{ack}}{C}}{k \times \frac{L_{data}}{C}}. \quad (2)$$

Substituting (2) into (1), we obtain the mean cycle time as

$$E[T_{cycle}] = \frac{N \times kL_{data} \times T_{IFS}}{kL_{data} - L_{ack}} - \frac{E[N_u] \times W_{max} \times (L_{ack}^2 - k^2 L_{data}^2)}{KCL_{data}(kL_{data} - L_{ack})}. \quad (3)$$

Thus, the maximum uploading and downloading throughputs can each be expressed as follows.

Uploading throughput:

$$\frac{E[N_u] \times W_{max}}{E[T_{cycle}]}; \quad (4)$$

Downloading throughput:

$$\frac{E[T_{cycle} - \frac{E[N_u] \times W_{max}}{kC} \frac{L_{ack}}{L_{data}}]}{E[T_{cycle}]} \times C, \quad (5)$$

where $E[T_{cycle}]$ is given by (1).

Next, we study how W_{max} affects the downloading throughput of each ONU. The maximum number of ACKs that each downloading ONU can send in a cycle is $\frac{W_{max}}{L_{ack}}$. In a cycle time, the maximum throughput of a downloading ONU is $\frac{W_{max}}{L_{ack}} \times \frac{kL_{data}}{E[T_{cycle}]}$. The maximum aggregate downloading throughput is $N_d \times \frac{W_{max}}{L_{ack}} \times \frac{kL_{data}}{E[T_{cycle}]}$. Thus, the maximum aggregate downloading throughput can be expressed by

$$min\{\frac{E[T_{cycle} - \frac{E[N_u] \times W_{max}}{kC} \frac{L_{ack}}{L_{data}}]}{E[T_{cycle}]} \times C, N_d \times \frac{W_{max}}{L_{ack}} \times \frac{kL_{data}}{E[T_{cycle}]}\}.$$

TABLE II
PARAMETERS USED IN THE SIMULATION

Parameter	Value
C	1Gbps
C_{BB}	10Gbps
T_{prop}	100 μs
T_{prop_BB}	200 μs
T_{IFS}	5 μs
L_{data}	534 bytes

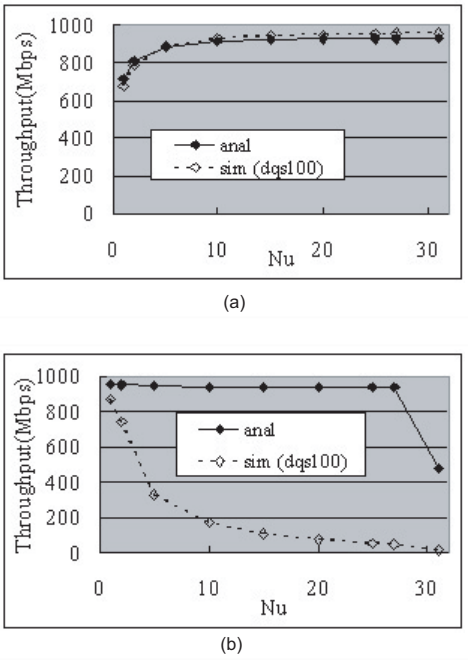
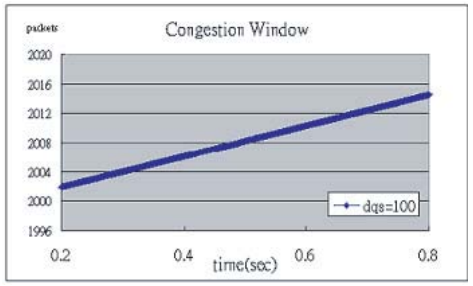


Fig. 5. Analysis vs. simulation results of aggregate TCP throughputs: (a) uploading ONUs; (b) downloading ONUs.

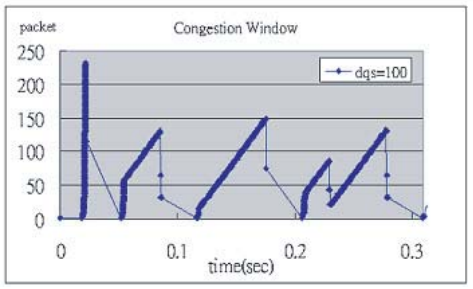
To examine if the channel capacity is fully utilized, we simulated TCP transfers over EPON via ns-2. The network topology and the parameters used in the simulation are shown in Fig. 4 and Table II, respectively. For simplicity, we assume the server is connected to the OLT directly. In Table II, C denotes the upstream data rate in EPON; C_{BB} and T_{prop_BB} represent the data rate and the propagation delay, respectively, in the backbone network; T_{prop} is the propagation delay in EPON, which makes the distance of the network about several kilometers.

In this simulation, the total number of ONUs is fixed at 32, the upstream and downstream data rates are both set to 1 Gbps, the delayed ACK parameter k is set to 1, and the maximum window size W_{max} is set to 64000 bytes. A TCP data packet size is set to 1024 bytes, and an ACK packet size, 64 bytes. The downstream buffer size is set to 100 packets.

