

# A Packet Partition Scheduling Mechanism for Bandwidth Aggregation over Multiple Paths

Pyung-Soo Kim<sup>1</sup>, Joo Young Yoon<sup>2</sup>, and Hanlim Kim<sup>2</sup>

<sup>1</sup>*Department of Electronics Engineering, Korea Polytechnic University,  
Siheung-City, Kyonggi-Do, 429-793, Korea*

[pskim@kpu.ac.kr](mailto:pskim@kpu.ac.kr)

<sup>2</sup>*KT Infra Lab.*

*17 Woomyeon-dong, Seocho-gu, Seoul, 137-792, Korea*

## Abstract

*This paper proposes a packet partition scheduling mechanism for bandwidth aggregation over end-to-end multi-path through multiple network interfaces. The proposed mechanism effectively partitions packets on the appropriate end-to-end path through the corresponding network interface. Each network interface is considered to have a partition counter associated with it. This partition counter varied by both weighted capacity and partitioned packets is used to determine if a network interface has enough credits to partition incoming packets on multiple paths. The capacity unit is shown to be a useful design parameter to make the performance of the proposed mechanism as good as possible. Through computer simulations, it is shown that the proposed mechanism can partition well randomly incoming packets for the multi-path communication and can outperform the simple weighted packet partition scheduling mechanism.*

## 1. Introduction

To access the Internet, today's mobile computers often have more than one network interfaces such as WiBro(IEEE 802.16e), 3G(GPRS, CDMA2000), WLAN(IEEE 802.11), Bluetooth(IEEE 802.15), etc [1]-[6]. These mobile computers are called "multi-homing capable" because they can connect to multiple networks. In recent, to explain the motivations for mobile nodes using multiple network interfaces and the scenarios where this may end up using multiple global addresses on their interfaces, the multi-homing has been investigated from an end-node point of view not from a site point of view [7].

Once multiple network interfaces are offered, users may want to select the most appropriate set of network interface(s) depending on the network environment,

particularly in wireless networks which are mutable and less reliable than wired networks. Users may also want to select the most appropriate interface per communication type or to combine a set of interfaces to get sufficient bandwidth or more bandwidth.

This paper considers a new wireless communication access paradigm called the "bandwidth aggregation" mechanism which combines a set of interfaces to provide end-to-end multi-path communication in order to improve the end users' communication experience. As shown in [8][9], in the bandwidth aggregation, the bandwidth offered by the multiple network interfaces can be aggregated to improve quality or support demanding applications that need high bandwidth. Through the bandwidth aggregation, users can accumulate enough bandwidth to meet their exact needs by combining more than one single Internet connection.

The bandwidth aggregation scheme consists of two main functional mechanisms; the bandwidth estimation mechanism and the packet partition scheduling mechanism. The bandwidth estimation mechanism has to measure or predict the available bandwidth of an end-to-end path which is its remaining capacity, that is, the amount of traffic that can be sent along the path without congesting it. The packet partition scheduling mechanism has to effectively partition packets on the appropriate end-to-end path through the corresponding network interface.

This paper proposes a packet partition scheduling mechanism for bandwidth aggregation over end-to-end multi-path through multiple network interfaces. In the proposed packet partition scheduling mechanism, each interface is considered to have a "partition counter" associated with it. This partition counter is used to determine if an interface has enough credits to partition a packet. The partition counter can get credits by the "weighted capacity" in bytes. This weighted counter is computed by multiplication of the capacity unit and the

weight. The capacity unit in bytes is a useful design parameter and thus can affect on the performance of the proposed mechanism. The weight is determined proportionately from the available bandwidth of end-to-end path through corresponding interface. In this paper, the available bandwidth of end-to-end path is assumed to be estimated from existing estimation mechanisms and thus known [10][11]. The weighed capacity is operated at the byte level and is added more to the partition counter with higher weight than that with less weight. The partition counter is also decreased by the size of packets being partitioned. Thus, the partition counter for each network interface is varied by partitioned packets as well as weighted capacity.

