Vol. 45 No. 12 IPSJ Journal Dec. 2004

Regular Paper

Fair Bandwidth Allocation in Diffserv Networks

Tomoya Hatano,† Hiroshi Shigeno,† Ken-ichi Okada†
and Yutaka Matsushita††

The Assured Forwarding (AF) service of the IETF Diffserv architecture provides a qualitative service differentiation between classes of traffic. However, different bandwidth is allocated between flows within the aggregate in AF service, when TCP flows with different Round Trip Times (RTTs) are aggregated into one AF class. This unfair situation for each TCP flow should be solved. Main reason of unfair bandwidth allocation is TCP congestion control. In this paper, we propose Fair Rate-conscious TCP (FR-TCP) to improve fairness of the bandwidth within the aggregate. FR-TCP (1) defines fairness and Fair Rate (FR) as minimum bandwidth of each aggregated flow, notifies FR to sender TCP and (3) controls TCP congestion window based on FR. We evaluate proposed FR-TCP through computer simulations with Network Simulator version 2 (ns-2). The results show that FR-TCP controls congestion window to get bandwidth according to Fair Rate and allocates fair bandwidth efficiently within the aggregate. FR-TCP controls sending rate promptly where the number of flows is dynamically changing. Even though FR-TCP and the current TCP are heterogeneous, FR-TCP showed the effectiveness.

1. Introduction

In the broadband era of the Internet, a growing number of applications require a form of end-to-end quality of service (QoS) such as delay, jitter, throughput, and error rate. The best-effort service model supported by the Internet does not meet such requirements.

To overcome the inherent limitation of the best-effort Internet, the differentiated services (Diffserv) [1),2)] are being embodied. An attractive feature of the Diffserv is that it does not require admission control or per-flow classification, and is therefore scalable on both the control and data paths.

Each user is associated with a Service Level Agreement (SLA), which is a contract between a customer and Internet Service Provider. Based on SLA, Diffserv network marks Diffserv Code Point (DSCP) on packets at the edge of the network and provides a qualitative service differentiation between classes of traffic according to DSCP. The packets in the same class are marked the same DSCP and classified into a class.

At network boundaries, these marked packets are treated according to Per Hop Behaviors (PHB) specified for each service class. Two PHB groups have been defined, namely Expedited Forwarding (EF) PHB group [3)] and Assured Forwarding (AF) PHB group [4)].

EF service is a PHB to provide virtual leased line to the flow. Since each router limits incoming rate of aggregate flow to guaranteed bandwidth in EF service, packets of aggregate flow rarely wait at the buffer of router in the network. EF service provides low loss, low latency, low jitter and guaranteed bandwidth service. Thus EF service is used when users want to make virtual private network.

On the other hand, AF service is qualitative PHB to realize Better Than Best Effort service. AF service guarantees average transmitting rate in the network. When there is unused bandwidth by other flows, every flows can increase their transmitting rate more than SLA. Consider contract rate is R Mbps. Traffic Congestion Control System (TCS) gives IN profile to packets of aggregate flow, when average transmitting rate of aggregate flow is less than R Mbps. And the packets exceeding R Mbps are marked with OUT of profile. When link gets congested, OUT packets will be discarded first. This system can be accepted and guarantee R Mbps to aggregate flow at each domain, even the flow passes through a number of Diffserv domains.

In AF service, TCP traffic is often used to provide assured bandwidth. However, AF service guarantees bandwidth to only aggregates of TCP flows. AF service leads to different bandwidth allocation between flows within the aggregate, when TCP flows with different Round

† Faculty of Science and Technology, Keio University
†† Tokyo University of Technology

Trip Times (RTTs) are aggregated into one AF class. RTT is the time it takes for a packet to travel from sender to receiver and then back to the sender, and it is the sum of propagation delay and queuing delay.

It is known that this problem arises when TCP flows with different RTTs send packets in the Internet even if they do not go through Diffserv networks [5]. Since, AF service in Diffserv networks is priority forwarding, it is needed to allocate fair bandwidth to each flow more than best-effort service and this problem in AF service should be solved promptly.

In this paper, we propose FR-TCP to improve the fairness of bandwidth in cooperation with network nodes and end hosts over Diffserv networks. We (1) define a fair bandwidth as FR, (2) introduce the FR notification mechanism to sender TCP and (3) propose congestion control mechanism with notified FR.

The remainder of this paper is organized as follows. Section 2 describes the problem of unfair bandwidth allocation. We propose new mechanisms in Section 3 and performance evaluations are presented in Section 4. Related work is expressed in Section 5, after which conclusions follow in Section 6.

