PAPER

Network-Supported TCP Rate Control for the Coexistence of Multiple and Different Types of Flows on IP over PLC

Adriano MUNIZ[†a)], Kazuya TSUKAMOTO[†b)], Masato TSURU[†c)], *Members*, *and* Yuji OIE[†d)], *Fellow*

SUMMARY With the approval of IEEE 1901 standard for power line communications (PLC) and the recent Internet-enable home appliances like the IPTV having access to a content-on-demand service through the Internet as AcTVila in Japan, there is no doubt that PLC has taken a great step forward to emerge as the preeminent in-home-network technology. However, existing schemes developed so far have not considered the PLC network connected to an unstable Internet environment (i.e. more realistic situation). In this paper, we investigate the communication performance from the end-user's perspective in networks with large and variable round-trip time (RTT) and with the existence of cross-traffic. Then, we address the problem of unfair bandwidth allocation when multiple and different types of flows coexist and propose a TCP rate control considering the difference in terms of end-to-end delay to solve it. We validate our methodology through simulations, and show that it effectively deals with the throughput unfairness problem under critical communication environment, where multiple flows with different RTTs share the PLC and cross-traffic exists on the path of the Internet.

key words: *power line communications (PLC), TCP rate control, multiple and different types of flows, unfairness problem*

1. Introduction

Power Line Communication (PLC) is a technology that makes use of existing power lines for data transmission. It is already in use but still questioned regarding whether or not it can offer a viable alternative to the predominant existing Wi-Fi and wired Ethernet technologies for in-home networking. Moreover, considering PLC and Wi-Fi, it is known that both of them face difficulties to provide a better communication quality to the users inside a large house because of their degradation factors [1].

It is expected that PLC will be widespread worldwide because of the acceptance of IEEE 1901 standard for power line communications [2] in September 2010 and the recent start of certification of PLC-related devices that comply with such standard [3]. In addition, despite of the diversified number of broadband power line technologies and their interoperability issue [4], in July 2011 HD-PLC Alliance announced that the world first IEEE 1901 compliant LSI is now ready to deploy to the market [3]. Thus, although PLC has not reached the mass market diffusion so far, there has been a great progress to accelerate the adoption of IEEE 1901 standard compliant products in the world.

The necessity of improving the electric grid infrastructure and the management of energy transmission and distribution has motivated a global push for Smart Grids. Among the many types of communication technologies that Smart Grid will probably make use of, PLC is a very good candidate because of its features and continuous improvement over the past few years [5], [6]. This is confirmed by the great progress that PLC has made for supporting Automatic Meter Reading (AMR) and Advanced Metering Infrastructure (AMI) applications [7]. Those are the reasons why we believe PLC is a promising technology on which to base the in-home-network.

Furthermore, nowadays it is increasing the demands for more convenience at in-home-network and in PLC there is nothing else to be built or installed in a house because the infra-structure needed to deploy this technology is already installed. However, like Wi-Fi, which faces problems such as interference, overcrowded spectrum and Ethernet that faces construction cost problems, PLC has its drawbacks as well. The fluctuations of source impedance, power level attenuation and the background noise caused by the electrical appliances are among the main ones. In order to overcome the above-mentioned problems related to PLC and get better communication performance, the authors proposed in [8] a TCP rate control method that was shown to be effective on an IP over PLC network environment. Nevertheless, in such environment neither large difference in RTTs of coexisting flows nor cross-traffic was addressed. Therefore, in this paper we take as reference the network-supported TCP rate control scheme for high-speed power line communications environment [8] and make a comparative study to investigate the end-to-end flow-level performance of IP over PLC through an unstable Internet environment (i.e. more realistic situation). Then, we propose a new scheme to deal with the unfair bandwidth allocation caused by the coexistence of multiple TCP flows and VoIP applications, whose voice quality depends on many parameters, such as one-way delay, jitter, packet loss rate, codec, voice data length [9]. Regarding the QoS requirements for VoIP applications, we restricted our investigations to one-way delay because in our view the time needed for a packet to cross the network from a source to the destination host is the key parameter that should be mainly investigated.

This paper starts with a brief explanation of the related works in Sect. 2 and an overview of the HD-PLC MAC in Sect. 3. In Sects. 4.1 and 4.2 we explain about the exper-

Manuscript received October 18, 2012.
Manuscript revised May 12, 2013.

[†]The authors are with Kyushu Institute of Technology, Iizuka-shi, 820-8502 Japan.
a) E-mail: adriano@infonet.cse.kyutech.ac.jp
b) E-mail: tsukamoto@cse.kyutech.ac.jp
c) E-mail: tsuru@cse.kyutech.ac.jp
d) E-mail: oie@cse.kyutech.ac.jp

DOI: 10.1587/transcom.E96.B.2587

iments we have made and the simulator of PLC network we have developed. The previous scheme is described in Sect. 5, then we address the throughput unfairness problem in coexisting multiple TCP flows and VoIP end-to-end delay problem in TCP/VoIP flows. We dedicate Sect. 6 to describe detailed procedure of our proposed scheme. In Sect. 7 we demonstrate the effectiveness of the proposed scheme through simulation results. Finally, Sect. 8 summarizes our contributions.

2. Related Work

The majority of the previous studies and works on PLC networks have been done on PHY/MAC layer. However, just a little attention has been given to the evaluation of these networks from an end-to-end perspective.

In literature [10] the construction of a PLC Test Bed with 5 nodes was described. With the objective of identifying the MAC and PHY limitations of this equipment, tests including uplink and downlink of TCP and UDP traffic simulated by iperf were realized.

In literature [11] the authors propose a scheme to estimate an achievable throughput for TCP communication limited to the home network and adaptively control the transmission rate of TCP to prevent buffer overflow at the PLC Sender.

Mizutani et al. [8] extends the work presented in literature [11] to a network-wide where the clients are connected to the servers related to a different network through Internet. However, to the best of our knowledge there are no studies that have focused on the evaluation of IP over PLC under adverse aspects like network congestion and cross traffic that contributes to the instability (e.g. large and variable RTT) of the flows in the Internet environment from an end-user perspective.

Thus, our research mainly differs from [8] in the following aspects: In order to investigate the above-mentioned adverse aspects in an Internet environment (i) we evaluate both multiple TCP and TCP/VoIP flows behavior with large difference in round trip time (RTT) and (ii) the existence of cross-traffic for a certain amount of time during the communication period. Based on the obtained results, we propose a new scheme to solve the problems.

3. HD-PLC MAC

The PLC technology under investigation in our study is HD-PLC (High-Definition Power Line Communication) and in this section we will provide an overview of the HD-PLC Media Access Control (MAC) for a better understanding of the proposed approach, which will be detailed later in this paper. Figure 1 shows the MAC frame structure of HD-PLC. The IP packets sent from sender are loaded in the transmit queue of PLC sender modem as service data unit (SDU) in PLC. Moreover, in order to reduce communication overhead multiple SDUs, whose number is determined based on the network condition, are concatenated in a container block

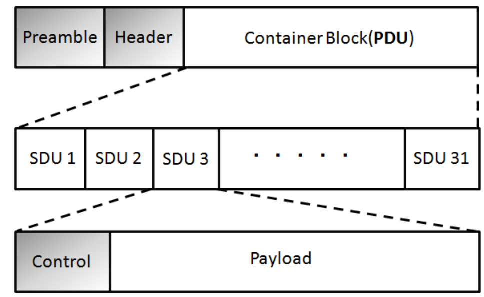


Fig. 1 HD-PLC MAC structure [12].

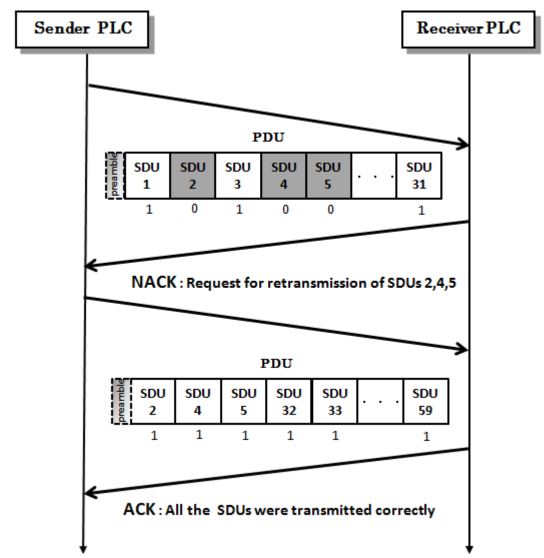


Fig. 2 SDU retransmission mechanism.

called protocol data unit (PDU) until the transmission timer expires or the 31 SDUs are received.

In addition, to improve the transmission efficiency, an error recovery mechanism is employed. The retransmission ARQ (Auto Repeat Request) is done by using the PLC-ACK, which has a bit field that informs each one of the transmitted SDUs was transmitted correctly (ACK-ACKnowledgment) through the bit field 1 or if a retransmission is necessary (NACK-Negative ACKnowledgment) through the bit field 0. Figure 2 shows that the receiver identifies the existence of errors in SDUs 2, 4, 5 transmitted by the PLC sender. Consequently, the receiver sends a PLC-NACK to request the retransmission of SDUs 2, 4, 5. After receiving the PLC-NACK from the receiver, PLC sender immediately retransmits all the SDUs indicated by the PLC-NACK and the new SDUs loaded in the transmit queue, up to the maximum number of SDUs permitted [12]. Following this, in case there is no error, the PLC receiver returns a PLC-ACK.