Firstly, the operation procedure of the proposed mechanism is described and its example is briefly given. Secondly, two performance indices for the proposed mechanism are described. As mentioned previously, the capacity unit is shown to be a useful design parameter to make the performance of the proposed mechanism as good as possible. To show how the capacity unit affects on performances indices, computer simulations are performed for some cases according to the capacity unit, which can provide practical guidance on the choice of a capacity unit. Finally, the proposed packet partition scheduling mechanism is compared with the simple weighted packet partition scheduling mechanism. From simulation results, the proposed mechanism is shown to be superior to the simple weighted mechanism.

The paper is organized as follows. In Section 2, the bandwidth aggregation is reviewed briefly. In Section 3, a packet partition scheduling mechanism is proposed for bandwidth aggregation over end-to-end multi-path. In Section 4, extensive computer simulations are performed. Finally, concluding remarks are made in Section 5.

## 2. Bandwidth Aggregation

The bandwidth aggregation is one of multiple access services enabled by simultaneous use of multiple network interfaces and the multi-path communication. In the bandwidth aggregation, the bandwidth offered by multiple network interfaces can be aggregated to improve quality or support demanding applications that need high bandwidth.

The bandwidth aggregation scheme consists of two main mechanisms; the bandwidth estimation mechanism and the packet partition scheduling mechanism.

The bandwidth estimation mechanism has to measure or predict the available bandwidth of an end-

to-end path which is its remaining capacity, that is, the amount of traffic that can be sent along the path without congesting it. Recently, the area of end-to-end available bandwidth estimation has attracted considerable interest for many quality of service (QoS) applications such as voice over IP (VoIP) and internet protocol TV (IPTV) [10][11]. Basically the available bandwidth is an important metric for several applications such as overlay networks, dynamic server selection, and inter-domain path monitoring. As a result, several estimation techniques and tools based on active measurements have been developed and thus can be applied for the bandwidth aggregation in the architecture to supporting multi-path communication. The packet partition scheduling mechanism not only has to effectively partition packets on the appropriate end-to-end path through the corresponding network interface, but also minimizes delay experienced by packets due to potential reordering caused by varying characteristics (delay, bandwidth, loss) of the multiple paths. Thus, essential to such a packet partition scheduling mechanism is the scheduler that partitions incoming packets to multiple interfaces in consideration of available bandwidth of end-to-end path through corresponding network interface. It is necessary for the scheduler to partition incoming packets in fairness so that the multi-interfaced node can achieve its full bandwidth aggregation potential.

## 3. Proposed Mechanism

### 3.1 Operation Procedure

Consider the multi-interfaced node which effectively aggregates bandwidths of end-to-end multi-path through multiple network interfaces.

In the proposed packet partition scheduling mechanism, each interface is considered to have a “partition counter” associated with it. This partition counter is used to determine if an interface has enough credits to partition a packet. The partition counter can get credits by the “weighted capacity” in bytes. The weighted capacity is defined as follows:

$$\text{Weighted capacity} = \text{Capacity unit} * \text{Weight.}$$

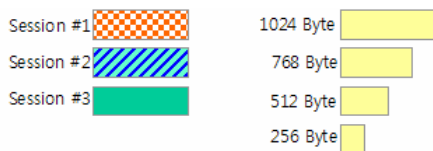
The capacity unit in bytes is a useful design parameter and thus can affect on the performance of the proposed mechanism. The weight is determined proportionately from the estimated available bandwidth of end-to-end path through corresponding interface. The weighed capacity is operated at the byte level and is added more to the interface with higher weight than that with less weight. The partition counter

is also decreased by the size of packets being partitioned. Thus, the partition counter for each network interface is varied by partitioned packets as well as weighted capacity.

The operation procedure for each round is as follows. For the first interface, packets are partitioned when corresponding partition counter is greater than the incoming packet's size. If it is lower, then the partition counter is increased by the weighted capacity and the incoming packet is partitioned on the current interface. Then the partition counter is decreased by the size of packet being partitioned. If the partition counter is still lower than the incoming packet's size, the incoming packet held back until the proposed mechanism moves on the next interface. After visiting all network interfaces, the round is finished. The above operation procedure in next round is repeated when there are incoming packets.