2. Unfair Bandwidth Allocation

In general, when we use AF service, each organization like individual and enterprise makes one SLA. Many flows are aggregated into one SLA and each aggregated flow receives shared distribution that is guaranteed by networks. Since each flow within one aggregate receives identical service, each flow should share guaranteed bandwidth equally. However, when multiple flows with different RTTs are aggregated, unfair bandwidth allocation between flows is observed.

Unfair bandwidth allocation in Diffserv network is related closely to TCP congestion control mechanism. First, we describe TCP congestion control mechanism.

2.1 New Reno [6]

There are many techniques for TCP congestion control mechanisms and New Reno is the most common TCP implementation on the Internet. We focus on New Reno as congestion control mechanism in this paper. New Reno is a variant of TCP Reno [7] with a little modification within Fast Recovery algorithm.

New Reno controls sending rate according to network condition using congestion window (cwnd). In congestion avoidance, New Reno increases cwnd by 1 segment of cwnd every 1 RTT and sets ssthresh at the half value of cwnd for each received Acknowledgment (ACK). Ssthresh is used to determine whether the slow start or congestion avoidance algorithm is used to control data transmission [8]. When cwnd is higher than ssthresh, congestion avoidance is used. Otherwise, the slow start is. For each loss event, cwnd is set to the value of the ssthresh in order to avoid network congestion. TCP Reno controls the cwnd as follows:

$$cwnd = cwnd + \frac{1}{cwnd} \quad \text{(each received ACK)} \qquad (1)$$

$$ssthresh = \frac{1}{2} cwnd \qquad (2)$$

$$cwnd = ssthresh \text{ (each loss event)} \qquad (3)$$

2.2 Reasons of Unfair Bandwidth Allocation

In New Reno, flow with longer RTT increases cwnd and ssthresh more slowly than flow with shorter RTT. New Reno connection with slower increasing window cannot adapt sending rate to network condition promptly, while connection with faster increasing window takes more available bandwidth. It leads to unfair bandwidth allocation. Moreover flow with longer RTT sets ssthresh at the less value than that of shorter RTT, because cwnd of flow with longer RTT is smaller then that of flow with shorter RTT. Therefore, cwnd of flow with longer RTT is set at the smaller value then that of flow with shorter RTT even for loss event.

There is another explanation way for unfair bandwidth allocation. Based on received ACK and packet loss, TCP congestion control mechanism controls the cwnd to transmit packets effectively. As a result, packet loss event arises periodically. Thus, throughput (T) is related to packet loss rate (p) and RTT, and TCP New Reno response function is computed as follows [9]:

$$T = \frac{1.22}{RTT \times p^{0.5}} \qquad (4)$$

As you can see, flow's RTT obviously has a great influence on its throughput and lead to unfair bandwidth allocation. From Eq. (4), throughput is inverse proportional to RTT, which leads to unfair bandwidth allocation.

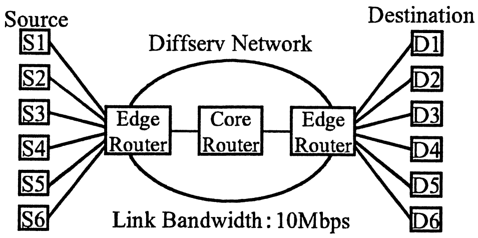

Fig. 1 Simulated network.

2.3 Situation Where Flows with Different RTTs are Aggregated

When each aggregated flow is transmitted under the same condition, bandwidth guaranteed by the SLA is allocated to each flow evenly. However, flows under various conditions are aggregated in usual situation. Such conditions result in unfair bandwidth allocation between flows. In this paper, we focus on the difference of RTT, which has a great impact on fairness of bandwidth in TCP flows[10].

In general, the simplest service called one-to-one service model guarantees bandwidth between one sender and one receiver. In this service, it is considered that condition of every flow is similar. However, as to AF service for many receivers, one-to-few service model and one-to-any service model are also proposed[11].

Taking cost of contracts and network utility into consideration, one-to-few and one-to-any service model are more effective. Under such circumstances, multiple flows with different RTTs are aggregated into one SLA. This is why the situation where flows with different RTTs are aggregated is more familiar in Diffserv networks.

2.4 Example of Unfair Bandwidth Allocation

In order to make unfair bandwidth allocation clearly understandable, we simulated one example of this problem.