Fig. 5 plots the aggregate throughputs for uploading and downloading ONUs, respectively. As the number of uploading ONUs increases, the aggregate uploading throughput increases and the aggregate downloading throughput decreases. Interestingly, the upstream channel is operated at the maximum aggregate throughput (i.e., the curves overlap), while the downstream channel is far under-utilized.

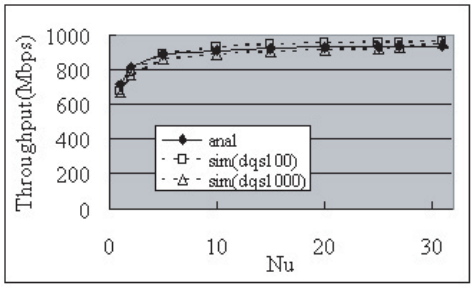


(a)

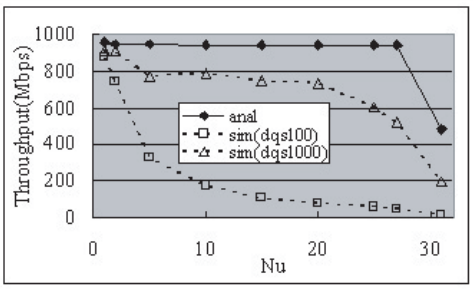


(b)

Fig. 6. Congestion window growth for uploading and downloading ONUs: (a) uploading ONUs; (b) downloading ONUs.



(a)



(b)

Fig. 7. The effect of different downstream buffer sizes on uploading and downloading throughputs: (a) uploading throughputs; (b) downloading throughputs.

To find out the problem of low utilization on the downstream channel, the TCP congestion windows of both of downloading and uploading ONUs are monitored, as shown in Fig. 6. The traces show that the windows of the uploading ONUs always increase linearly (Fig. 6(a)), while those of the downloading ONUs exhibit a saw-tooth pattern (Fig. 6(b)) and grow at a rate lower than the uploading ones. The saw-tooth pattern for downloading ONUs is caused mainly by the batch arrival of ACK packets from the downloading ONUs to the OLT, due to the polling mechanism of the EPON MAC layer. The batched arrivals of ACKs expand the congestion windows of TCP at the source quickly. When the congestion window size is large, a large volume of data packets burst from the server to the OLT, resulting in an overflow of the downstream buffer at the OLT. Thus, some data packets may be dropped. Worse, once a buffer overflow occurs, it is very likely that multiple data packets will be dropped. In TCP Reno, multiple packet losses often results in a timeout, thus degrading the throughput drastically. To prevent buffer overflow at downloading ONUs, we use the delay-bandwidth product of the channel between the OLT and the server as the minimum requirement of the downstream buffer in the OLT. For example, the link capacity between the OLT and the server is 10Gbps and the round trip propagation delay in the network is 600 μs as in the previous simulation. Then, the minimum buffer requirement is 750000 bytes, i.e., 750 packets.

Fig. 7 shows the simulation result with a 1000-packet downstream buffer, instead of 100 packets as in the previous simulation. We can see that the aggregate throughput of downloading ONUs is closer to the theoretically achievable throughput, while that of uploading ONUs remains intact. Fig. 8 shows the aggregate throughputs with different values of W_{max}, but the other parameters remain the same as in Fig. 7. We can see that a larger W_{max} results in a larger uploading throughput, but a lower downloading throughput. Thus, from this study, we learn that the aggregate downloading throughput is limited by the downstream buffer size, while the uploading throughput is limited by the maximum transmission size in a cycle, i.e., W_{max}.

C. Fairness

The fairness index f defined in [5] is given by

$$f = \frac{(\sum_{i=1}^{n} x_i)^2}{n(\sum_{i=1}^{n} x_i^2)}, \tag{6}$$

where n is the number of concurrent flows in the network and x_i is the throughput achieved by flow i, $1 \leq i \leq n$. The fairness index is always between 0 and 1. The larger the fairness index, the fairer the system. When the fairness index is equal to one, all flows in the network receive an equal share of the bandwidth.

Based on the same simulation setup as in Fig. 7, we obtain the fairness index for each of the downloading and uploading flows, as a function of N_u, in Fig. 9. Fig. 9(a) shows that the bandwidth is quite evenly shared by uploading ONUs. The reason is that the uploading ONUs always have packets to send in their buffers. As a result, they often request for the maximum size in their REPORT messages, and the OLT always grants each of them W_{max} in a cycle. Since each uploading ONU obtains the same transmission time in each cycle, the fairness index for uploading flows is close to one.

