Wireless Packet Scheduling with Signal-to-Noise Ratio Monitoring

Hiroto Aida[1], Yosuke Tamura[2], Yoshito Tobe[2], Hideyuki Tokuda[1 2]

[1]Faculty of Environmental Information,
[2]Graduate School of Media and Governance,
Keio University, 5322 Endo, Fujisawa, Kanagawa 252-8520, Japan
{*haru, tamura, tobe, hxt*}*@ht.sfc.keio.ac.jp*

Abstract

A key challenge in packet scheduling over a wireless channel is to achieve effective throughput against bursty errors. Although there have been several proposals on packet scheduling for a wireless channel, they are only based on bi-modal channel states: good or bad. In this paper we propose a new packet scheduling scheme for wireless links based on receiver's Signal to Noise Ratio (SNR). The scheme is referred to as SNR-based Packet Scheduling (SPS). In SPS, outgoing packet flows are scheduled such that the flows with higher SNR values have larger weights in scheduling. SPS can achieve effectiveness by adopting a flexible mapping between weight and SNR. We present measured results of the relationship between throughput and SNR as well as the design and implementation of SPS. In our experiments, SPS has shown its effectiveness in enhancing total throughput for hosts with good and bad wireless connectivity over a wireless LAN.

1. Introduction

The increase of wireless networks demands a special attention to data transmissions over wireless links. In particular, packet-based wireless communications will be important to the Internet work in the future. Wireless LANs have already being widely deployed in many campuses and conference rooms.

In such wireless LANs, base stations forward packets between wireless-LAN nodes or between a wireless node and a wired segment. One problem in a wireless LAN is diversity in stabilizing the connectivity between the base station and each node depending on the location of the node. A node suffers from frequent bursty errors when it has a weak connectivity with its base station. Therefore, if the base station forwards packets irreverently to stabilize connectivity with the nodes, good-put of packet in its segment will be worsened.

To overcome the problem, scheduling algorithms of fair queuing for wireless links [1, 8, 11, 12] have been proposed. These algorithms provide a variety of degree of short-term as well as long-term fairness, throughput and delay bounds for flows in the presence of channel errors. However, they have a common problem in settling a channel state; the channel state is classified into two: good or bad.

In contrast to the existing approaches, we do not classify the channel state into only two states. We take Signal-to-Noise Ratio (SNR) which is closely related to stabilizing of connection to downlink links into account for packet scheduling at the base station. Our packet scheduling scheme is referred to as SNR-based Packet Scheduling (SPS). In SPS, outgoing packets are scheduled flexibly depending on SNR at the destination of the packets.

In this paper we describe the design and the implementation of SPS. In our experiments, SPS has shown its effectiveness in enhancing total throughput for hosts with good and bad wireless connectivity over a wireless LAN.

The remainder of this paper is organized as follows. Section 2 discusses previous works in the area of queuing scheme for wireless links. Section 3 describes basic observation between SNR and TCP throughput. In Section 4 we proposes a design of SNR-based Packet Scheduling. Section 5 describes implementation of SPS. Section 6 presents an evaluation of our scheme, and we conclude the paper and future work in Section 7.

2. Related Work

There have been several recent efforts that address the issues of inefficiency due to location-dependent channel errors. Channel State Dependent Packet Scheduling (CSDPS) [1] was the first approach towards

Proceedings of the 25th Annual IEEE Conference on Local Computer Networks (LCN'00)
0742-1303/00 $10.00 © 2000 IEEE

per-flow scheduling in consideration of per-flow channel states. The CSDPS scheduler forwards packets of flows in good state while it awaits in bad state. Since CSDPS cannot provide fairness in throughput and does not address guaranteeing delay, modified schemes such as IWFQ [8], CIF-Q [11], and LTFS [12] were proposed to accommodate compensation. IWFQ prioritizes flows with recovered channel state by utilizing a characteristic that service finish time of unsent packets becomes earlier. CIF-Q sets a per-flow variable *log*, which is the difference between WFQ-based and actual service amount. CIF-Q provides compensation by prioritizing flows with larger lag. LTFS reserves bandwidth for compensation in advance. In addition to these schemes, Havana [6] framework aims at improving performance at the application-level by conducting channel state based packet scheduling with compensation. The Havana consists of three components: a channel predicator which probes for channel state, a compensator which compensates flows in bad channel state, and an adaptor which controls the rate at the application level.