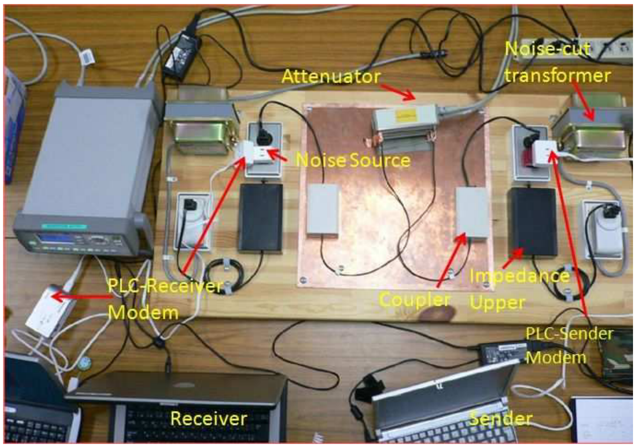


Fig. 3 Experimental environment.

4. Simulation of PLC Network

4.1 Experimental Environment

Since there is no standard model of auto-tuned PHY rate and packet-level error patterns over PLC, we first conducted preliminary experiments using real production-level PLC modems on a test-bed environment to experimentally obtain those essential parameters for realistic simulation settings. That is, the objective of our study is not to modify the HD-PLC standard but to investigate the end-to-end flow-level performance of IP over PLC without changing the features of BL-PA510, a HD-PLC based modem provided by Panasonic Systems Network Corporation, which was used in our experiments.

Our experiment was realized with the use of an experimental environment that avoids interference from outside noise. The experimental environment is shown in Fig. 3, where the sender and receiver communicate to each other through the PLC-Sender modem and PLC-Receiver modem. We added different levels of attenuation and some types of noise-source to the power line to measure the packet-level error rate and the PHY rate over PLC under different situations. However, the pattern that we used to be simulated by our simulator was the combination of the packet-level error rate and PHY rate for inverter light with the attenuation level 45dB because it significantly impacts the performance of home network. In this work we would rather give attention to an unstable communication environment than a stable one because PLC network is subject to many degradation factors.

4.2 Simulation

The network topology used in our study is shown in Fig. 4, and the parameters used are given in Table 1.

We assume that sender1 is located inside Japan, sender2 is in the USA and sender3 is in China. Then, the propagation delays implemented in our study between sender1, sender2, sender3 and the PLC_modem1 are 30 ms, 90 ms and 60 ms, respectively. In addition, the propagation delay assumed between receiver1, receiver2, receiver3 and the PLC_modem2 is 0.3 ms, and we further assume the Internet and the PLC bandwidth to be 100 Mbps and 18.62 Mbps, respectively.

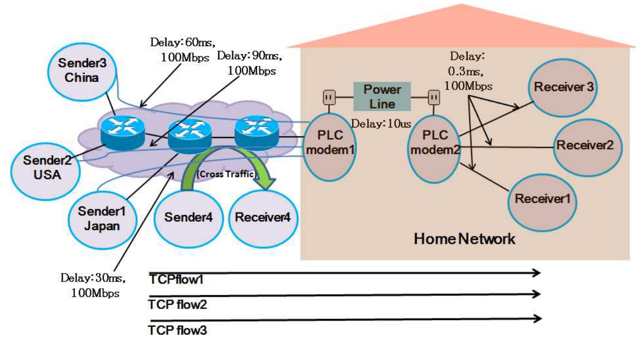


Fig. 4 Network topology I.

Table 1 Simulation parameters.

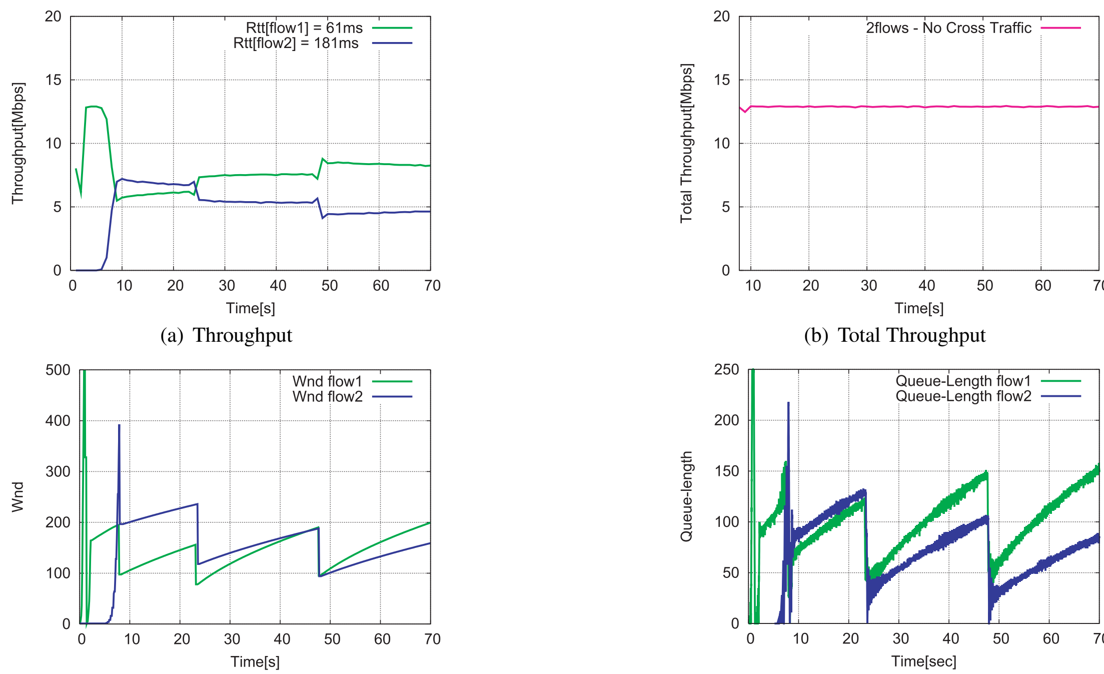
Transport Protocol	TCP NewReno/SACK, UDP
PLC bandwidth (Phy_Rate)	18.62 Mbps
Internet bandwidth	100 Mbps
Buffer Size	256 packets
Packet Size	1500 bytes
Delay	Sender1(JAPAN)-PLC_modem1 30 [ms] Sender2(USA)-PLC_modem1 90 [ms] Sender3(CHINA)-PLC_modem1 60 [ms] PLC_modem2-Receiver 0.3 [ms]
Error Pattern	Inverter Light Att:45 dB

We developed a module on the NS-2 network simulator, which is a discrete event-driven network simulator directed to the networking research community. Besides, the data (Phy_rate and packet level error rate) we previously obtained in our PLC experimental environment (Fig. 3) were loaded in our NS-2 module. Thus, we could simulate the realistic behavior of how data are sent through power line communication under different power level attenuation and different noise sources.

We set packet size as 1500 bytes TCP communication. Besides, we added CBR cross traffic with rate of 90 Mbps that is injected along the communication path between sender4 and receiver4. Moreover, we assumed large RTT difference among the coexisting flows in order to simulate the behavior of a real Internet environment making use of the network topology shown in Fig. 4. In network topology I (Fig. 4) we have a scenario where the maximum of 3 TCP flows coexist on IP over PLC network with the presence of CBR cross traffic in the Internet environment.

5. A Previous Scheme — Network Supported TCP Rate Control Scheme For High-Speed PLC Environment

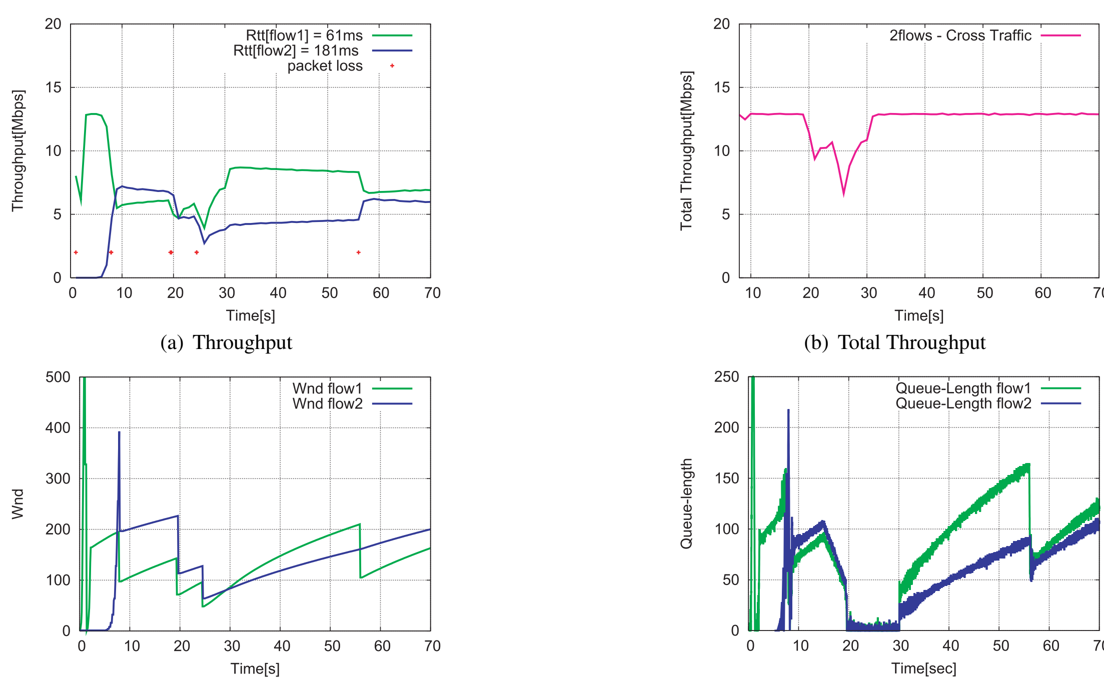
This reference scheme was proposed by Mizutani et al. [8] with the purpose to extend the work in literature [11] in or-



(a) Throughput (b) Total Throughput

(c) Wnd (d) Queue Length

Fig. 5 TCP NewReno/SACK: 2TCP flows (w/o cross traffic).