We evaluate New Reno through computer simulations with Network Simulator version 2 (ns-2)[12]. The simulated network is shown in **Fig. 1**. Bandwidth of each link is 10 Mbps. The contracted bandwidth 9 Mbps is shared among all 6 TCP flows in the aggregate. It is reasonable for individual flow to achieve 1.5 Mbps as fair bandwidth. RTTs of flows from 1 to 6 are 40, 80, 120, 160, 200 and 240 ms, respectively.

Table 1 reflects each throughput through computer simulation when TCP flows with different RTTs are aggregated. As shown in Table 1, the achieved throughput of individual flow varies due to its RTT. Although the aggregate as a whole exceeds the contracted rate, we see flows with shorter RTTs show higher throughput, while flows with longer RTTs do not achieve fair bandwidth allocation. Therefore, bandwidth allocation between flows is unfair when RTTs of flows are different.

Table 1 Simulation result when flows with different RTTs are aggregated.

Flow No	RTT (ms)	Throughput (Mbps)
1	40	3.06735
2	80	2.08503
3	120	1.59616
4	160	1.27275
5	200	1.11065
6	240	0.85136
Total		9.98330

3. Proposed Mechanism

In this section, we propose Fair Rate-conscious TCP (FR-TCP) to improve fairness of the bandwidth within the aggregate. We (1) define fairness and Fair Rate (FR) as minimum bandwidth of each aggregated flow, (2) make edge routers notify FR to sender TCP and (3) propose TCP congestion control mechanism based on FR to realize average bandwidth to improve fairness of bandwidth.

3.1 Definition of Fairness

AF service leads to unfair bandwidth allocation between flows within the aggregate. To allocate fair bandwidth among all flows in any circumstance of RTT, we need to improve the existing mechanism. All flows should share contracted bandwidth equally.

We define fairness as the state where bandwidth of each flow sharing the bottleneck link between every flow is allocated equally. It is reasonable to allocate bandwidth equally, even when flows have different RTTs, because bandwidth of the all flows is allocated equally when each flow has the same situation. In addition, we define the appropriate forwarding rate that each flow within the aggregate should achieve, as FR. FR is calculated at edge routers as follows.

$$FR = \frac{\mathit{Contracted\ Rate}}{\mathit{Number\ of\ Flows}}. \tag{5}$$

Contracted Rate is allocated bandwidth for aggregate flows and contracted in SLA.

3.2 FR Notification

Edge routers compute FR with contracted

bandwidth and aggregated flow number. Since the number of aggregated flow is changing dynamically, FR-TCP catches up accurate number with the help of Policy Server [13].

Policy Server manages contracted information and resource in networks and provides feasible policy control according to dynamic network change. When users make new connection of guarantee service, the sender sends a request message to Policy Server. When Policy Server detects the change in the number of aggregated flows, it upgrades information in edge routers.

FR computed by edge routers needs to be notified to sender TCP in order to control sending rate based on FR. In FR-TCP, edge routers write FR on packets and the receiver TCP is informed of FR. The receiver TCP piggybacks FR on ACK, then sender TCP receives FR within one RTT.

3.3 Congestion Control of FR-TCP

3.3.1 Sending Rate Control

TCP increases cwnd for each received ACK and decreases it for each loss event. In FR-TCP we make change in cwnd decrease and not in cwnd increase.

FR-TCP increases 1 segment of cwnd every 1 RTT for each received ACK, which is the same response as New Reno. The proposed FR-TCP is based on New Reno congestion control in order to promote introducing FR-TCP to TCP heterogeneous situation. For each loss event, cwnd is set to the value of the ssthresh in order to avoid network congestion. To keep throughput above FR, FR-TCP always apply ssthresh to FR conscious value.

Since sending rate is computed as

$$sending\ rate = \frac{cwnd \times Size_{seg}}{RTT}, \qquad (6)$$

target cwnd should be

$$cwnd = \frac{RTT \times target\ rate}{Size_{seg}}, \qquad (7)$$

where $Size_{seg}$ is sending segment size. In order to let every flow send packets at the rate more than FR, minimum of cwnd, which is ssthersh, should be set $\frac{RTT \times FR}{Size_{seg}}$.

In this paper, for clarity, we call existing ssthresh $ssthresh_{exist}$ and new ssthresh $ssthresh_{FR}$. Sender FR-TCP calculates $ssthresh_{FR}$ with FR and estimated RTT☆.