### 3.2 Example

In this example, the multi-interfaced node has three network interfaces. These interfaces are called the Green (high speed), Yellow (medium speed), Red (low speed) interfaces, respectively. Since this paper focuses on the packet partition scheduling mechanism, available bandwidths for three paths through corresponding interfaces are assumed to have fixed weight ratio 4:2:1. The capacity unit is set by 256 bytes and thus these interfaces have weighted capacity 1024, 512, 256 bytes, respectively. Initially, all partition counters for each interface are set with 0. There are three kinds of session flows and they have four kinds of packet type with different sizes such as 256, 512, 768, 1024 bytes as shown in Figure 1..



**Figure 1.** Session flows and packet types in example

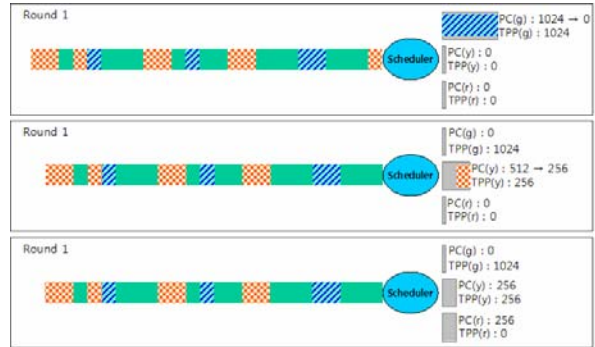
#### ▪ Incoming Packets

As shown in Figure 2, incoming packets are waiting to be partitioned to the most appropriate interface.



**Figure 2.** Incoming Packets

#### ▪ 1<sup>st</sup> Round



**Figure 3.** Partitioned packets at the 1<sup>st</sup> Round

At Green interface, since the partition counter, PC(g), is less than the incoming packet's size, the weighted capacity is added and thus PC(g)=1024. Then, since the PC(g) is not less than the incoming packet's size, the incoming packet is partitioned to Green interface and thus PC(g)=0. Currently, total amount of partitioned packets, TPP(g), to Green interface is 1024 bytes. At Yellow interface, since PC(y) is less than the incoming packet's size, the weighted capacity is added and thus PC(y)=512. Then, since the PC(y) is not less than the incoming packet's size, the incoming packet is partitioned to Yellow interface and thus PC(y)=256. Currently, TPP(y)=256. At Red interface, since PC(r) is less than the incoming packet's size, the weighted capacity is added and thus PC(r)=256. Then, since the PC(r) is still less than the incoming packet's size, move to Green interface. Currently, TPP(r)=0. Partitioned packets to three interfaces at the 1<sup>st</sup> round are shown in Figure 3.

#### ▪ 2<sup>nd</sup> Round

At Green interface, since the PC(g) is less than the incoming packet's size, the weighted capacity is added and thus PC(g)=1024. Then, since the PC(g) is not less than the incoming packet's size, the incoming packet is partitioned to Green interface and thus PC(g)=256. Currently, TPP(g)=1792. At Yellow interface, since PC(y) is less than the incoming packet's size, the weighted capacity is added and thus PC(y)=768. Then, since the PC(y) is not less than the incoming packet's size, the incoming packet is partitioned to Yellow interface and thus PC(y)=256. Currently, TPP(y)=768.

At Red interface, since  $PC(r)$  is less than the incoming packet's size, the weighted capacity is added and thus  $PC(r)=512$ . Then, since the  $PC(r)$  is still less than the incoming packet's size, move to Green interface. Currently,  $TPP(r)=0$ . Partitioned packets to three interfaces at the 1<sup>st</sup> round are shown in Figure 4.

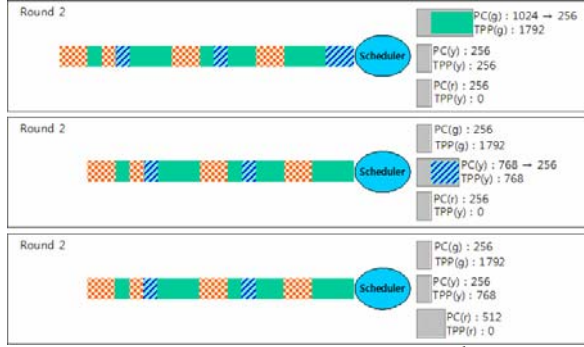


Figure 4. Partitioned packets at the 2<sup>nd</sup> Round

The above operation procedure in next round is repeated when there are incoming packets.