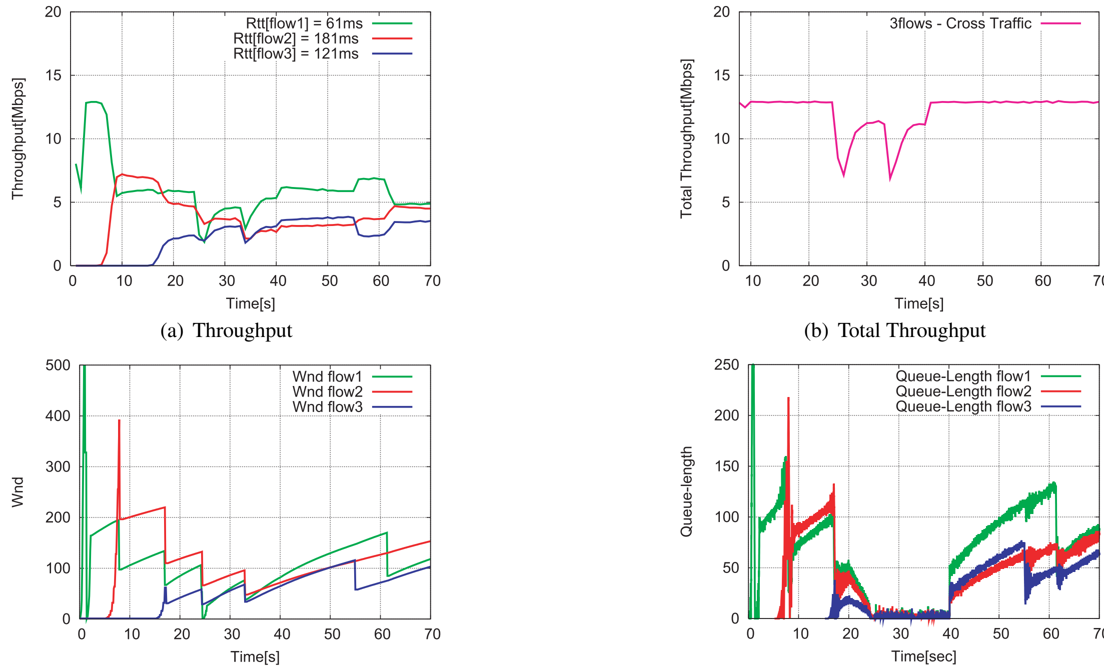
(a) Throughput (b) Total Throughput

(c) Wnd (d) Queue Length

Fig. 6 TCP NewReno/SACK: 2TCP flows (w/ cross traffic).

der to maximize the throughput over PLC connected to an environment where neither RTT difference nor cross traffic over the Internet was taken into consideration. The scheme is the same idea of the TCP protocol. In TCP, the receiver informs its buffer-capacity to the sender through *awnd* every time the receiver sends the TCP-ACK. Therefore, the sender's window-size *wnd* will take the smallest value between congestion window *cwnd* and advertised window *awnd* to provide a way to avoid network congestion and overflow of the receiver side.

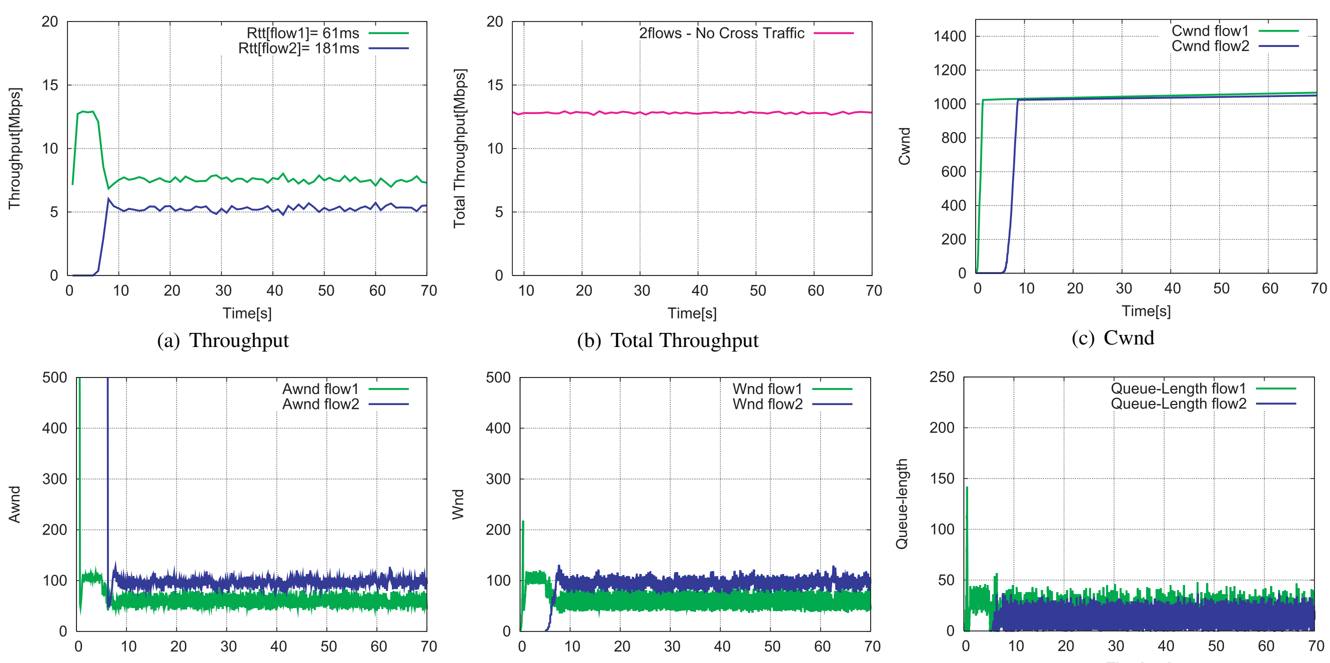
The scheme in discussion modifies the *awnd* for each TCP-ACK packet in PLC_modem1 only in case the value set



(a) Throughput (b) Total Throughput

(c) Wnd (d) Queue Length

Fig. 7 TCP NewReno/SACK: 3TCP flows (w/ cross traffic).



(a) Throughput (b) Total Throughput (c) Cwnd

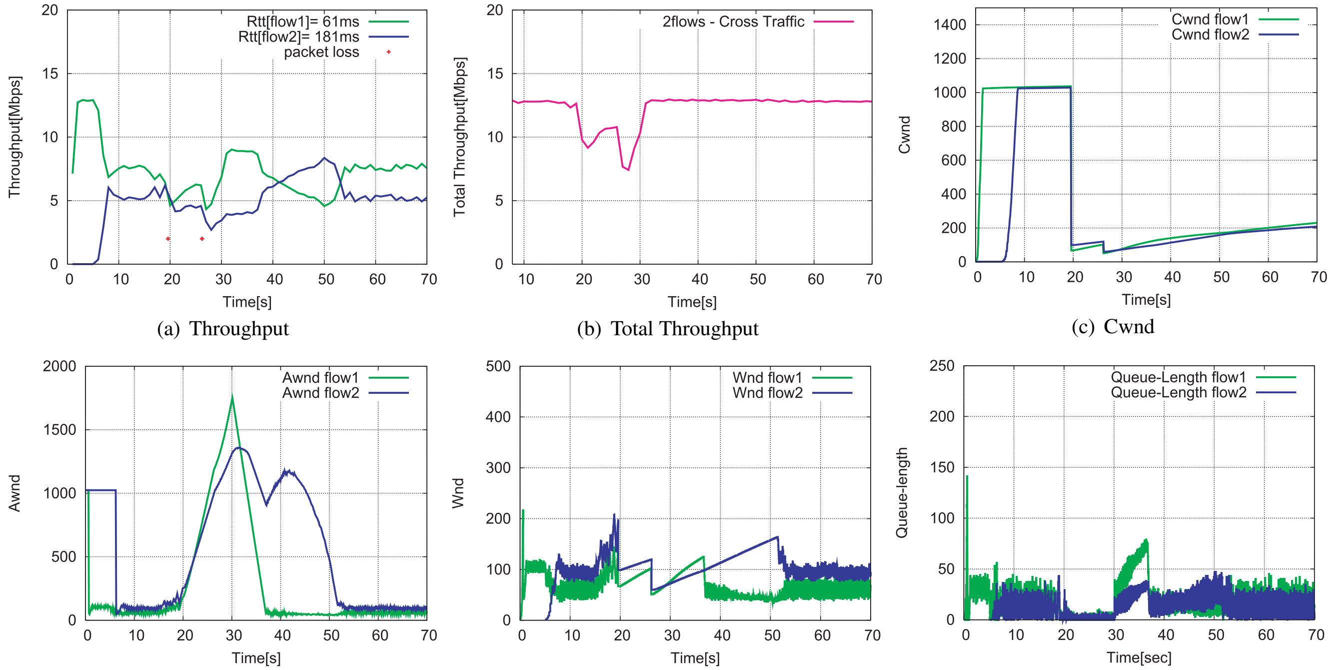
(d) Awnd (e) Wnd (f) Queue Length

Fig. 8 Previous scheme: 2TCP flows (w/o cross traffic).

by the the receiver is larger than the desired one calculated by the PLC_modem1. This approach can safely provide a way to determine the appropriate window size *wnd* of the sender according to the transmission condition on the PLC.