☆ Estimation of current RTT (srtt) is calculated as follows, with measured RTT (rtt) and constant number (α).
$srtt = \alpha \times srtt + (1 - \alpha) \times rtt$

Therefore FR-TCP controls the cwnd as follows:

$$cwnd = cwnd + \frac{1}{cwnd} \quad \text{(each received ACK)} \quad (8)$$

$$Ssthresh_{FR} = \frac{RTT \times FR}{Size_{seg}} \qquad (9)$$

$$cwnd = ssthresh_{FR} \quad \text{(each loss event)} \quad (10)$$

In our system, each flow can always keep its cwnd more than $ssthresh_{FR}$. When the flow's cwnd is $ssthresh_{FR}$, sending rate will be $FR(= \frac{ssthresh_{FR} \times Size_{seg}}{RTT})$. Therefore the flow can send packets at least at FR. And when there is uncontracted link bandwidth, each flow can make use of it.

3.3.2 Utilization of Explicit Congestion Notification (ECN) [14]

We also introduce ECN mechanism into FR-TCP congestion control mechanisms. When ECN detects congestion, FR-TCP sets cwnd to $ssthresh_{FR}$. ECN enables to avoid decreasing sending rate caused by retransmission.

When packet loss is detected by reception of duplicated ACKs or time out of retransmission timer, TCP congestion control prefers retransmitting lost packets to transmitting normal data. During retransmission, no new segments are transmitted and transmission rate is very low. Since FR-TCP retransmission control is based on New Reno, overhead for retransmission is proportional to RTT. Flows with longer RTT take longer time to recover from retransmission than flows with shorter RTT. Then only FR-TCP window control does not achieve all data transmission at above FR.

ECN mechanism enables FR-TCP to detect congestion without packet loss. After FR-TCP detects congestion, it set cwnd to $ssthresh_{FR}$ to keep data transmission at above FR.

4. Evaluation

We evaluate proposed FR-TCP through computer simulations in comparison to New Reno with ns-2. The simulated network is shown in **Fig. 2**. Bandwidth of each link is 10 Mbps. Multiple flows are aggregated in Diffserv network. Link delay is changed in order to aggregate TCP flows with different RTTs. At the core router, RIO [15] 〔Random Early Detection (RED) with IN and OUT〕 was simulated. RIO implements RED [16] dropping policy, instead of

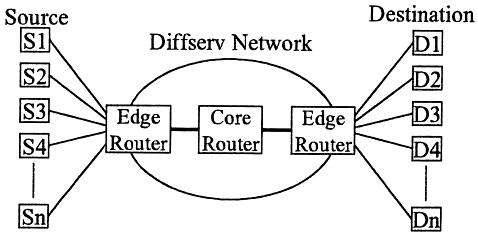

Fig. 2 Simulated network.

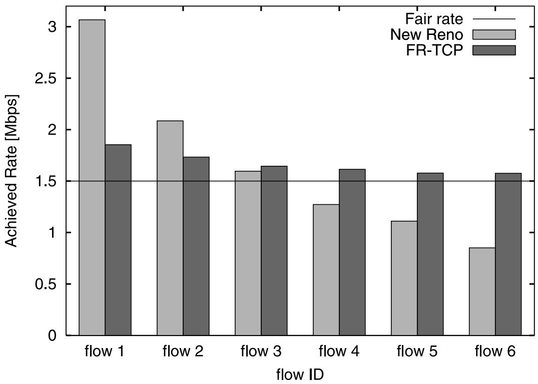

Fig. 3 Result of situation where RTTs of flow are different. Throughput of each flow is shown. RTTs of flow 1 to 6 are 40, 80, 120, 160, 200 and 240 ms.

drop tail, for both IN and OUT of profile traffic on packets.

We evaluate FR-TCP in point of fairness of bandwidth, adaptability to multiple contracts, adaptability to dynamic change of flow number and adaptability to heterogeneous environment.

4.1 Fairness of Bandwidth

We run three types of simulation to analyze fairness of bandwidth.

4.1.1 Situation Where RTTs of Flow Are Different

Figure 3 shows measured throughput of each flow with FR-TCP compared to New Reno through computer simulation as we have explained earlier. The contracted bandwidth 9 Mbps is shared among all 6 flows in the aggregate and FR is 1.5 Mbps. RTTs of flows from 1 to 6 are 40, 80, 120, 160, 200 and 240 ms, respectively.