The channel state defined in these proposals are models as being in one of the two possible states, good(G) or bad(B). If a flow is in good channel state, packets of the flow are sent. Otherwise, they are not sent. For instance in the case of CSDPS, wireless channel is expected to be estimated with characteristics of bursty channel errors in wireless networks, and packet loss characteristics is modeled in 2-state Markov model. In contrast to these works, we use more precise channel state and accommodates more flexible policies of packet scheduling.

3. Relationship between SNR and Throughput

As we mentioned in the previous section, previous works only assume one of the two channel states: G or B. In this section, we investigate the relationship between SNR and available TCP throughput to examine whether or not the two channel states are sufficient for packet scheduling.

The SNR represents a channel condition and is expressed as the ratio of signal to noise in electrical power. SNR is defined as follows:

$$\mathrm{SNR} = 10\log_{10}[\mathrm{S}/\mathrm{N}], \qquad (1)$$

where S and N are signal and noise, respectively. In general the theoretical value of Bit Error Ratio (BER), P_e, can be computed if a digital modulation scheme is specified. For phase shift keying (PSK) which is used in NCR WaveLAN [15], P_e is obtained as follows:

$$P_e = \frac{1}{2}\mathrm{erfc}\left(\frac{\mathrm{S}}{\mathrm{N}}\right)^{\frac{1}{2}}, \qquad (2)$$

where erfc is the error function. Thus, SNR is closely related to BER and implied to affect a throughput of a flow.

3.1. Experimental Environment

Figure 1 shows the experimental environment, used for computing TCP throughput. In Figure 1, an access point AP and a wired host CH are connected to each other by a wired 100-Mbps Ethernet. Both of AP and CH are IBM PC-AT compatible computer with 300 MHz Pentium II and 126-Mbyte RAMs. Mobile hosts MH1-MH2 are SONY VAIO PCG505EX with 233-MHz Pentium and 64-Mbyte RAMs. MH1-MH2 and AP are equipped with 2.4-GHz NCR WaveLAN [15]. MH1-MH2 communicate with CH via AP. FreeBSD 3.4-RELEASE is installed on MH1-MH2, CH and AP with a slight modification for bridging at AP.

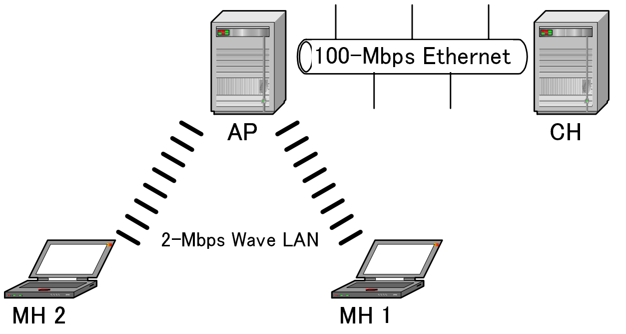

Figure 1. Experimental environment

3.2. Basic Observations

First, we observed how TCP throughput varied when SNR changed. We modified the average value of SNR at MH1 by changing the position of MH1 to AP. Each time the position of MH1 was changed, TCP data was continuously transmitted from CH to MH1 for three minutes. The average TCP throughput was measured at CH using Network Performance Benchmark (Netperf)[10], whereas the average value of SNR sampled every 3 seconds was calculated at MH1.