On the contrary, the fairness index for downloading flows is poor, as shown in Fig. 9(b). This is because the resource can be shared fairly among downloading flows only when the

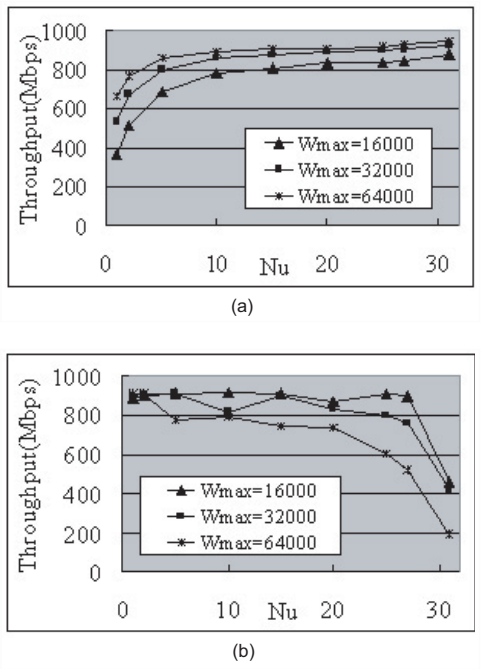


Fig. 8. The effect of different values of on uploading and downloading throughputs: (a) uploading throughputs; (b) downloading throughputs.

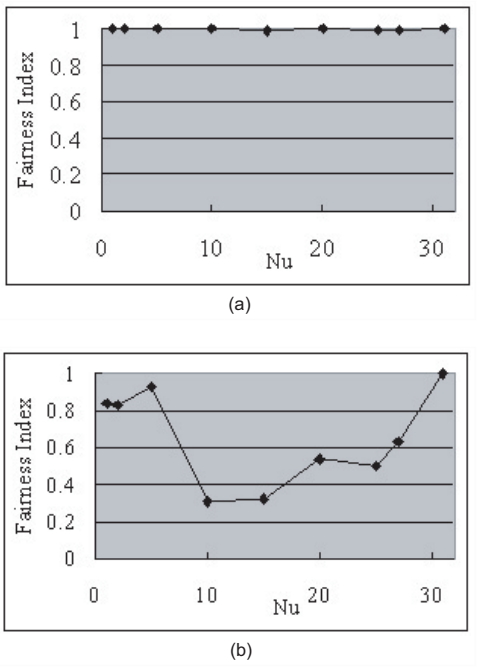


Fig. 9. Fairness among multiple flows for uploading and downloading ONUs: (a) uploading ONUs; (b) downloading ONUs.

TABLE III

TABLE III. THROUGHPUT OF EACH DOWNLOADING ONU (ND=6, NU=26)

ONU ID	Throughput (Mbps)
1	246.63
2	111.03
3	130.77
4	26.31
5	20.57
6	13.83

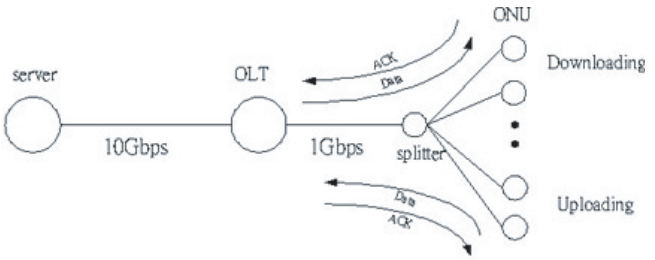


Fig. 10. The data flows of uploading and downloading ONUs in an EPON system.

number of downloading flows is much smaller than the number of uploading ones. When these two numbers are close, the fairness index drops lower than 0.6. For example, consider the case with 6 downloading ONUs and 26 uploading ONUs. For each uploading ONU, the average bandwidth obtained is 35.61 Mbps, the maximum and minimum bandwidths are 21.46 Mbps and 37.19 Mbps, respectively, and the fairness index is 0.993. However, for each downloading ONU, the average bandwidth is 91.52 Mbps, the maximum and minimum bandwidths are 246.63 Mbps and 13.83 Mbps, respectively, and the fairness index is only 0.549. The throughput achieved by each downloading ONU is listed in Table III. The low utilization caused by buffer overflow mentioned in Sec. II-B also induces the unfairness problem to the downloading flows. As shown in Table III, the downloading ONUs with smaller ONU IDs have larger bandwidth shares.

As shown in Fig. 10, each downloading ONU receives TCP packets on the downstream channel, and transmits ACK packets on the upstream channel. Due to the operation of EPON's MAC protocol, the ONUs can send multiple packets in a cycle in the uplink direction. When it is a downloading ONU's turn to transmit, the ONU can send at most W_{max}/L_{ack} ACKs into the upstream channel. Under the assumption that the backbone outside EPON is not the bottleneck, it is a reasonable assumption that all ACK packets arrive at the source in the same sequence as they depart from the OLT. Thus, the source of each downloading flow will receive multiple ACKs from its ONU. Those data bursts with destination ONUs being scheduled in the front of a transmission cycle are more likely to see an empty downstream buffer, while those near the end of a cycle are not. As a result, the dropping rates of those packets whose destination ONUs are scheduled near the end of a transmission cycle are higher. Worse, TCP data packets arrive at the OLT in batches; hence, if a buffer overflows, it is very likely that more than one packet will be dropped. In the Reno version of TCP, which is widely used in the Internet nowadays, multiple-loss of packets in a congestion window will cause a timeout event, shrinking the congestion window size to one. As a consequence, the performance is severely degraded.