Fairness index (F)[17] of New Reno and FR-TCP are 0.835 and 0.996, respectively. Where throughput of ith flow is R_i and number of flows is n, F is calculated as follows. It is fair, when F is close to 1.

$$F = \frac{(\sum_{i=1}^{n} R_i)^2}{n(\sum_{i=1}^{n} {R_i}^2)} \tag{11}$$

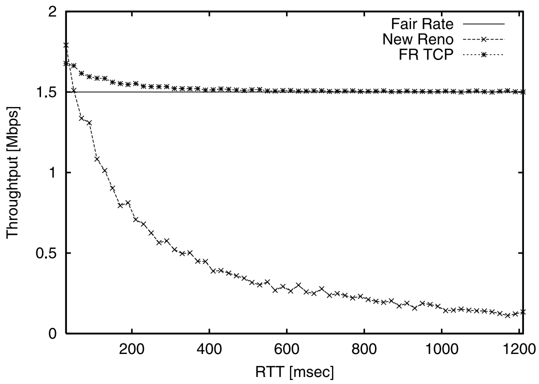

Fig. 4 Result of situation where RTT of only one flow is different from the others. It shows throughput dynamics of the flow when RTT is changing. RTTs of flow 1 to 5 are 40 ms and flow 6 is t ms.

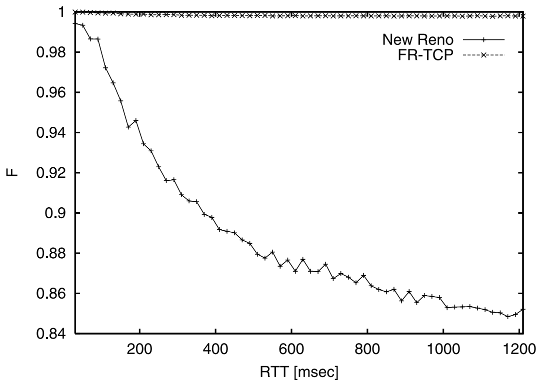

Fig. 5 Result of situation where RTT of only one flow is different from the others. It shows fairness of bandwidth against RTT is shown. RTTs of flow 1 to 5 are 40 ms and flow 6 is t ms.

As for F, proposed FR-TCP improves fairness of bandwidth in comparison to New Reno. FR-TCP in Fig. 3 shows that every flow using FR-TCP transmits data at above FR, while flows with longer RTTs transmit less data than FR when New Reno is used.

4.1.2 Situation Where RTT of Only One Flow Is Different from The Others

Figure 4 shows FR-TCP throughput dynamics of one flow with changed RTT in the network and **Fig. 5** shows FR-TCP dynamics of F, compared to New Reno. In this simulation, we evaluate fairness of bandwidth when multiple flows with same RTT except one are aggregated and RTT of the only one flow is changed. The contracted bandwidth 9 Mbps is shared among all 6 flows in the aggregate and FR is 1.5 Mbps. RTTs of flows from 1 to 5 are

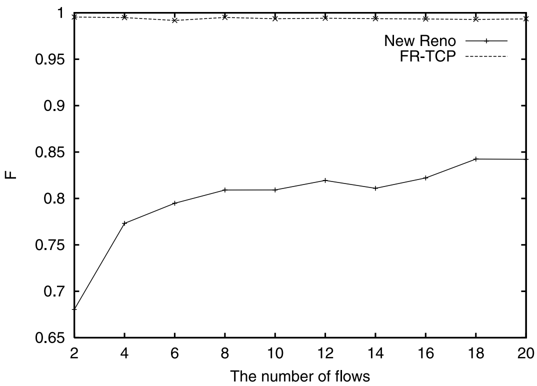

Fig. 6 Result of situation where different number of flows are aggregated. Fairness of bandwidth against the number of flows is shown. RTTs of one half of the total flows are 40 ms and RTTs of the others are 240 ms.

40 and RTT of flow 6 is t (from 30 to 1210) ms.

As we see in Fig. 4, throughput of flow 6 in New Reno decreases when RTT increases. On the other hand, FR-TCP transmits data of flow 6at more than FR even though only one of RTT is much longer than the rest. Moreover Fig. 5 assures that F of FR-TCP keeps around 1 and FR-TCP improves fairness of bandwidth. It is concluded that FR-TCP is more effective as changing RTT is relatively longer.

4.1.3 Situation Where Different Number of Flows Are Aggregated

Figure 6 shows FR-TCP dynamics of F, compared to New Reno when the various number of flows with different RTTs is aggregated. The contracted bandwidth 9 Mbps is shared among 2 to 20 flows in the aggregate. RTTs of one half of the total flows are 40 ms and RTTs of the others are 240 ms.

From Fig. 6, F of New Reno is far from 1 regardless of the number of flows. It means New Reno allocates unfair bandwidth. On the other hand, FR-TCP keeps F around 1. Therefore, FR-TCP can allocate fair bandwidth regardless the number of flows.