Figure2 shows the obtained results from 90 times of measurements. As seen in the figure, TCP throughput begins to decrease when the average SNR becomes less than 10 dB, and TCP throughput is almost zero when the average SNR is below 3 dB. It should be noted that

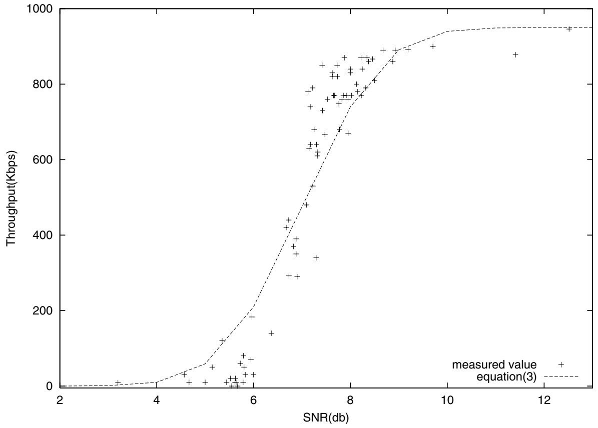

Figure 2. Relationship between SNR and TCP throughput

there can be a curve for approximation to express the correlation between the TCP throughput and SNR.

Next we observed the dynamic behavior of SNR at MH1 fixing its position. Figure 3 shows a snapshot of SNR values sampled at 3-s interval for 40 minutes. In this snapshot, the average value of SNR is approximately 8 dB.

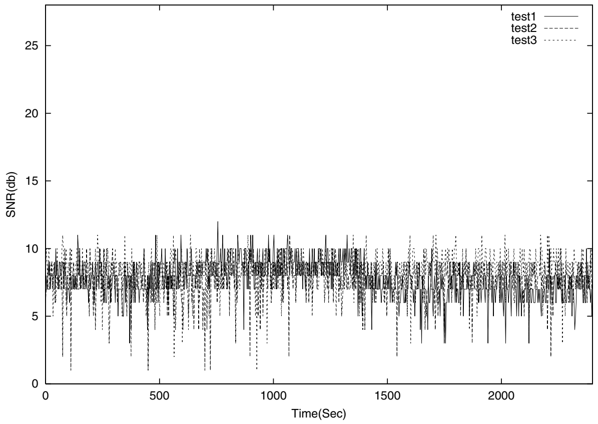

Figure 3. Fluctuation of SNR

Although the average value of SNR remains stable, SNR varies from 1 to 11 dB. Thus we can notice a considerable fluctuation in the level of SNR. We model this fluctuation with Gaussian distribution function. Assuming that the level of sampled SNR is Gaussian distributed, calculated probability density functions for 8-dB average SNR are shown in Figure 4. Furthermore, if we assume that degradation in throughput is proportional to the probability that SNR is below a certain level x.

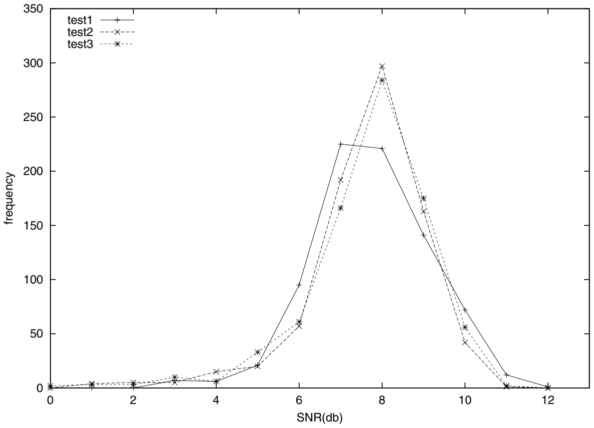

Figure 4. p.d.f. of SNR

Let x be 5 dB. When the average value of SNR is SNR, the expected throughput T due to the degradation is as follows.

$$T = T_{max} \frac{1}{\sqrt{2x}\sigma} \int_{-\infty}^{x} e^{-\frac{(y-\mathrm{SNR})^2}{2\sigma^2}} dy \qquad (3)$$

where T_{max} and σ are the maximum TCP throughput and the standard deviation, respectively. Assuming that σ does not change, the calculated T is shown in as the dashed line in Figure 2. The key observation here is not exact identification of the relationship between SNR and the throughput, but is that we can model the intermediate region between good and bad SNR area.