5.1 Problem in Coexisting Multiple TCP Flows

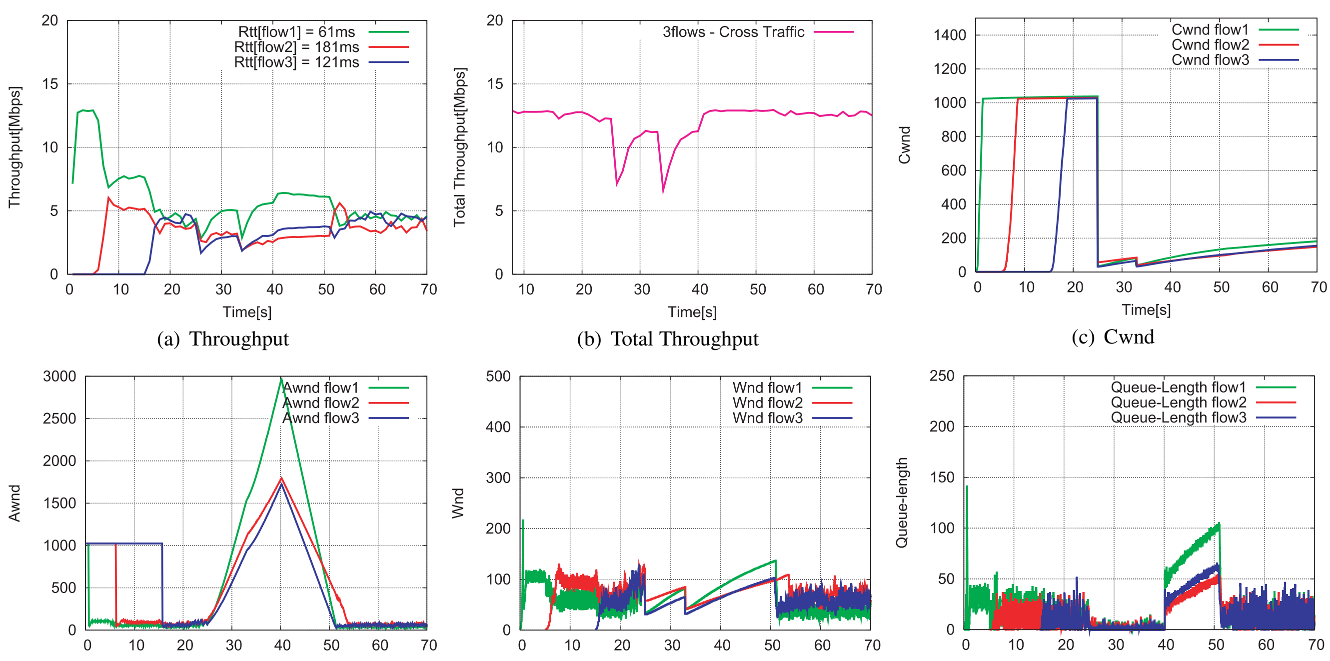
For a comparison purpose, we evaluate both TCP NewReno/SACK and the reference scheme [8] to see how competing flows consume the network bandwidth in net-



(a) Throughput (b) Total Throughput (c) Cwnd

(d) Awnd (e) Wnd (f) Queue Length

Fig. 9 Previous scheme: 2TCP flows (w/ cross traffic).



(a) Throughput (b) Total Throughput (c) Cwnd

(d) Awnd (e) Wnd (f) Queue Length

Fig. 10 Previous scheme: 3TCP flows (w/ cross traffic).

work topology I (Fig. 4). We focus on the scenario where two flows coexist, under an unstable environment (Inverter 45 dB) connected to the Internet in order to simplify the performance problems of the existing scheme. In addition, we assume that Sender 1 and 2 are connected to the Internet and there is a large RTT difference between them with the existence and non-existence of cross traffic.

The reference scheme [8] is made up of four phases. First, the RTT for each flow is measured during the connection establishment. Second, the previously obtained estimated RTT is utilized to calculate the maximum number of transmitted SDUs within the RTT period for each

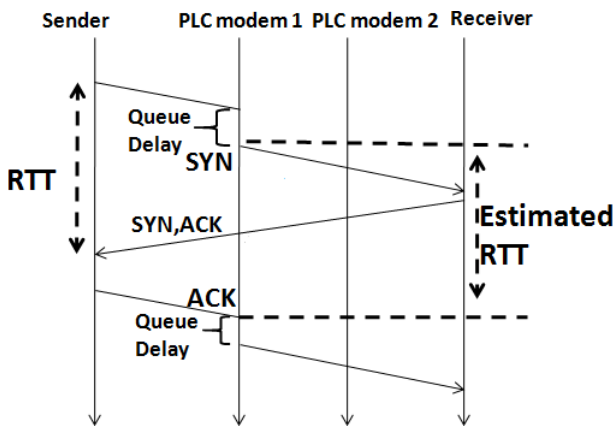


Fig. 11 Estimation of RTT.

flow (*Sample_Awnd[i]*). Then, as shown in Eq. (1), which is referred in [8], the appropriate amount of data transmission (*awnd*) is determined based on the estimated RTT and the number of coexisting TCP flows to maintain an adequate transmit queue length of PLC_modem1. Finally, a dynamic adaptation scheme of *awnd* is implemented in order to adjust the amount of information that must be sent by the TCP sender according to the network conditions. Therefore, besides monitoring the transmit queue length of PLC_modem1, it has the function of updating and returning the appropriate *awnd* value of each flow to the TCP sender. The pseudocode for the existing dynamic adaptation scheme of *awnd* can be found in Fig. 12.

Another important function implemented was the estimation of the RTT between the sender and the receiver based on a passive measurement technique [13]. The idea behind this technique is to estimate the RTT by using the SYN and ACK packets in a measurement point (e.g. PLC modem 1). More specifically, that happens during the connection establishment when the PLC modem 1 records both the TCP SYN packet departure time and the TCP ACK packet arrival time of each flow when they pass through this measurement point. Then the RTT estimation is calculated based on the difference between the TCP ACK packet arrival and TCP SYN packet departure time. The sketch of this estimation RTT technique is shown in Fig. 11.

However, this technique estimates the RTT period of a flow only during the three-way handshake (i.e., only during the connection establishment). After that, RTT can dynamically vary because of factors like network congestion and route change during the communication. Thus, a scheme that can also deal with the appearance of those negative elements during the communication period is necessary.

$$Awnd[i] = Sample\_Awnd[i] \times \frac{\frac{1}{RTT[i]}}{\sum_{i=1}^{N} \frac{1}{RTT[i]}} \qquad (1)$$

if $Queued\_pkt[i] \geq \frac{L}{N}$ **then**
  $Awnd[i] = Awnd[i] - \frac{1}{\frac{L}{N}}$
else
  $Awnd[i] = Awnd[i] + \frac{1}{\frac{L}{N}}$
end if

Fig. 12 Previous algorithm for dynamic adaptation of *awnd*.

where L: The maximum number of SDUs in a PDU in the current condition, N: The number of TCP flows coexisting, and *Queued_pkt[i]*: The number of SDU packets of the i-th flow remaining in Transmit queue of PLC modem 1.

The metrics we use to evaluate the simulation results are the throughput, congestion window (*cwnd*), advertised window (*awnd*), window size (*wnd*) and the queue length of PLC_modem1.

5.1.1 Problems in RTT Difference

We assume that flow2 begins the communication 5s after flow1 does. Taking into consideration the performance evaluation for the traditional TCP NewReno/SACK, Fig. 5(c) shows that even though flow2 changes to the congestion avoidance right after the communication starts, it maintains a high *wnd* value and increases gradually. Because of this, flow1 obtains low throughput for a considerable amount of time. However, to clarify the cause of the throughput unfairness (Fig. 5(a)) between flow1 and flow2, we also investigated the queue length for each flow at the PLC_modem1 (Fig. 5(d)). By comparing Figs. 5(c) and 5(d), we observed that the queue length is constantly high. More specifically, as flow1 occupies more than 70% of the queue length at the moment the flow2 starts the communication, the frames belonging to flow2 tend to be dropped because of buffer overflow. That is the reason why flow2 enters into the congestion avoidance phase very soon.