4.2 Adaptability to Multiple Contracts

Table 2 reflects measurement of throughput for whole contract (*Rate*) and fairness of bandwidth per contract (F), compared to New Reno. In this simulation, we evaluate FR-TCP when 2 AF bandwidth guarantee agreements, A and B, are contracted. Contracted bandwidth of A is 6 Mbps to be shared among all 6 flows in the aggregate and FR is 1.0 Mbps. Contracted bandwidth of B is 3 Mbps to be shared among all

Table 2 Results of situation where 2 AF bandwidth guarantee agreements are contracted. Rate and F of Contract A and B are expressed.

SLA	New Reno		FR-TCP	
No	*Rate*	*F*	*Rate*	*F*
A	6.3427	0.89063	6.5413	0.99840
B	3.6567	0.89486	3.4587	0.99622

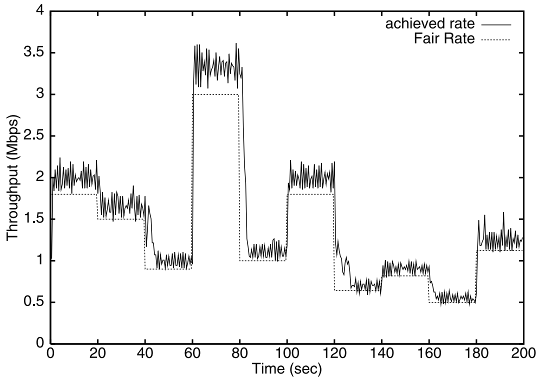

Fig. 7 TCP throughput dynamics at sender in the case of changing number of flows.

6 flows in the aggregate and FR is 0.5 Mbps. RTT of 6 flows in the both contract are 40, 80, 120, 160, 200 and 240 ms.

As shown in Table 2, FR-TCP improves fairness of bandwidth in both contracts, keeping almost same total rate as New Reno. We conclude that FR-TCP improves fairness when multiple contracts are made in the same link.

4.3 Adaptability to Dynamic Change of Flow Number

Figure 7 shows throughput dynamics at sender, when the number of aggregated flows is dynamically changing. The contracted bandwidth 9 Mbps is shared among randomized flows changing every 20 seconds. RTT of every flow is 40 ms.

Even if FR is dynamically changing, we see FR-TCP adapts its sending rate to the change promptly. It is found that FR-TCP works effectively even when the number of flow is changed dynamically.

4.4 Adaptability to Heterogeneous Environment

Figure 8 shows measured throughput of each flow of FR-TCP and New Reno in 2 conditions. In this simulation we evaluate FR-TCP in heterogeneous environment where New Reno and FR-TCP exist within the aggregate. The contracted bandwidth 9 Mbps is shared among all 6 flows in the aggregate and fair Rate is 1.5 Mbps. The condition of 6 flows is shown in **Table 3**.

In condition B, every flow of FR-TCP (flow 1

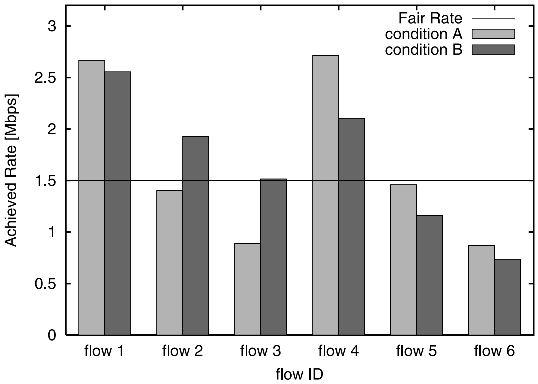

Fig. 8 Result of adaptability to heterogeneous environment.

Table 3 The condition of flow in the simulation of adaptability to heterogeneous environment.

flow No.	1	2	3	4	5	6
RTT (ms)	40	120	240	40	120	240
condition A	New Reno			New Reno		
condition B	FR-TCP			New Reno		

to 3) realizes throughput higher than FR under such heterogeneous environment. F of condition A and condition B are 0.829 and 0.884, respectively. As F of condition B shows higher value than one of condition A, condition B is more fair than condition A. This is because FR-TCP improves F of FR-TCP flow (flow 1 to 3) in condition B, while F of New Reno flow (flow 4 to 6) in condition B keeps the same fairness as condition A. As a result, FR-TCP improves fairness and realizes FR in heterogeneous environment with New Reno.

5. Related Work

So far, much research on fair bandwidth within the aggregate has been done.

Lin, et al.[18] reported enhancement of Time Sliding Window profiler and RIO queue management mechanism at the core routers for resource allocation. However, this proposal requires changing mechanism at core router, which seems to be the negation of the Diffserv principles and has concerns about performance decrement.