4. Design of SPS

We discussed the relationship between SNR and throughput in the previous section. In this section, we describe a new packet-scheduling scheme for wireless links based on receiver's SNR. The scheme is referred to as SNR-based Packet Scheduling (SPS). In SPS, flows are classified based only on the destination hosts.

4.1. The Assumed Environment

Figure 5 shows the assumed network configurations for SPS, which consist of wireless hosts and base stations. The base stations have a function of bridging between wired and wireless networks. Each mobile host is able to access WAN or LAN via the base station. SPS deals with communication over last-hop wireless links. More specifically, SPS scheduler is implemented in the base station. Within a cell that one base station covers, the strength of wireless links between the base station and mobile hosts varies depending on the mobile hosts and other electromagnetical interferences.

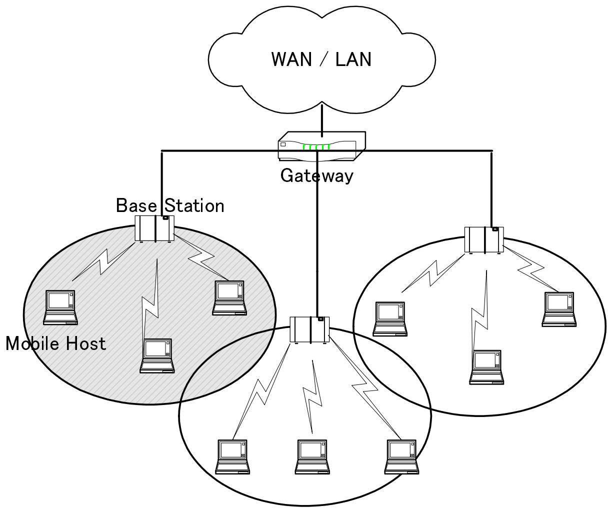

Figure 5. Assumed environment

4.2. Overview of SPS

The SPS in the base station consists of two components: SPS Monitor and SPS Forwarder. SPS Monitor identifies per-host SNR by being periodically notified of information about SNR from the hosts and predicting the current per-host SNR based on the notified information. In contrast, SPS Forwarder determines allocation of weight to each flow based on the current per-host SNR calculated at the SPS Monitor. The allocated bandwidth is equivalent to the link bandwidth multiplied by the allocated weight. It also perform packet scheduling at the output of the multiple queues in the send buffer based on the determined bandwidth. Figure 6 illustrates the structure of SPS.

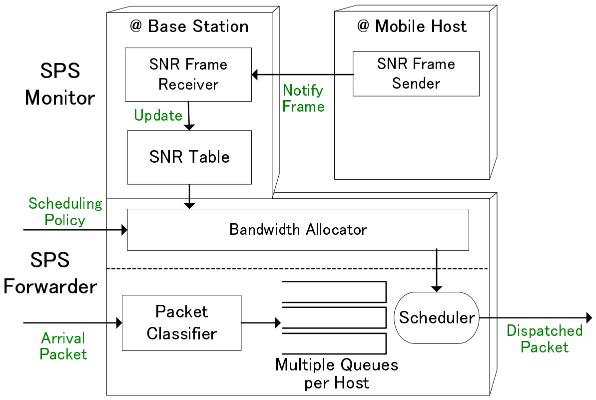

Figure 6. Components of SPS

SPS Monitor

SPS Monitor is responsible for monitoring the SNR of each mobile host. The base station usually cannot obtain SNR of mobile hosts unless each host notifies its SNR to the base station. Therefore, each mobile host periodically notifies the base station of its SNR over SNR frames. As a result, SNR Packet Receiver at the base station receives SNR information of each mobile host, and updates SNR Table using the received information. The SNR table maintains a pair of unique ID and SNR of each mobile hosts.

SPS Forwarder

SPS Forwarder maintains per-host separate queues of packets. Bandwidth Allocator in SPS Forwarder allocates bandwidth to each per-host queue based on a given queuing discipline and the SNR table. When the base station receives a packet, Packet Classifier in the SPS Forwarder determines which queue the packet is enqueued. Within each queue, packets are served in an FIFO order. However, service across different queues depends on a specific policy employed within the packet scheduler.