### 3.3 Performance Indices and Useful Design Parameter

There are two performance indices in the proposed mechanism. The one is the ratio of partitioned packets in bytes, which should be close to the weight ratio of network interfaces to make the fair partition performance as good as possible. The other is the amount of the partitioned packets in bytes, which should be much to make the end-to-end throughput performance as good as possible. However, the end-to-end throughput is determined when partitioned packets arrive at the receiver-side and dependent on the available end-to-end path bandwidth. Therefore, the ratio of partitioned packets can be more important performance index than the amount of the partitioned packets.

As mentioned before, the capacity unit is a useful design parameter to determine weighted capacity that affect on both the ratio of partitioned packets and the amount of the partitioned packets. Too big value of the capacity unit can introduce excessive credits for network interfaces, which means that network interfaces have enough credits to partition a packet. Thus, incoming packets are more likely to be partitioned simultaneously on every interfaces each round. Actually, this is not fair partition according to weights for each network interfaces, which can thus degrade the performance for the ratio of partitioned packets. On the other hand, too small value of the capacity unit can introduce deficient credits for

network interfaces, which mean that network interfaces don't have enough credits to partition a packet. Thus, incoming packets are less likely to be partitioned on every interfaces each round, which can thus degrade the performance for the amount of partitioned packets.

Thus, the important issue here is how to choose an appropriate capacity unit to make the performance of the proposed mechanism as good as possible. Following computer simulations will provide practical guidance on the choice of the capacity unit.

## 4. Computer Simulations

In this section, the computer simulation is performed to evaluate the proposed packet partition scheduling mechanism. As shown in Section 3.2, the multi-path communication is considered for multi-interfaced node with three network interfaces. Available bandwidths for three paths through corresponding interfaces are assumed to have fixed weight ratio 4:2:1. To make a clearer verification, 30 simulations are performed and each simulation generates randomly incoming packets with four kinds of packet type with different sizes such as 256, 512, 768, 1024 bytes. Through extensive computer simulations, it is shown that the proposed mechanism can partition well randomly incoming packets for the multi-path communication.

### 4.1 Performance Evaluation According to Capacity Unit

To show how the capacity unit affects on performances indices, extensive computer simulations are performed for four cases according to the capacity unit. For each case, the capacity unit is set by 128, 256, 384, 512 bytes, respectively. For four cases, Table 1 shows average values for the amount of partitioned packets and the ratio of partitioned packets for each interface. As shown in Table 1 and Figure 5, simulation results illustrate the proposed mechanism's compromise between the amount of partitioned packets and the ratio of partitioned packets according to the capacity unit. As mentioned before, too big value of the capacity unit can introduce excessive credits for network interfaces, which cannot provide fair partition according to weights for each network interfaces and thus degrade the performance for the ratio of partitioned packets. On the other hand, too small value of the capacity unit can introduce deficient credits for network interfaces, which can thus degrade the performance for the amount of partitioned packets. When the capacity unit is 256 bytes, the simulation

result is shown to be more favourable than other cases from the view of both the ratio of partitioned packets and the amount of the partitioned packets.

#### 4.2 Performance Comparison with Simple Weighted Packet Partition Scheduling Mechanism

The proposed packet partition scheduling mechanism is compared with the simple weighted packet partition scheduling mechanism. For each round, the simple weighted mechanism partitions packets according to only the weight ratio of network interfaces without the partition counter and the capacity unit. That is, the simple weighted mechanism does not consider the size of incoming packets unlike the proposed mechanism.

As shown in Table 1, Figure 6 and 7, the proposed mechanism is shown to be superior to the simple weighted mechanism for both the ratio of partitioned packets and the amount of the partitioned packets. In particular, as shown in Table 2, the performance variation of the proposed mechanism is smaller than that of the simple weighted mechanism. This observation is from that the proposed mechanism partitions packets with the consideration of incoming packets' size using the partition counter and the capacity unit, while the simple weighted mechanism does not since long-sized packets can be partitioned on specific network interface unfairly.