As far edge router, Yeom, et al.[19] proposed that edge routers manage information of each flow and improved fairness when flows with different RTTs were aggregated. Nandy et al.[20] improved queuing algorithm to solve unfair allocation when flows with different RTTs are aggregated. In these proposals edge routers need to manage RTT of each flow, which leads to scalability problem.

CSFQ[21] is also proposed for fair queuing. In this architecture, edge routers have to compute each flow's rate. Estimating each flow's rate also leads to scalability problem when number of flow gets greater. In FR-TCP, network nodes do not have to manage information of each flow. Edge routers manage information of the aggregate, count the number of flow and estimate fair bandwidth, which lightens scalability problem other proposals has.

Another approach for fair bandwidth allocation, taken by XCP[22], uses explicit feedback from routers for congestion control. XCP need make much change in packet header, end host and edge router. On the other hand, FR-TCP realizes fair allocation with easy calculations.

Fang, et al.[23] set window increase during congestion avoidance and increased cwnd of every flow at a constant rate to remove the influence of RTT differences. When New Reno flows and flows with proposed mechanism are aggregated, a problem on unfairness arises due to different congestion control mechanism. The cause of this problem is that New Reno flows increase window at a different rate compared to flows with proposed mechanism.

Lahanas, et al.[24] also proposed congestion control algorithm for flows with different RTTs and calculated appropriate window increase and decrease. This proposal is efective but needs complicated calculation. FR-TCP realizes fair bandwidth allocation with easy calculation.

6. Conclusion

In this paper we propose FR-TCP to improve the fairness of bandwidth in cooperation with network nodes and end hosts over Diffserv networks. We (1) define a fair bandwidth as FR, (2) introduce the FR notification mechanism to sender TCP and (3) propose congestion control mechanism with notified FR.

We evaluate the capability of FR-TCP mechanism through computer simulations. The result shows FR-TCP improves the fairness of bandwidth allocation efficiently regardless of RTT and the number of aggregated flows. When multiple contracts are made in the same link, FR-TCP realizes the fair allocation of each contract. FR-TCP controls sending rate promptly where the number of flows is dynamically changing. Even though FR-TCP and the current TCP are heterogeneous, FR-TCP showed the effectiveness. Therefore, the proposed FR-TCP makes greater contribution to the fair bandwidth allocation under various environments.

An immediate extension to this work is fair allocation of uncontructed bandwidth. FR-TCP does not realize fair allocation of uncontructed bandwidth completely. Some mechanism that improves fairness of unused resource allocation is needed.

Acknowledgments This work was supported by a special grant from COE.

References

1) Nichols, K., Blake, S., Baker, F. and Black, D.: Definition of the Differeniated Service Field (DS Field) in the IPv4 and IPv6 Headers, *IETF*, RFC2474 (Dec. 1998).
2) Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and Weiss, W.: An Architecture for Differentiated Services, *IETF*, RFC2475 (Dec. 1998).
3) Jacobson, V., Nichols, K. and Poduri, K.: An Expedited Forwarding PHB, *IETF*, RFC2598 (Jun. 1999).
4) Heinanen, J., Baker, F., Weiss, W. and Wroclawski, J.: Assured forwarding PHB group, *IETF*, RFC2597 (Jun. 1997).
5) Fang, W.: Differentiated services: Architecture, mechanisms and an evaluation, Princeton Ph.D. Thesis (2000).
6) Floyd, S. and Henderson, T.: The NewReno Modification to TCP's Fast Recovery Algorithm, *IETF*, RFC2582 (Apr. 1999).
7) Stevens W.: CP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, *IETF*, RFC2001 (Jan. 1997).
8) Allman, M., Paxson, V. and Stevens, W.: TCP Congestion Control, *IETF*, RFC2581 (1999).
9) Padhye, J., Firoiu, V., Towsley, D. and Krusoe, J.: Modeling TCP Throughput: A Simple Model and its Empirical Validation, *ACM SIGCOMM '98*, pp.303–314 (1998).
10) Ibanez, J. and Nichols, K.: Preliminary Simulation Evaluation of an Assured Service, *IETF*, Draft (1998).
11) Tartarelli, S., Brunner, M. and Banchs, A.: An one-to-any Assured Rate Per-Domain Behavior for Differentiated Services, Internet draft, *IETF* (Feb. 2001).
12) The network simulator ver.2 - ns-2. http://www.isi.edu/nsnam/ns/
13) Waters, G.J., et al.: Policy framework architecture, IETF Internet Draft (Feb. 1999).
14) Ramakrishnan, S.: A proposal to add explicit congestion notiffcation (ecn) to ip, *IETF*, RFC2481 (Jan. 1999).
15) Clark, D. and Fang, W.: Explicit allocation of best-effort packet delivery service, *IEEE/ACM Transactions on Networking*, Vol.6, No.4, pp.362–373 (1998).
16) Braden, B., et al.: Recommendations on Queue Management and Congestion Avoidance, *IETF*, RFC2309 (Apr. 1998).
17) Jain, R.: Throughput Fairness Index: An Explanation, ATM Forum 99-0045 (1999).
18) Lin, W., Zheng, R. and Hou, J.: How to make assured services more assured, *ICNP'99*, pp.182–191 (1999).
19) Yeom, I. and Reddy, N.: Realizing throughput guarantees in a differentiated services network, *ICMCS*, Vol.2, pp.372–376 (1999).
20) Seddigh, N. and Nandy B.: Intelligent Traffic Conditioners for Assured Forwarding Based Differentiated Services Networks, *NETWORKING 2000*, pp.540–554 (2000).
21) Stoica, I., Shenker, S. and Zhang, H.: Core-stateless fair queuing: A scalable architecture to approximate fair bandwidth allocations in high speed networks, *Proc. ACM SIGCOMM* (Aug. 1998).
22) Katabi, D., Handley, M. and Rohrs. C.: Congestion Control for High Bandwidth-Delay Product Networks, *ACM SIGCOMM* (2002).
23) Fang, W. and Peterson, L.: TCP mechanisms for Diff-Serv Architecture, Princeton University, CS Dept., Technical Report TR-605-99 (1999).
24) Lahanas, A. and Tsaoussidis, V.: τ-AIMD for asynchronous receiver feedback, in Proceedings, *8th IEEE International Symposium*, Vol.2, pp.735–740 (2003).