However, flow2 is superior to flow1 in terms of throughput until the second buffer overflow occurs as shown in Fig. 5(d). Figure 5(c) shows that when the second buffer overflow occurs, the *wnd* of flow 1 increases gradually and faster than flow2. As a result, the queue length (Fig. 5(d)) for flow1 becomes higher, thereby leading to a higher throughput during the rest of the communication period.

The buffer overflow at PLC sender modem only occurs when we evaluate the traditional TCP New Reno/SACK. Note that in TCP communication, TCP sender determines the amount of data to be transmitted to the TCP receiver based on its *w*nd. That is, as the traditional TCP employs a congestion control mechanism with the objective to avoid network congestion and overflow at the receiver side, it is common the occurrence of buffer overflow along the network path between TCP sender and receiver. Consequently, since TCP is not aware of the PLC network condition, the amount of information transmitted by the TCP sender based on *w*nd can unknowningly exceed the maximum PLC network capacity.

Regarding the previous scheme, Fig. 8(a) shows the throughput for the coexistence of *2 flows with large RTT and without cross traffic*. The result shown in Fig. 8(a) highlights the flow with small round trip time consumes large of the network bandwidth. That is, the flow with smaller RTT (e.g. flow1) reaches high throughput during all the communication time even after the second flow starts the communication in 5 s, consequently causing a not friendly allocation of the available bandwidth.

Furthermore, these findings show that the existing dynamic adaptation scheme is not effective when there is a considerable RTT difference between the flows. That is, this method does not update the *awnd* value considering the RTT period. The key problem in such circumstances is that once the *awnd* is not accurately modified for each TCP-ACK packet in PLC sender modem, the sender will not determine an appropriate *wnd* (Fig. 8(e)) as long as *awnd* (Fig. 8(d)) is smaller than *cwnd* (Fig. 8(c)).

5.1.2 Problems with the Existence of Heavy Cross-Traffic

We also evaluated the traditional TCP NewReno/SACK and the existing scheme in the worst-case scenario, where besides the RTT difference, CBR cross traffic with rate of 90 Mbps is also injected along the communication path between sender4 and receiver4. In this scenario, the coexistence of 2 and 3 flows are analyzed. When 2 flows coexist, the cross traffic comes into the communication path in 15 s and leaves in 30 s. However, in case of 3 flows we assume that the cross-traffic is injected in 20 s and leaves in 40 s. Considering TCP NewReno/SACK, we observed that during the cross-traffic, the *Queued_pkt* (transmit-queue in PLC_modem1) becomes very low (Figs. 6(d) and 7(d)). Nevertheless, when the cross traffic leaves the queue length gets extremely high. That is due to the gradual *wnd* increase of each flow (Figs. 6(c) and 7(c)) without any particular control. Therefore, the throughput unfairness is shown in Figs. 6(a) and 7(a).

Turning to the existing scheme, We observed that during the cross-traffic, the *Queued_pkt* (transmit-queue in PLC_modem1) becomes very low as well (Figs. 9(f) and 10(f)) and because of the condition imposed by this scheme ($Queued\_pkt[i] < \frac{L}{N}$), *awnd* keeps on increasing unnecessarily as shown in Figs. 9(d) and 10(d). During this time interval, the existing scheme does not work properly. However, when the cross traffic leaves, the amount of information passing through PLC network increases and the transmit queue at PLC_modem1 starts being occupied again.

Hence, Fig. 9(d) shows that the *awnd* of the first flow decreases immediately after the cross traffic leaves, because the increasing speed of *wnd*(Fig. 9(e)) of each one of the flows, which is determined by the *cwnd* in such situation, is different. Consequently, the queue length is not occupied by each flow with the same frequency. Thereby, leading to the *awnd* decreasing delay of flow2. Fig. 10(d) highlights that for the coexistence of 3 TCP flows, the same procedure will occur when cross traffic leaves the communication path. Furthermore, as expected, Figs. 9(e) and 10(e) demonstrate that *wnd* receives the minimum value between *cwnd* (Fig. 10(c)) and *awnd* (Fig. 10(d)) (i.e., *Wnd = min (cwnd,awnd)*).

In short, as Figs. 9(a) and 10(a) suggest, the reference scheme suffers from the throughput instability problem for coexisting multiple flows in such environments.

5.2 Problem in Coexisting Multiple TCP and VoIP Flows with no QoS

In this section, considering that some PLC solutions may still not provide QoS (Quality of Service) management support, we focus on the network topology II (Fig. 13), where 3 TCP flows coexist with 2 bi-directional VoIP flows without QoS. We assume that Sender 1, 2 and 3 are connected to the Internet and Receiver 1, 2 and 3 are limited to the home network. In addition, we also assume the existence of heavy cross-traffic (90 Mbps) on the Internet. Both TCP flows and VoIP flow 1 pass through PLC modem 1 in the same direction. As a result, VoIP flow 1 delay goes over 150 ms for some period and shows to be inconstant (Figs. 14(a) and 15(a)). Regarding both traditional TCP NewReno and the existing scheme, they could not keep the queue length short Figs. 14(b) and 15(b). In case of the existing scheme, since it could not employ a properly controlled *awnd* for each TCP flow, the *awnd* unnecessarily increases during the cross traffic.

Therefore, VoIP flow 1 such circumstances does not satisfy the QoS requirements for VoIP applications according to the ITU-T recommendation Y.1541 [16].

6. Fairness Scheme For the Bandwidth Allocation

The dynamic adaptation of *awnd* in response to the change in the network condition implemented in the previous scheme [8] makes PLC_modem1 increase and decrease *awnd* gradually based on the change in the Transmit queue length. In that case, the scheme obtained excellent throughput performance because in the studied environment was not assumed adverse situations like large RTT difference between the flows and cross-traffic. That is, the estimated RTT did not differ a lot from one flow to the other. Consequently, the dynamic adaptation occured smoothly.

However, when we bring this work to an Internet environment where there are large RTT variations between the flows caused by adverse factors such as network congestion and the existence of cross traffic. The existing scheme faces throughput unfairness problems as shown in Figs. 8(a) and 9(a). Thus, we propose a new fairness scheme for the appropriate bandwidth allocation in this heterogeneous network. The RTT estimation is also made during the connection-establishment time (Fig. 11) and the appropriate calculation of *awnd* is the same of the previous scheme [8] shown in Eq. (1).

In literature [14], a TCP algorithm called TCP Hybla was proposed to solve the RTT-unfairness problem in het-

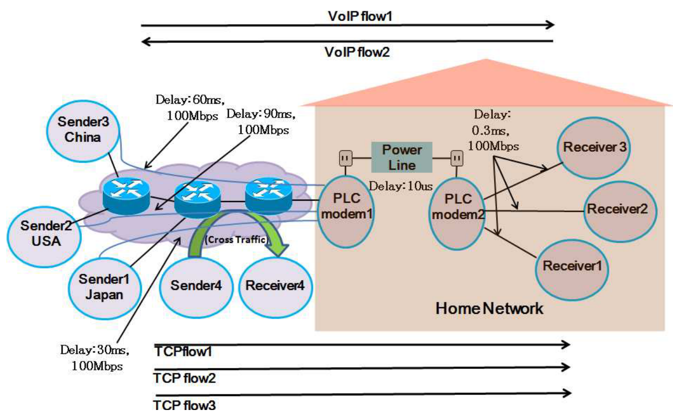
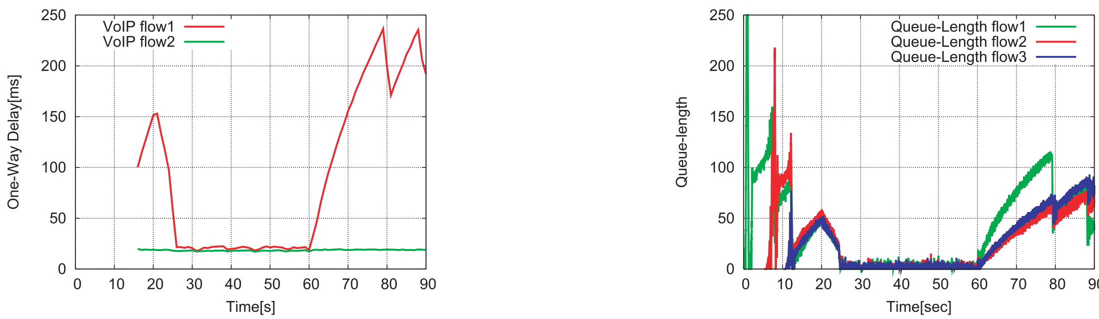
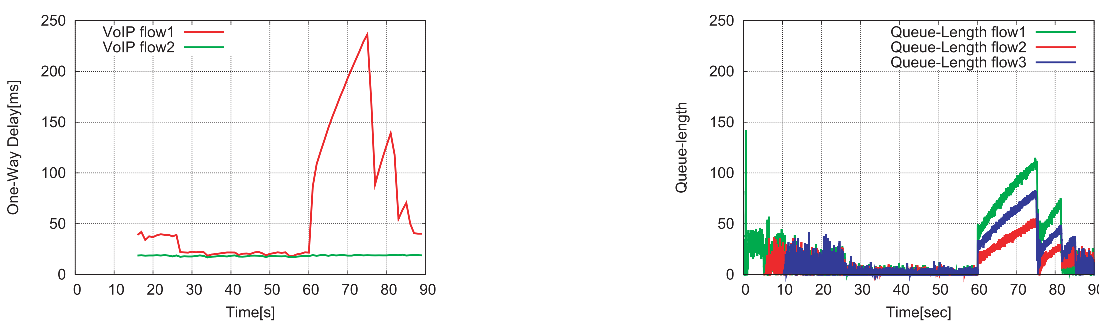


Fig. 13 Network topology II.