#### 5. Conclusions

In this paper, the packet partition scheduling mechanism has been proposed for bandwidth aggregation over end-to-end multi-path through multiple network interfaces. The proposed mechanism effectively partitions packets on the appropriate end-to-end path through the corresponding network interface. Each network interface is considered to have a partition counter associated with it. This partition counter varied by both weighted capacity and partitioned packets is used to determine if a network interface has enough credits to partition incoming packets on multiple paths. The capacity unit has shown to be a useful design parameter to make the performance of the proposed mechanism as good as possible. Via extensive computer simulations, it has been shown that the proposed mechanism can partition well randomly incoming packets for the multi-path communication and can outperform the simple weighted packet partition scheduling mechanism.

#### 6. Acknowledgement

This work was supported by the IT R&D program of MKE/IITA. [2008-S001-01, Development of Original Technology for Next-generation Tactical Defense Communication Network]

#### 7. References

- [1] 3rd Generation Partnership Project, "3GPP system to WLAN interworking: System description," TR 23.234, <http://www.3gpp.com>, June 2004.
- [2] Chakravorty, R., Vidales, P., Subramanian, K., Pratt, I., Crowcroft, J., "Performance issues with vertical handovers: Experiences from GPRS cellular and WLAN hot-spots integration," In: Proc. of the IEEE PerCom Cellular and WLAN hotspots Integration, pp. 155~164, 2004.
- [3] W. Wu, N. Banerjee, K. Basu, and S. K. Das, "SIP based vertical handoff between WWANs and WLANs," IEEE Wireless Communications, Vol. 12, No. 3, pp. 66 – 72, 2005.
- [4] WiMAX Forum-NWG, "End-to-End Network System Architecture : 3GPP-WiMAX Interworking," WiMAX Forum, April 2006.
- [5] Kim, P.S. and Kim, Y.J., "New authentication scheme for vertical handovers between IEEE 802.16e and 3G wireless networks," International Journal of Computer Science and Network Security, Vol.6, No.9B, pp. 138-143, 2006.
- [6] Kim, D., Ganz, A., "Architecture for 3G and 802.16 wireless networks integration with QoS support," In: Proc. of the 2nd Int'l Conf. on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine05), pp. 1~8, 2005.
- [7] Ernst, T. et al, "Motivations and Scenarios for Using Multiple Interfaces and Global Addresses", paper-ietf-monami6-multihoming-motivation-scenario-03 (work in progress), November.
- [8] Krueger, M. et al, "Small Computer Systems Interface protocol over the Internet (iSCSI) Requirements and Design Considerations", RFC 3347, July 2002.
- [9] Chebrolu, K., and Rao, R., "Bandwidth Aggregation for Real-Time Applications in Heterogeneous Wireless Networks", IEEE Trans. on Mobile Computing, Vol. 5, No. 4, pp. 388-403, 2006.
- [10] Hu, N., and Steenkiste, P, "Evaluation and characterization of available bandwidth probing

techniques,” IEEE Journal on Selected Areas in Communications, Vol. 21, No. 6, pp. 879–894, 2003.

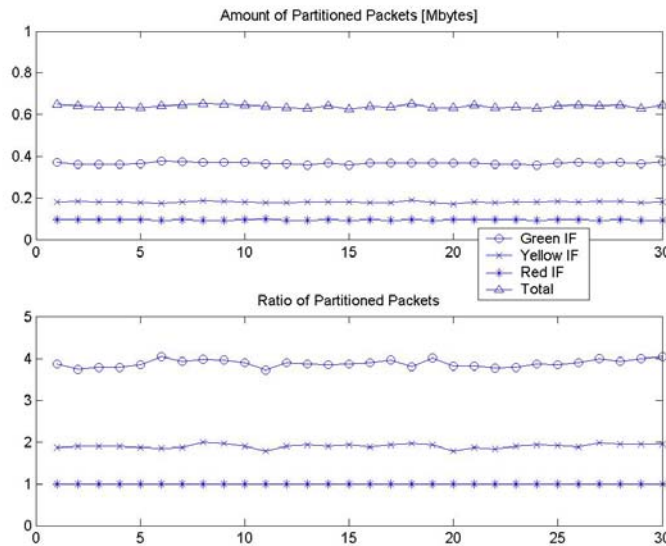
[11] Prasad, R. S., Murray, M., and Dovrolis, C., and Claffy, K., “Bandwidth estimation: metrics, measurement techniques, and tools,” IEEE Network, Vol. 17, No. 6, 2003.