Next, we derive the two bounds of the fairness index for IPACT. For simplicity, we assume that upon receiving an ACK packet, the OLT sends a data packet on the downstream channel, and upon receiving a data packet, the OLT sends

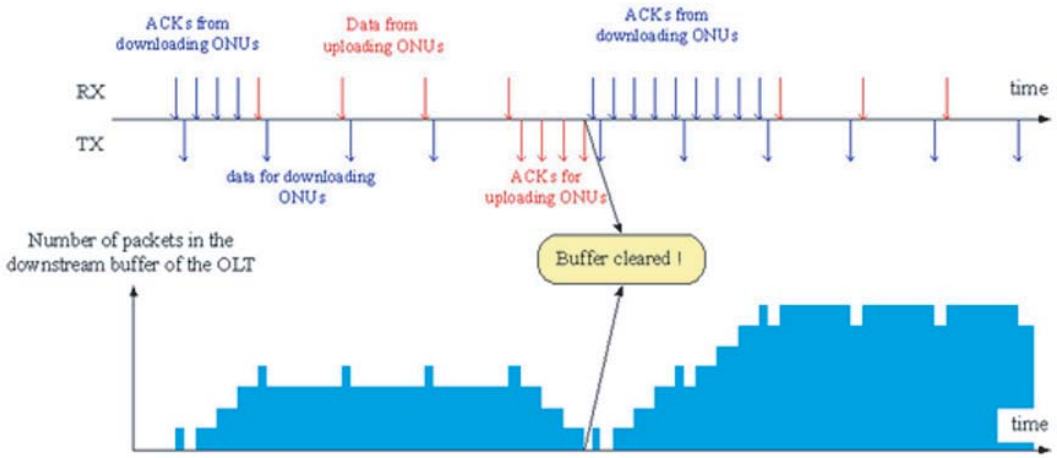


Fig. 11. The buffer occupancy of the OLT for a series of arrivals and departures of packets.

an ACK packet in reply. Fig. 11 shows the buffer occupancy of the OLT for a series of arrivals and departures of TCP data and ACK packets. Packets arrive at the receiving (Rx) interface of the OLT (i.e., from the upstream channel) and depart from the transmission (Tx) interface of the OLT (i.e., to the downstream channel). We observe that during the period that the OLT receives a batch of ACK packets, the buffer grows due to using a longer time to transmit data packets; during the period that the OLT receives a batch of data packets, the buffer decreases due to using a shorter time to send ACK packets. Thus, a downloading ONU following an uploading ONU obtains a larger share of bandwidth than a downloading ONU following a downloading ONU because its packets tend to see a larger effective buffer at the OLT.

Consider an extreme case with a cyclic sequence of $(DD...DUU...U)$, where D represents a downloading ONU and U is an uploading ONU. That is, each downloading ONU follows another downloading ONU, except for the first one (i.e., ONU_1), which follows an uploading ONU. If the batch of data packets from the uploading ONUs is large enough, the downstream buffer can be cleared.

Assume that the first packet of the first ONU (i.e., ONU_1) in a cycle arriving in the downstream buffer sees an empty queue. Let $CW_{max}^{(1)}$ denote the maximum number of packet bursts the server can send for ONU_1 in a cycle without a loss, thus $CW_{max}^{(1)} = B_d$, where B_d is the downstreaming buffer size. Once the congestion window expands beyond $CW_{max}^{(1)}$, packet losses occur. The congestion window may either be reduced by half (single loss) or drop to one (multiple losses, i.e., timeout). Thus, the average window size of ONU_1 is $3B_d/4$ (i.e., $\frac{B_d + B_d/2}{2}$) if each loss is assumed a single loss, or $B_d/2$ (i.e., $(B_d + 1)/2$) if all losses are assumed multiple losses. The first packet of the second ONU (i.e., ONU_2) in a cycle arriving at the downstream buffer sees a buffer with an effective size of $B_d/4$ (for single loss) or $B_d/2$ (for multiple losses), and its average window size is $3B_d/16$ (for single loss) or $B_d/4$ (for multiple losses). Generally, the average window size of ONU_i is $\frac{3}{4^i}B_d$ (for single loss) or $\frac{1}{2^i}B_d$ (for multiple losses).

Assume that the average RTT is approximately identical for all downloading ONUs. We have $\frac{Throughput_i}{Throughput_j} = \frac{aveCW_i}{aveCW_j}$ for all downloading flows i and j. Thus, we can use the average congestion window to predict the fairness. The two bounds of the fairness index for downloading flows can be expressed as follows.

Upper Bound:

$$f_{upper} = \frac{(\sum_{i=1}^{N_d}(\frac{3}{4^i})B_d)^2}{N_d(\sum_{i=1}^{N_d}(\frac{3}{4^i})B_d)^2} = \frac{9(1-(\frac{1}{4})^{N_d})^2}{15N_d(1-(\frac{1}{16})^{N_d})}, \quad (7)$$

Lower Bound:

$$f_{lower} = \frac{(\sum_{i=1}^{N_d}(\frac{1}{2^i})B_d)^2}{N_d(\sum_{i=1}^{N_d}(\frac{1}{2^i})B_d)^2} = \frac{(1-(\frac{1}{2})^{N_d})^2}{3N_d(1-(\frac{1}{4})^{N_d})}, \quad (8)$$

To verify the analytical result for fairness, we again simulated TCP transfers via ns-2 with the following parameters: C = 1Gbps, C_{BB}=10Gbps, T_{prop} = 100 μs, T_{propBB} =200 μs, T_{IFS} = 5 μs, L_{data} = 1000 bytes, L_{ack} = 64 bytes, and W_{max} = 64000 bytes.