(a) End-to-End Delay (b) Queue Length

Fig. 14 TCP NewReno/SACK: VoIP/3TCP flows (w/ cross traffic).



(a) End-to-End Delay (b) Queue Length

Fig. 15 Previous scheme: VoIP/3TCP flows (w/ cross traffic).

erogeneous networks with both long and short RTT connections (e.g. satellite and wired, respectively). This algorithm modifies the New Reno's Slow Start and Congestion Avoidance to make them semi-independent of RTT. That is, It was proposed considering the fact that in TCP New-Reno the congestion window size is inversely dependent on RTT. Consequently, a flow with a shorter RTT has advantage compared to one with longer RTT.

Our new scheme has been proposed based on this algorithm mainly because they gave special importance to the degradation problem of TCP throughput with New Reno standard congestion control. On the other hand, our proposed TCP rate control scheme does not change the TCP congestion control. It only makes some necessary modifications at the PLC_modem1 based on the transmission condition on the PLC. The pseudocode for the proposed adaptation scheme of *awnd* can be found in Fig. 20

A similar scaling factor ρ that was proposed by TCP Hybla algorithm is also applied in our scheme. However, the mechanisms under investigation in our study are different. In order to determine the ρ value, first we make use of the RTT estimation done by PLC_modem1 for each flow during the connection establishment time. Then, to obtain a normalized performance to the flow with the lowest RTT, we calculate $\rho[i]$ by using the equation $\rho[i] = \frac{RTT[i]}{RTT_{ref}}$, where RTT[i] is the RTT of each flow and RTT_{ref} is the smallest

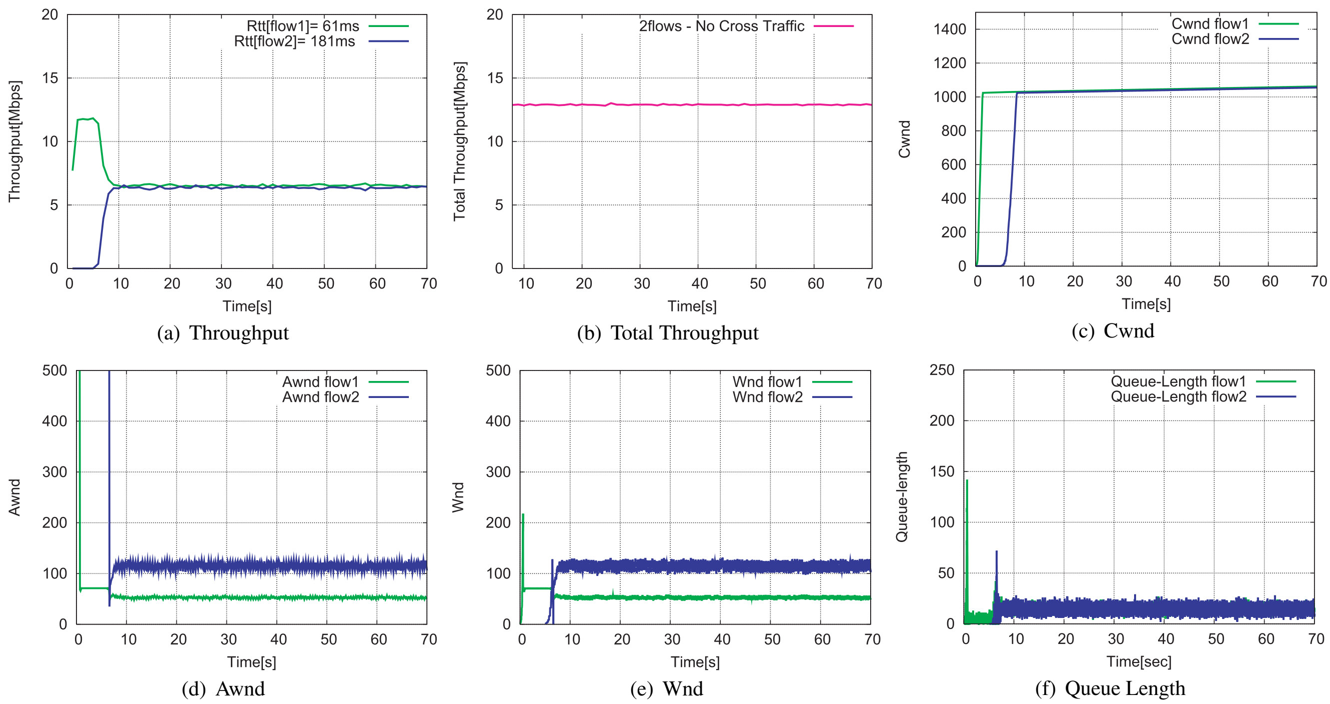


Fig. 16 Proposed scheme: 2TCP flows (w/o cross traffic).

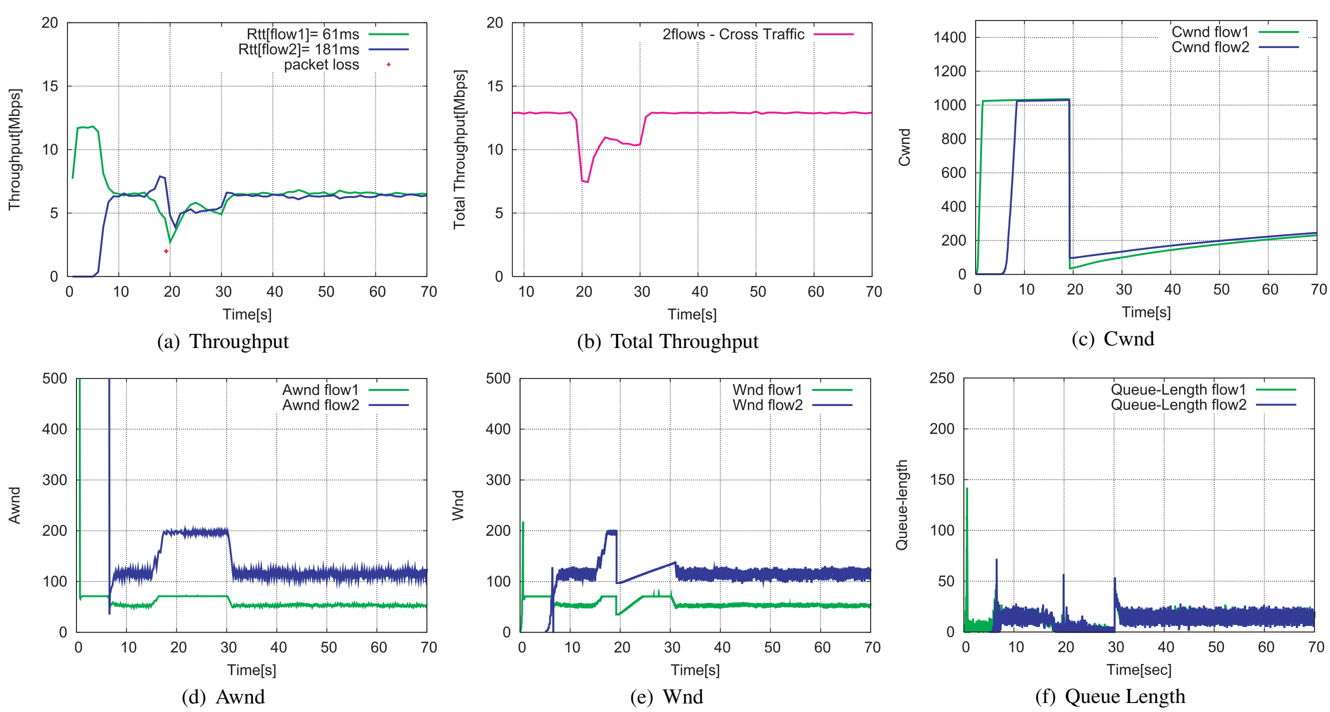
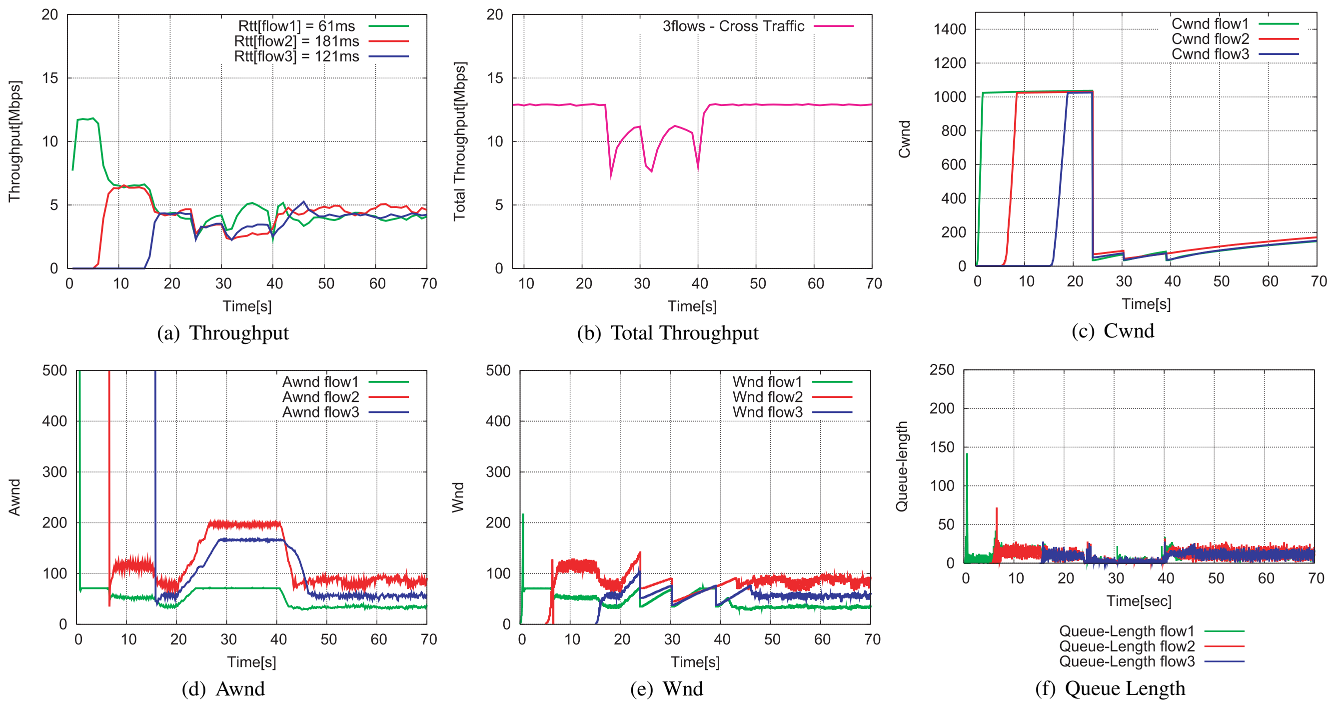