The Packet scheduler selects a queue for dequeuing depending on the weight by Bandwidth Allocator. Bandwidth Allocator periodically obtains SNR for active queues and changes their weights.

The pseudo-code of the SPS is shown in Figure 7.

4.3. Packet Scheduling Policies

SPS accommodates a flexible mapping between weight and SNR, unlike conventional schemes. Here we propose two schemes: Good or Bad (G/B) and piecewise linear schemes. For further explanations, we introduce some definitions: traffic capacity, C, level of channel condition, l, and weight, w.

- Let the traffic capacity for a wireless link be 100 units.
- The level of channel condition varies between 0 and 10. The value 0 corresponds to the worst state.
- The weight is set to a value between 0 and 1.

Let us assume that there are four hosts: Hosts A-D. If the scheduling discipline is FIFO, the C for the four hosts are as follows.

Destination Host	C [unit]
Host A	25
Host B	25
Host C	25
Host D	25

SPS Notations:

```
F(i)           : The packet virtual finish time
                 in session i.
SF(i)          : The session virtual finish time
                 in session i.
active_r       : Sum of all active session weight.
weight         : weight of the session.
L              : Length of the packet.
s_w(s,flag)    : Apply flag scheme and return weight
flag           : A scheduling policy. flag{G/B,
                 Piece-wise linear etc.}
snr(dest)      : Search newest SNR of destination.
snr_prev(dest) : Search previous SNR of destination.
```

Enqueuing:

```
enqueue(packet P, i)
   if not active(i)
      activate(i)
      active_r += r(i)
   if queue(i) is empty
      F(i) = SF(i) = max(F(i), V(t)) + L / weight
   else
     SF(i) += L / weight
   put(P, queue(i))
```

Dequeuing:

```
dequeue()
   i = min(active queues F(i))
   P = get(queue(i))
   t += L / r
   if active(i)
      F(i) += Lnext / r(i)
      for ever
         j = min(active queues SF(j))
         tmp_t = prev_t + (SF(j) - V(t)) *
         active_r / r
      if tmp_t > t
         V(t) += (t - prev_t) * r /
         active_r
         prev_t = t
         return P
      V(t) = SF(j)
      deactivate(j)
      active_r -= r(j)
```

Calculate queue weight:

```
r(i)
   s = snr(destination of session i);
   s = (a * s) + ((1 - a)(s_prev(destination
   of i)))
      weight = s_w(s, flag)
      return (weight)
```

Figure 7. Pseudocode for neighbor-state based queuing

SPS G/B scheme

The SPS-GB has one threshhold D. If SNR is more than D, w of the mobile host is 1, otherwise w is 0. This algorithm is as flows.

```
if (l < D)
   w = 0
elseif (l >= D)
   w = 1
else
   abort();
```

Figure 8 shows the relationship between SNR and weight in SPS-GB when D is 5. An example of bandwidth assignment shown in Table 1.

SPS Piece-wise linear scheme

The SPS-PW has several thresholds. Let D^{i} (i=1, ... , n) be the thresholds, where $Di < Dj$ ($i < j$). In SPS-PW, w increases slope, mi of linearly in every [Di, Dj]. This algorithm is as follows,

```
if (l < D1)
   w = 0
elseif (D1 <= l < D2)
   w = m1(l - D1)
.....

elseif (D[n-1] <= l < Dn)
   w = m[n-1] (l - D[n-1])
elseif (l >= Dn)
   w = 1
else
   abort()
```

This scheme models the relationship between l and w more accurately than SPS-GB or a linear curve. Figure 9 shows relation between SNR and weight that $D1 = 2$, $D2 = 4$, $D3 = 6$, and $D4 = 8$. An example of bandwidth assignment in Table 1.

5. Implementation

In this section, we explain the implementation of device driver extension and each function for SPS. We have implemented SPS Monitor at mobile hosts and the base station, and SPS Forwarder at the base station. FreeBSD 3.4-RELEASE with PAO Wave LAN driver is used for the implementation.