**Table 1.** Simulation results according to capacity unit

Proposed Mechanism	Capacity Unit (Bytes)	Amount of partitioned packets (Average, Bytes)				Ratio of partitioned packets (Mean)
		Green IF	Yellow IF	Red IF	Total	
Proposed Mechanism	128	224093	98636	51353	374084	4.37:1.92:1
	256	366148	179251	94395	639795	3.88:1.90:1
	384	350626	258048	137062	745736	2.56:1.88:1
	512	335965	198513	179165	813644	1.88:1.67:1
Simple Weighted Mechanism		300645	151193	74845	526694	4.11:2.09:1

**Table 2.** Comparison of performance variation (Normalized by Red IF)

Proposed Mechanism		Green IF	Yellow IF	Red IF
		Max	4.04	
Proposed Mechanism	Min	3.71	1.78	1
	Mean	3.88	1.90	
	Standard Dev.	0.09	0.05	
	Max	4.50	2.33	
Simple Weighted Mechanism	Min	3.64	1.75	
	Mean	4.03	2.03	
	Standard Dev.	0.23	0.13	



**Figure 5.** Simulation result of proposed mechanism when capacity unit = 256 bytes

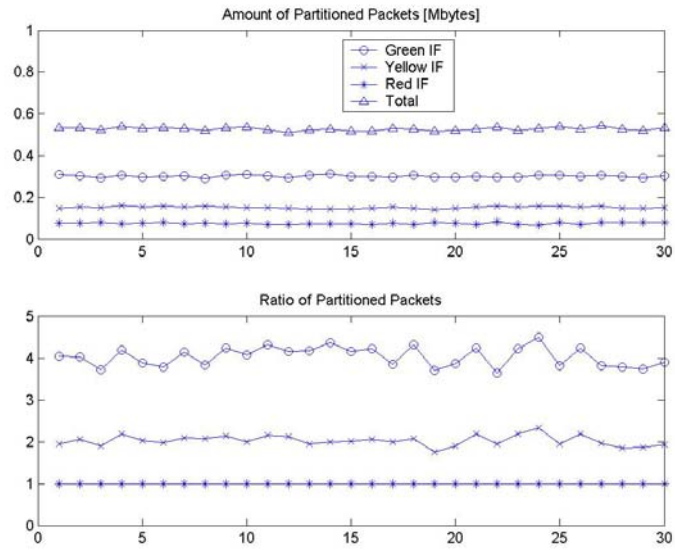


Figure 6. Simulation result of simple weighted mechanism

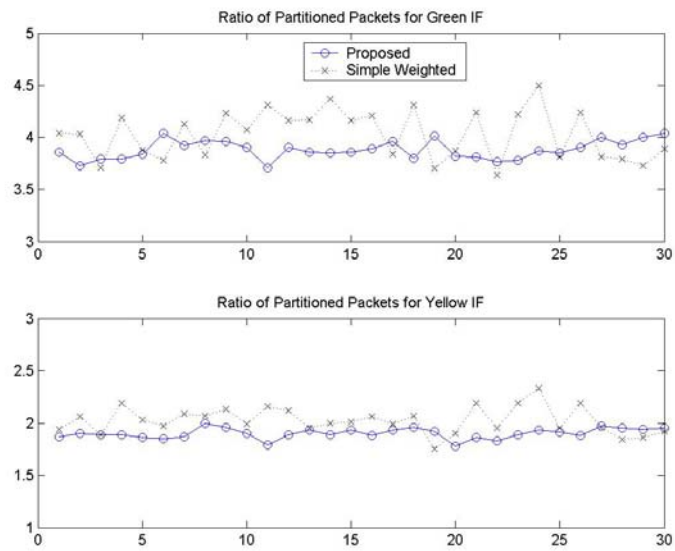


Figure 7. Ratio of partitioned packets for two mechanisms