Fig. 17 Proposed scheme: 2TCP flows (w/ cross traffic).

RTT among the coexisting flows, which is called the reference RTT.

Let me explain how our scheme, which is described in Fig. 20, works in two different cases where the "large RTT difference" and the "existence of cross-traffic" are addressed. First of all, we use the Eq. (1) referred in [8] to calculate the initial *awnd* of each flow.

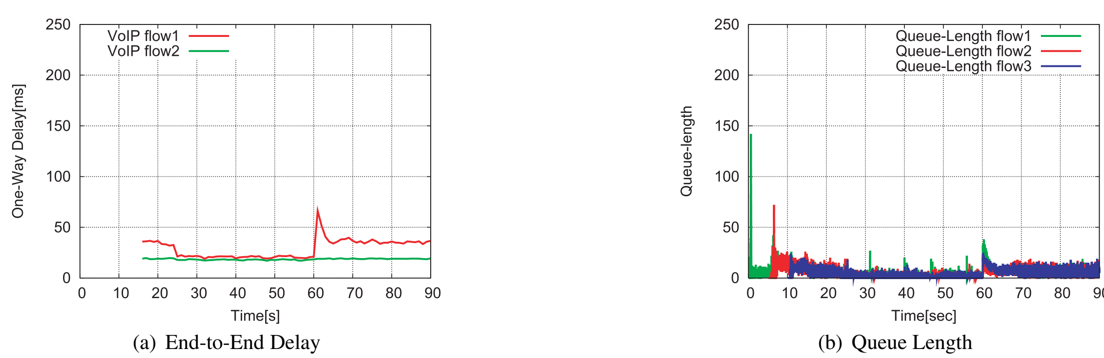
In the first case, let's assume there is a large RTT difference between two coexisting flows and there are no adverse factors that make RTT fluctuates, such as cross traffic. In such circumstances, lines 1 to 7 of the proposed algorithm (Fig. 20) is used. We compare the queue length condition



(a) Throughput (b) Total Throughput (c) Cwnd

(d) Awnd (e) Wnd (f) Queue Length

Fig. 18 Proposed scheme: 3TCP flows (w/ cross traffic).



(a) End-to-End Delay (b) Queue Length

Fig. 19 Proposed scheme: VoIP/3TCP flows (w/ cross traffic).

if $Awnd[i] \leq Sample\_Awnd[i]$ **then**

  if $Queued\_pkt[i] \leq \frac{L}{N}$ **then**

    $Awnd[i] = Awnd[i] + \frac{2^{\rho[i]}}{Awnd[i]}$

  else

    $Awnd[i] = Awnd[i] - \frac{2^{\rho[i]}}{Awnd[i]}$

  end if

else

  $Awnd[i] = Awnd[i] - 2^{\rho[i]-1}$

end if

Fig. 20 Proposed Algorithm for Dynamic Adaptation of *awnd*.

(i.e. $Queued\_pkt[i]$, which is the number of SDU packets of the i-th flow remaining in the transmit queue) to the maximum number of SDUs in a PDU [L] determined by HD-PLC standard divided by the number of TCP flows [N]. Then, although the estimated RTT is not updated according to the network conditions, our proposed scheme compensates for this drawback by using the following equation $\frac{2^{\rho[i]}}{Awnd[i]}$. Note that as previously explained, the scaling factor ρ is decided after the RTT estimation of each flow and not updated afterwards. That is, by using scaling factor ρ and *awnd* we can make the increasing and decreasing rate change not only proportionally with the RTT of each flow, but also gradually because *awnd* avoids scaling factor ρ from receiving a large value. Because of that, the available network resource at PLC network (i.e. Queued buffer at PLC modem 1) will be used efficiently. In this way, *awnd* is updated. That is, in case $Queued\_pkt[i] \leq \frac{L}{N}$, *awnd* is increased by $Awnd[i] = Awnd[i] + \frac{2^{\rho[i]}}{Awnd[i]}$, otherwise it is decreased by $Awnd[i] = Awnd[i] - \frac{2^{\rho[i]}}{Awnd[i]}$. As a result, the awnd is dynamically updated based on the queue length condition at PLC modem 1.

In the second case, let's assume both large RTT difference and the existence of cross traffic. In such circumstances, the RTT is expected to fluctuate considerably. The changes in the RTTs will influence on the amount of information that is sent through the network path. More specifically, when cross-traffic is addressed in the Internet environment the number of SDUs that pass through the PLC network decreases substantially. For that reason, the scheme proposed by authors in [8] increases *awnd* unnecessarily (Fig. 10(d)). In addition, since *awnd* determines the *wnd* as long as it is lower than *cwnd*, the entire approach becomes invalidated during the time it increases unnecessarily, thereby exceeding $Sample\_Awnd$. Thus, their approach is not well suited for such circumstances. To solve this problem, in our proposed scheme, we implemented an upper bound limit control. As indicated in lines 8 to 10 of the proposed algorithm (Fig. 20), if *awnd* is higher than $Sample\_Awnd$ we hold it back by using the equation: $Awnd[i] := Awnd[i] - 2^{\rho[i]-1}$. $2^{\rho[i]-1}$ shows to be an efficient method to decrease *awnd* based on the RTT of each flow because it determines the rate at which *awnd* should be decreased. That is, $\rho - 1$ is used in order to not make *awnd* decrease suddenly and consequently affect the throughput performance of the flows in the network. In details, for each flow there will be a different decreasing rate based on different RTTs of individual TCP flows. In addition, it is worth noting that the decreasing rate mentioned here is only used in case $Sample\_Awnd$ is exceeded.

As a result, we have got to keep the balance in terms of network resource allocation between the competing flows even in a critical network condition.

7. Performance Evaluation

We dedicate this section to examine the performance of the proposed scheme in the same environment (Fig. 4), where up to three TCP flows coexist on the same unstable PLC network connected to the Internet under adverse situations. In our previous work [15], we only evaluated the existence of two coexisting TCP flows. However, in this study, we present a deeper evaluation of our scheme to clarify its effectiveness. Thus, we consider not only two but the coexisting of three TCP flows and VoIP as well.

In case of 2 TCP flows, we assume that TCP flow2 starts the communication 5 s after the TCP flow1 starts. Figs. 16(a) and 17(a) show that when TCP flow2 starts the communication, there is no loss in terms of throughput compared to TCP flow1. That is, PLC_modem1 modifies *awnd* (Figs. 16(d) and 17(d)) according to the transmission conditions of PLC and passes the modified *awnd* information to the TCP sender through TCP ACK in order to make control of the window size (*wnd*) of TCP. Furthermore, In such cirmcunstances, it must be emphasized that both existing and proposed schemes keep the queue length short (Figs. 8(f) and 16(f)).

When cross-traffic is injected along the communication path in 15 s and leaves in 30 s, the upper bound limit employed by our approach works well by limiting the maximum *awnd* value necessary for each flow without degrading their throughput performance as shown in Fig. 17(a). Consequently, Fig. 17(f) when compared to Fig. 9(f) shows that our scheme keeps the transmit queue length stable even in a critical situation, when both large RTT differences and heavy cross-traffic are considered simultaneously. Moreover, different from the *wnd* in Fig. 16(e), which consists of the lower value between *cwnd* and *awnd* (Figs. 16(c) and16(d)), in the presence of cross-traffic *wnd* (Fig. 17(e)) is reduced to half of the minimum value between *cwnd* and *awnd* (19.2s). That is due to the packet loss with duplicate acks. We can see that such packet loss occurs with less frequency in Fig. 17(a) than in Figs. 6(a) and 9(a).