Fig. 12 plots the fairness indices for both of the analytical and simulation results. The fairness index is defined by

$$f = \frac{(\sum_{i=1}^{N_d} x_i)^2}{N_d \sum_{i=1}^{N_d} x_i^2},$$

where x_i refers to the throughput of ONU_i in the simulation and to the average congestion window of ONU_i in the analysis. The solid curve with squares (i.e., anal(0.5)) indicates that when a loss occurs, TCP enters the slow start phase and the congestion window is set to one (i.e., multiple losses leading to a timeout); the solid curve with triangles (i.e.,

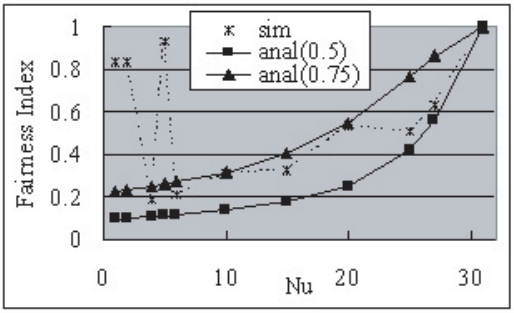


Fig. 12. Two analytical bounds vs. simulation results.

anal(0.75)) indicates that when a loss occurs, TCP remains in the congestion avoidance phase and the congestion window size is halved (i.e., single loss, detected by three duplicate ACKs).

We can see that the two analytical curves provide two tight bounds for the simulation results when the number of uploading ONUs is large. The abnormal effect at small N_u is because when there are few uploading ONUs, the downstream buffer may not be cleared up in the beginning of each cycle, i.e., after all uploading ONUs' transmissions in each cycle. Consequently, the occupation of the downstream buffer never falls to zero, and the first packets from all downloading ONUs in each cycle have an equal opportunity to see a non-filled buffer in the steady state.

III. Two Polling Mechanisms for IPACT-Based EPONs

In this section, two polling mechanisms are proposed to improve the fairness problem in IPACT-based EPONs. Simulations were conducted via ns2 to evaluate the performance of the proposed mechanisms. The results show that our mechanism can improve fairness among downloading TCP flows without degrading their aggregate throughput.

A. Problem Statement

Consider that there are N_d downloading ONUs and N_u uploading ONUs. Uploading ONUs transmit data packets on the upstream channel, while downloading ONUs transmit ACK packets on the upstream channel. The uploading ONUs often send "W_{max}" worth of packets in a cycle. Thus, the average number of packets sent by each of them is W_{max}/L_{data}. When these data packets arrive at the server, the server will send "W_{max}/L_{data}" worth of ACK packets. The number of these ACKs is typically in the order of ten, which is unlikely to overflow the downstream buffer in the OLT. Even if an overflow occurs and several ACK packets are dropped, the throughput will not seriously degrade, thanks to the cumulative ACK mechanism used in TCP. However, this is not the case when it comes to downloading ONUs. The size of an ACK packet is 64 bytes. Thus, an ONU can transmit thousands of ACK packets in a cycle. When a burst of ACK packets arrives at the server, a burst of data packets will be sent, which may cause an overflow. ONUs in different positions in a cycle will perceive available buffer of different sizes. This causes unfairness in bandwidth sharing among downloading ONUs.

TABLE IV

TABLE IV. AN EXAMPLE OF POLLING SEQUENCES IN FDP

$1^{st} cycle$	1 2 3 4
$2^{nd} cycle$	2 3 4 1
$3^{rd} cycle$	3 4 1 2
$4^{th} cycle$	4 1 2 3
$5^{th} cycle$	1 2 3 4

Two polling mechanisms are proposed below to solve the fairness problem among TCP downloading flows in the IPACT-based EPON.

B. Fixed Deferment Polling (FDP)

The first mechanism we propose is called Fixed Deferment Polling (FDP). With FDP, the OLT in each cycle performs a circular left shift to the polling sequence in the previous cycle. In other words, the first ONU in the current cycle will become the last ONU in the next cycle, while the sequence of the other ONUs remains intact. For example, if there are four ONUs, the polling sequences of ONUs in five cycles are shown in Table IV.