5.1. Device Driver Extension

Two kinds of extensions have been made to the device driver: extensions to Bridge and ALTQ.

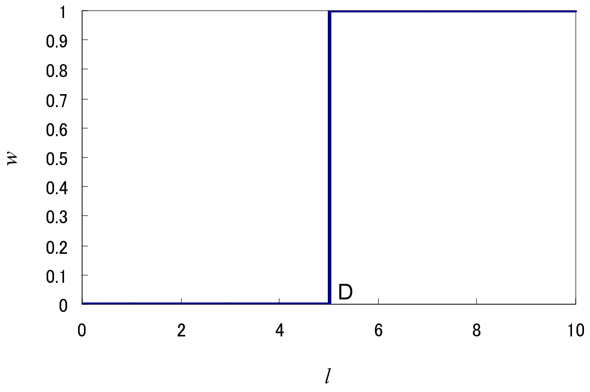

Figure 8. Relationship between SNR and weight in G/B scheme

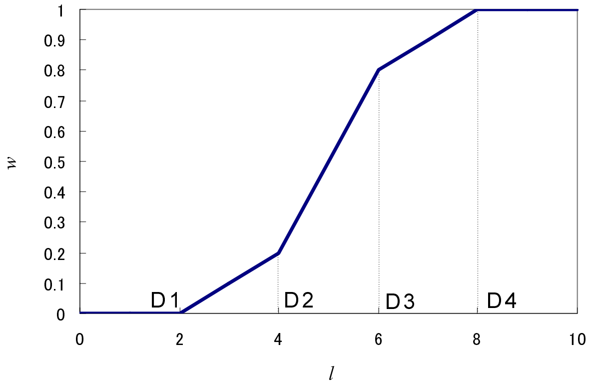

Figure 9. Relationship between SNR and weight in Piece-wise Linear scheme

Extension to Bridge

We supplemented Wave LAN driver "wlp" with bridge code, because Wave LAN driver for FreeBSD 3.4-RELEASE does not provide bridge mechanism. To put it more concretely, we wrote code that forwards outgoing packets to another interface card if a destination Ethernet address is not local before `ether_input` routine.

Extension to ALTQ

To facilitate packet scheduling, we have applied ALTQ [2]. ALTQ provides queuing schemes required to realize resource-sharing and quality of service. Although ALTQ supports many kinds of network drivers, ALTQ does not support "wlp". Hence we have extended the "wlp" driver for ALTQ support.

Table 1. Example of assignment with G/B scheme

Destination Host	l	w	C [unit]
Host A	2	0	0
Host B	4	0	0
Host C	6	1	50
Host D	8	1	50

Table 2. Example of assignment with Piece-wise scheme

Destination Host	l	w	C [unit]
Host A	1	0	0
Host B	3	0.1	10
Host C	5	0.5	40
Host D	9	1	50

5.2. SNR Frame Sender

SNR Frame Sender is implemented at the mobile wireless host. SNR Frame Sender is a module that sends SNR notification to the base station periodically. SNR Frame Sender has three functions. First, SNR Frame Sender obtains SNR for notification to the base station. We obtained SNR which was supplied by a register on the WaveLAN card. Second, SNR Frame Sender creates SNR frame. This frame contains the IP address of the mobile host because ALTQ maintains nodes by IP addresses. Figure 10 shows the `ether_snr` structure that has definition for SNR frame. Figure 11 shows frame format of SNR frame. Third, SNR frame sender sends packets to the base station at a regular interval.

```
struct ether_snr {
u_long
snr_spa; /* sender protocol address */
u_int32_t snr;
/* Signal-to-Noise Ratio */
}
```

Figure 10. ether_snr structure

5.3. SNR Frame Receiver

SNR Frame Receiver is implemented at the base station. SNR Frame Receiver receives SNR frame from

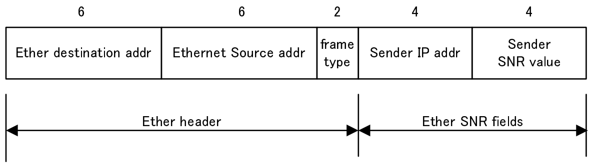

Figure 11. SNR frame format

```
struct snrentry {
u_long snr_spa;
u_int32_t snr;
struct snrentry *st_parent;
};
```

Figure 13. snrentry structure

mobile hosts. To distinguish SNR frames from other Ethernet frames, we created a new ether_type ETHERTYPE_SNR, and we supplemented the 'case' code to */sys/net/if_ethersubr.c* shown in Figure12.