In this worst-case scenario, when we consider 3 TCP flows, our findings for *awnd* and *queue-length* in Figs. 18(d) and 18(f) emphasizes the validity of our scheme.

After all the performance evaluation made so far, it is also fundamental to note that the objective of obtaining the throughput fairness between the coexisting flows in the above-mentioned situations, did not cause side effects in terms of total throughput. Figs. 5(b), 6(b), 7(b) and Figs. 8(b), 9(b), 10(b) show the total throughput results for TCP NewReno/SACK and the existing scheme, respectively. Our findings confirm that when compared to our proposed scheme, there was no degradation in the total throughput as can be seen in Figs. 16(b), 17(b) and 18(b). In sum, these results have led us to conclude our proposed scheme obtained the throughput fairness without degrading the maximum communication performance.

In addition, the VoIP end-to-end delay problem addressed when 3 TCP/VoIP flows coexist in Sect. 5.2 was also solved as shown in Fig. 19(a). In Fig. 19(b), the new scheme keeps the transmit queue length very low and stable, even with the existence of cross traffic (20 s–60 s). Hence, we could maintain the VoIP end-to-end delay very low during all the communication period.

Finally, we can affirm based on these results that our proposed dynamic adaptation of *awnd* determines the appropriate *wnd* (Figs. 16(e), 17(e) and 18(e)) according to the network conditions. That is, in case its value is lower than the *cwnd* one (Fig. 18(c)). As a result, the throughput fairness among multiple coexisting TCP flows is achieved efficiently (Figs. 16(a), 17(a) and 18(a)). Furthermore, Our proposed scheme guarantees a satisfactory VoIP one-delay latency even in an unstable Internet environment.

8. Conclusion

In the present paper, we have proposed a new scheme to allocate the available bandwidth fairly when multiple TCP flows coexist over heterogeneous networks (i.e. IP over PLC Network connected to the Internet) in a realistic environment.

The contribution of this work is three-fold. First, it treats the large RTT difference that can occur between coexisting flows. Second, the proposed scheme copes with the existence of heavy cross traffic that significantly influences the RTT values, and thus yield a drastic performance imbalance. Finally, we also show that even though QoS is not taken into consideration in our study, when TCP/VoIP coexist in an unstable environment, our scheme mitigates the degradation in network performance and fulfills the necessary one-way delay QoS requirement for VoIP flows based on ITU-T Recommendation Y.1541 [16]. Therefore, through the simulation results we could show that our proposed rate control scheme can efficiently cope with the coexisting of multiple and different types of flows passing through the Internet.

In this paper, we addressed TCP communication and the coexistence of TCP and VoIP without QoS. However, in the Internet there are many other types of applications, such as video-on-demand (VoD) and real-time streaming. Further studies are needed to analyze whether or not our proposed approach fulfills the requirements of those applications in case QoS management was not provided.

References

[1] Y.-J. Lin, H.A. Latchman, S. Katar and M.K. Lee, "A comparative performance study of wireless and power line networks," IEEE Commun. Mag., vol.41, no.4, pp.54–63, April 2003.

[2] IEEE Standard for Broadband Over Power Line Networks: Medium Access Control and Physical Layer Specifications, IEEE Std. 1901-2010, Sept. 2010.

[3] HD-PLC High Definition Power Line Communication. Available at http://www.hd-plc.org

[4] S. Galli, M. Koch, H.A. Latchman, S. Lee, and V. Oksman, "Chap.7: Industrial and international standards on PLC-based networking technologies," Power Line Communications, 1st ed., H. Ferreira, L. Lampe, J. Newbury, and T. Swart, Eds., ch. 7, John Wiley & Sons, New York, NY, 2010.

[5] J. Liu, B. Zhao, L. Geng, and Z. Yuan, "Current situations and future developments of PLC technology in China," IEEE International Symposium on Power Line Communications and its Applications (IEEE ISPLC2012), Beijing, China, March 2012.

[6] I. Berganza Valmala, G. Bumiller, H.A. Latchman, M.V. Ribeiro, A. Sendin Escalona, E.R. Wade and L.W. Yonge, "Chap.8: Systems and implementations," Power Line Communications, 1st ed., H. Ferreira, L. Lampe, J. Newbury, and T. Swart, Eds., ch. 8, John Wiley & Sons, New York, NY, 2010.

[7] S. Galli, A. Scaglione, and Z. Wang, "For the grid and through the grid: The role of power line communications in the smart grid," Proc. IEEE, vol.99, no.6, pp.–, June 2011.

[8] M. Mizutani, Y. Miyoshi, K. Tsukamoto, M. Tsuru, and Y. Oie, "Network-supported TCP rate control for high-speed power line communications environment," Elsevier Simulation Modeling Practice and Theory (SIMPAT), vol.19, no.1, pp.69–83, Jan. 2010.

[9] H. Toral-Cruz, J. Argaez-Xool, L. Estrada-Vargas and D. Torres-Roman, "An introduction to VoIP: End-to-end elements and QoS parameters," VoIP Technologies, Dr. Shigeru Kashihara (Ed.), ISBN: 978-953-307-549-5, InTech, Available from: http://www.intechopen.com/books/voip-technologies/an-introduction-to-voip-end-to-end-elements-and-qos-parameters

[10] B. Zarikoff and D. Malone, "Construction of a PLC test bed for network and transport layer experiments," IEEE International Symposium on Power Line Communications and its Applications (IEEE ISPLC2011), Udine, Italy, April 2011.

[11] M. Mizutani, Y. Miyoshi, K. Tsukamoto, M. Tsuru, and Y. Oie, "Network-supported TCP rate control for high-speed power line communications," The Second International Workshop on Information Design (WIND'09), Open University of Catalonia, Barcelona, Spain, Nov. 2009.

[12] HD-PLC White Paper Ver.1.06, 27 Oct 2009. Available at http://www.hd-plc.org/

[13] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towesly, "Inferring TCP connection characteristics through passive measurements," Proceedings of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE INFOCOM, Hong Kong, March 2004.

[14] C. Caini and R. Firrincieli, "TCP Hybla: A TCP enhancement for heterogeneous networks," International J. Satellite Communications and Networking, vol.22, pp.547–566, 2004.

[15] A. Muniz, K. Tsukamoto, M. Tsuru, and Y. Oie, "Network-supported TCP rate control for the coexistence of multiple flows in IP over PLC," IEEE International Symposium on Power Line Communications and its Applications (IEEE ISPLC2012), Beijing, China, March 2012.

[16] Recommendation Y.1541, Network Performance Objectives for IP-Based Services, Feb. 2006.

Adriano Muniz received his B.E. degree in Computer Science from UESPI University, Brazil, Teresina-PI, in 2003. He was granted a Japanese Government (MEXT) Scholarship and studied at Kyushu University, Japan, as a research student from 2005 to 2007. Then, he received his M.E. degree in Computer Science from Kyushu Institute of Technology (KIT), Japan, in 2009. He was also granted a Rotary Yoneyama Scholarship to further his studies as a Ph.D. student. He is currently working towards his Ph.D. degree in Computer Science at KIT. His research interests include computer networks and wireless communications.

Kazuya Tsukamoto received his D.E. degree in Computer Science from Kyushu Institute of Technology (KIT), Japan, in 2006. From April 2006 to March 2007, he was a Japan Society for the Promotion of Science (JSPS) Research Fellow at KIT. Beginning in July 2006 and ending in January 2007, he spent six months at the Department of Information and Computer Science, University of California, Irvine, on leave of absence from KIT. As of April 2007, he is an assistant professor with the Department of Computer Science and Electronics, KIT. His research interests include performance evaluation of computer networks and wireless networks. He is a member of the IPSJ, ACM, and the IEEE.

Masato Tsuru received B.E. and M.E. degrees from Kyoto University, Japan in 1983 and 1985, respectively, and then received his D.E. degree from Kyushu Institute of Technology, Japan in 2002. He worked at Oki Electric Industry Co., Ltd., Information Science Center, Nagasaki University, and Japan Telecom Information Service Co., Ltd. In 2003, he moved to the Department of Computer Science and Electronics, Kyushu Institute of Technology as an Associate Professor, and then has been a Professor in the same department since April 2006. His research interests include performance measurement, modeling, and management of computer communication networks. He is a member of the ACM, IEEE, IPSJ, and JSSST.

Yuji Oie received the B.E., M.E., and D.E. degrees from Kyoto University in 1978, 1980, and 1987, respectively. From 1980 to 1983 he worked at Nippon Denso Company Ltd., Kariya, Japan. From 1983 to 1990 he was with the Department of Electrical Engineering, Sasebo College of Technology. From 1990 to 1995 he was an associate professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, KIT. From 1995 to 1997 he was a professor at the Information Technology Center, NAIST. Since April 1997 he has been a professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, KIT. His research interests include performance evaluation of computer communication networks, high-speed networks, and queuing systems. He is a fellow of the IPSJ, and a member of the IEEE and the ACM.