Fig. 13 illustrates how FDP works, as compared to IPACT, where Gxy represents a GATE message sent to ONU$_x$ in the yth cycle, and Rxy, a REPORT message from ONU$_x$ to request a transmission chance in the yth cycle. Consider a four-ONU system. With IPACT, the OLT sends one polling message to each ONU in a fixed sequence, say, ONU$_1$, ONU$_2$, ONU$_3$, and ONU$_4$. With FDP, the OLT sends a GATE to each ONU in a circular left shift to the receiving sequence of REPORT messages in each cycle (i.e., the polling sequence in the previous cycle). Suppose that the polling sequence in Cycle 1 is ONU$_1$, ONU$_2$, ONU$_3$, and ONU$_4$. On receipt of the first REPORT, the OLT temporarily sets aside the GATE (i.e., G12) to ONU$_1$. A GATE message is sent to each of the successive ONUs (i.e., ONU$_2$, ONU$_3$, ONU$_4$) when their REPORT messages are received, as in IPACT. The OLT sends G12 to ONU$_1$ right after sending G42 to ONU$_4$. Since G22 is sent before the next cycle starts and G12 is sent before the next cycle ends, the upstream bandwidth is still fully utilized.

C. Random Deferment Polling (RDP)

In spite of the simplicity of FDP, the relative polling sequence of ONUs stays intact long term, i.e., the original sequence repeats every N cycles in an N-ONU system. For example, the repetition of polling sequences in a 4-ONU system is (1234, 2341, 3412, 4123, 1234, ...). Thus, for a total of N ONUs, if the original sequence is $DD...DUU...U$, this polling pattern repeats every N cycles.

To truly randomize the relative sequences of ONUs, we propose another mechanism called Random Deferment Polling (RDP). With RDP, the OLT is associated with a parameter called N_{def}. Consider an N-ONU system. In each cycle, the OLT randomly selects a number between 1 and N_{def}, say, d, and defers the GATE for the first ONU $(N - d)$ times.

For instance, there are 4 ONUs and N_{def} is 3 in the system shown in Fig. 14, using the same notation as in Fig. 13.

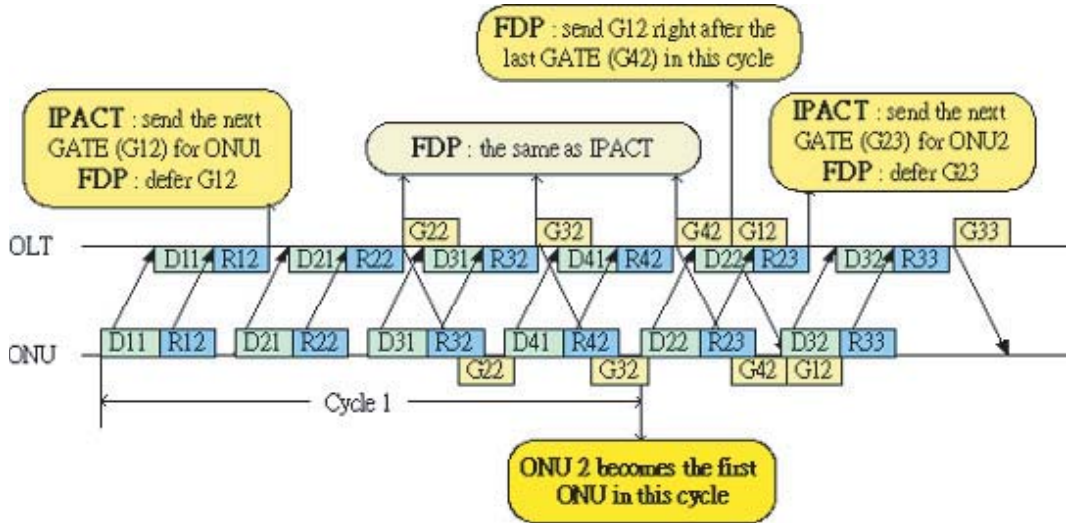


Fig. 13. The operation of FDP.

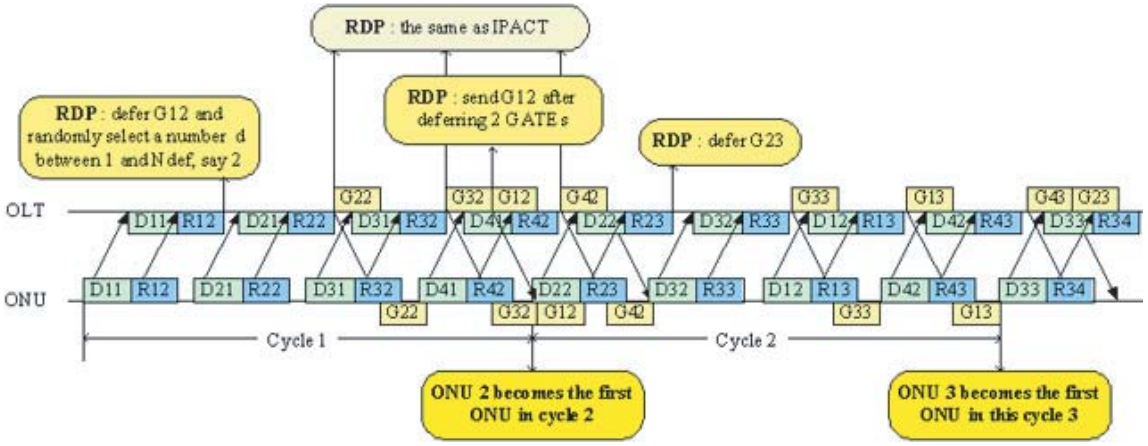


Fig. 14. The operation of RDP.