(Received November 18, 2003)
(Accepted September 3, 2004)

Tomoya Hatano received his B.S. and M.E. in Department of Information and Computer Science from Keio University, Japan in 2002 and 2004 respectively. He works at NTT Access Network Service Systems Laboratories, NTT Corporation. His research interests are Internet QoS.

Hiroshi Shigeno received his B.S., M.E. and Ph.D. degrees all in instrumentation engineering from Keio University, Japan in 1990, 1992 and 1997 respectively. Since 1998 he has been with the Department of Information and Computer Science, Keio University, Yokohama, Japan, where he is now an associate professor. His current research interests are in computer network architecture and protocols, mobile and ubiquitous computing, and multimedia applications on highspeed networks. He is a member of ACM, IEEE, IEICE, and IPSJ.

Ken-ichi Okada received his B.S., M.E. and Ph.D. in instrumentation engineering from Keio University, in 1973, 1975 and 1982 respectively. He is currently an associate professor in the Department of Information and Computer Science at Keio University. His research interests include CSCW, groupware, human computer interaction and mobile computing. He has published over 100 technical papers and reports and books entitled “Collaboration and Communication,” “Intelligent Inspired Information Society,” “Designing Communication and Collaboration Support Systems” and “Introduction to Groupware.” He is a member of IEEE, ACM, and IPSJ. He was a chair of SIGGW, a chief editor of IPSJ Journal, and an editor of IEICE Transactions. Dr. Okada received the IPSJ Best Paper Award in 1995, 2001 and IPSJ 40th Anniversary Paper Award in 2000.

Yutaka Matsushita was born in 1939. He received his B.E. in Electrical Engineering from Keio University, M.S. in Computer Science from University of Illinois, and Ph.D. in Electrical Engineering from Keio University, respectively in 1963, 1969, and 1978. From 1963 to 1988, he joined OKI Electric Industry Ltd. Where he had served as the director of the integrated systems division. From 1989 until 2002, he was a professor in Department of Information and Computer Science of Keio University. From April 2003 he is the Dean of School of Computer Science in Tokyo University of Technology. His more recent works have been in the area of groupware, mobile computing, Internet QOS, fragrance over Internet and human computer interaction. He served as the editor-in-chief of the journal of IPSJ and the chairman of SIG DPS and SIG GW of IPSJ, and the vice-president of IPSJ. He also served as the chairman of SIG MIS and SIG IN of IEICE, and the chairman of SIG CyberSpace of the Virtual Reality Society of Japan. He ia an author or co-author of more than 30 books. He received IPSJ best author award in 1993 and IPSJ 40th anniversary 90’s decade best author award in 2000. He received best paper awards twice in 1995 and 2000 respectively. He received distinguished services award of IPSJ in 2001. He is a fellow of both IEICE and IPSJ. He is a member of IPSJ, IEICE, IEEE, and ACM.