TABLE V

TABLE V. AN EXAMPLE OF POLLING SEQUENCES IN RDP

$1^{st}cycle$	1 2 3 4	select 2
$2^{nd}cycle$	2 3 **1** 4	select 1
$3^{rd}cycle$	3 1 4 **2**	select 3
$4^{th}cycle$	1 **3** 4 2	select 1
$5^{th}cycle$	3 4 2 **1**	select 2

Assume that the polling sequence in Cycle 1 is ONU_1, ONU_2, ONU_3, and ONU_4. When receiving R12 from ONU_1 in Cycle 1 (i.e., the first ONU in cycle 1), the OLT randomly selects a value between 1 and 3, say, 2, and defers ONU_1 two (i.e., 4-2) times in the next cycle. Thus, the polling sequence in Cycle 2 is G22, G32, G12, G42. Similarly, when receiving R23 from ONU_2 (i.e., the first ONU in cycle 2), the OLT randomly selects another value between 1 and 3, say, 1, and defers ONU_2 three (i.e., 4-1=3) times in the next cycle. Thus, the polling sequence in Cycle 3 becomes G33, G13, G43, G23. More examples are shown in Table V.

Note that the mechanism proposed in [6], which is designed to share upstream bandwidth of optical access networks and performs XOR operation on a bit-by-bit basis, still fails in solving the extreme case addressed in this paper. Consider the example that there are N ONUs downloading files and N ONUs uploading files and the original polling sequence is $DD...DUU...U$. With this scheme, the sequence is still $DD...DUU...U$ in the next cycle. In cycle (N+1), the sequence is changed to $UU...UDD...D$ and in cycle (2N+1), the sequence reverts back to $DD...DUU...U$ again. In other words, it alternates between the sequences $DD...DUU...U$ and $UU...UDD...D$ every N cycles. This is because if the least significant bit of counter B changes, it will only change the position of two adjacent ONUs, and the relative sequence of the groups of downloading ONUs or uploading ONUs stay unchanged (i.e., the sequence of $DD...DUU...U$ still remains). If the most significant bit changes, the group of downloading ONUs and the group of uploading ONUs switches their position in the sequence (i.e., $DD...DUU...U$ –> $UU...UDD...D$). Therefore, it still fails to solve the problem addressed in our paper. In contrast with this scheme, RDF defers the GATE for the first ONU in each cycle and successfully destroys the polling pattern in the extreme case described in the paper.

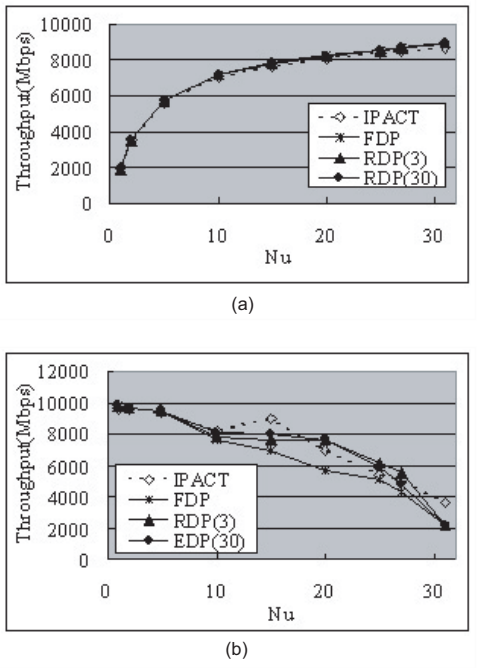


Fig. 15. Aggregate throughputs of different approaches: (a) uploading throughputs; (b) downloading throughputs.

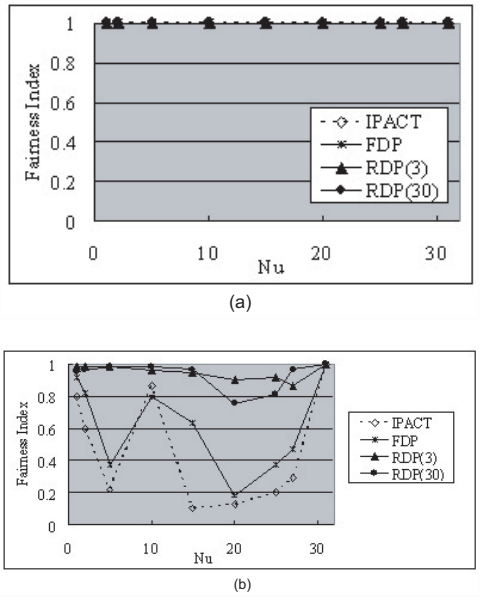


Fig. 16. Fairness indices of different approaches: (a) uploading throughputs; (b) downloading throughputs.

D. Simulation Results

In this simulation, the parameters were set as follows: N = 32, W_{max}= 64000 bytes, T_{IFS} = 5μs , L_{data} = 1000 bytes, L_{ack} = 64 bytes, and B_d = 1000 packets. We compare Fixed Deferment Polling (FDP) and Random Deferment Polling (RDP) with IPACT. We have two different settings of the parameter N_{def} for RDP. The curve with triangles is for RDP with N_{def} set to 3, and that with diamonds is with N_{def} set to 30.

Fig. 15 shows the aggregate throughputs for both of the uploading ONUs (Fig. 15(a)) and downloading ONUs (Fig. 15(b)). We find that all three mechanisms have fully utilized the uploading channel and have similar performance with respect to the aggregate downloading throughput. Among them, RDP outperforms FDP, thanks to that with FDP, the ordering among all ONUs remains unchanged long term. RDP, on the other hand, is a truly random mechanism, making the resource sharing fairer. Note that the hump in IPACT's throughput when N_u is 15 in Fig. 15(b) is probably caused by random simulation anomalies.

Fig. 16 shows the fairness indices for both of the uploading (Fig. 16(a)) and downloading (Fig. 16(b)) flows. The fairness indices of the uploading flows for all the three mechanisms are all close to 1, indicating that the upstream bandwidth is fairly shared by all uploading ONUs. Fig. 16(a) shows that FDP does not have good improvement in the fairness among downloading flows, again due to the global ordering still being maintained. RDP improves the fairness performance greatly; the fairness index stays above 0.75 when N_{def} is 3.

The effects of different N_{def} values are also examined. In Figs. 15 and 16, when N_{def} is 30, the performance with respect to aggregate uploading and aggregate downloading throughputs is compatible to RDP(3). The fairness index of uploading flows are also close to one. However, we find that the fairness index of downloading flows is not as good as RDP(3). This is because the basic concept of RDP is to put the first ONU in this cycle to the last few ONUs in the subsequent cycle. If N_{def} is large, although the randomness increases, the first ONU in this cycle may still be the first few ONUs in the subsequent cycle. As a consequence, it may take longer for all downloading ONUs to have an identical throughput.

IV. Conclusions

In this paper, we study TCP performance over EPON with respect to throughput and fairness. The maximum aggregate throughputs for both of downloading and uploading flows are derived, and two tight bounds of the fairness index are obtained. We also propose two polling mechanisms to solve the unfairness problem, and evaluate their performance by ns-2 simulations. The results show that the proposed mechanism can improve the fairness among downloading TCP flows without degrading the aggregate throughput and the fairness of the uploading TCP flows.

In this paper, we have not captured the packet loss effect on TCP performance in our analytical model. In the future, we will consider this problem and modify our model to reflect this effect.

References

[1] G. Kramer and G. Pesavento, "Ethernet passive optical network (EPON): building a next generation optical access network," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 66–73, Feb. 2002.

[2] IEEE, IEEE 802.3ah Ethernet in the First Mile, `http://grouper.ieee.org/groups/802/3/efm/`

[3] G. Kramer, B. Mukherjee, and G. Pesavento, "IPACT: a dynamic protocol for an Ethernet PON (EPON)," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 74–80, Feb. 2002.

[4] UCB/LBNL/ISI/VINT Network Simulator - ns, Version 2, `http://www.isi.edu/nsnam/ns/`

[5] R. Jain, "Throughput fairness index: an explanation," *ATM Forum Contribution 99-0045,* Feb. 1999.

[6] N. Miki, A. Otaka, N. Tamaki, and R. Watanabe, "Optical access systems suitable for computer communication," in *Proc. GLOBECOM'98*, Access Networks Mini Conference.

Kai-Chien Chang received the B.S. and M.S. degrees in Electronic Engineering from the Department of Electrical Engineering, National Taiwan University in 2001 and 2003, respectively. She is currently an engineer in Media-Tek Inc. in Taiwan. Her research interest is in wireless communications.

Wanjiun Liao received her BS and MS degrees from National Chiao Tung University, Taiwan, in 1990 and 1992, respectively, and the Ph.D. degree in Electrical Engineering from the University of Southern California, Los Angeles, California, USA, in 1997. She joined the Department of Electrical Engineering, National Taiwan University (NTU), Taipei, Taiwan, as an Assistant Professor in 1997. Since August 2005, she has been a full Professor. Her research interests include wireless networks, multimedia networks, and broadband access networks.

Dr. Liao is currently an Associate Editor of *IEEE Transactions on Wireless Communications* and of *IEEE Transactions on Multimedia*. She was a tutorial co-chair of IEEE INFOCOM 2004, and the TPC vice chair of IEEE Globecom 2005 (Autonomous Network Symposium). Dr. Liao has received many research awards. Papers she co-authored with her students received the Best Student Paper Award of the First IEEE International Conferences on Multimedia and Expo (ICME) in 2000, and the Best Paper Award of the First International Conference on Communication, Circuits and Systems in 2002. She was awarded K. T. Li Young Researcher Award of the ACM Taipei Chapter for her research achievements in 2003. Dr. Liao was elected as one of Ten Distinguished Young Women in Taiwan in 2000.