</sys/net/if_ethersubr.c>

```
case ETHERTYPE_SNR:
schednetisr(NETISR_SNR);
inq = &snrintrq;
break;
```

Figure 12. Discrimination of ETHERTYPE_SNR

SNR Frame Receiver updates the SNR Table after receiving SNR frame. First, SNR Frame Receiver obtains the IP address of the mobile host from the frame. Next, SNR Frame Receiver check whether there exists an entry of the `snrentry` structure shown in Figure 13 in the SNR Table. If the entry already exists in the structure, SNR Frame Receiver updates the `snrrentry` structure's snr based on the received SNR frame. However, if the entry does not exist in the `snrentry` structure, SNR Frame Receiver creates a new entry in the `snrentry` structure.

5.4. SNR Table

SNR Table maintains information obtained from SNR frame for a flexible mapping between weight and SNR. After the base station analyzes a received packet, it updates or adds a new entry to the SNR table. Thus the base station maintains the newest information about the combination of SNR Table SNR value and IP address of mobile hosts.

5.5. SPS Forwarder

We explain the implementation of SPS Forwarder that performs bandwidth allocation in base station. SPS uses alternate queuing framework to apply to a lot of queuing algorithm based on SNR. Currently ALTQ supports Class Based Queuing (CBQ) [5], Hierarchical Fair Service Curve(HFSC) [14], Weighted Fair Queuing(WFQ) [3] and Random Early Detection(RED) [4] etc. As another queuing scheme, we added SPS queuing scheme in the ALTQ.

Figure 14 shows the SPS Forwarder processing. First, the `snrlookup` routine reads the SNR Table and obtains the pair of SNR and the destination address of active queues. It decide weight of each queue from SNR according to formula that has relationship between SNR and weight. Second, the `create_config` routine creates the configuration file based on decided weight. Third, the `qcmd_init` routine enables filters for all the interfaces by reading the configuration file and setting up interfaces. the `qcmd_init` routine is providing libaltq.

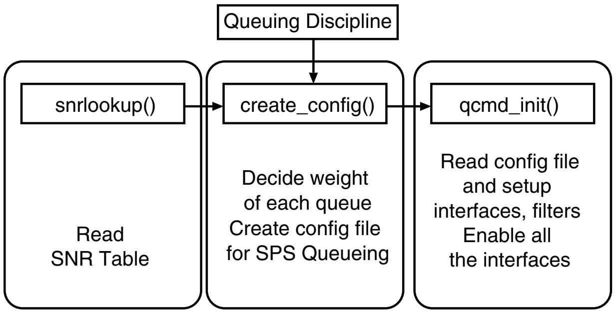

Figure 14. SPS Forwarder processing

6. Evaluation Results

This section presents the results of the experiments with SPS. The experimental environment is identical to the environment shown in Figure 5. In this section we use two mobile hosts (MH1 and MH2). We performed the experiment where CH sends Constant Bit Rate (CBR) UDP flows to MH1 and MH2. Transmission data amounts to 14.7 Mbyte, and a packet size is 1470 byte at each experiment, the SNR is sampled at 3-5 seconds interval. In the experiment, the bandwidth of wireless link is 2 Mbps.

6.1. Comparison between FIFO and Bandwidth-Controlled Scheduling

First, as a preliminary study, we conducted ten trials of the two following situations.

- Case 1: measurement of the throughput from CH to MH1 in good condition.
- Case 2: measurement of the throughput from CH to MH1 and MH2 at the same time in a good condition.

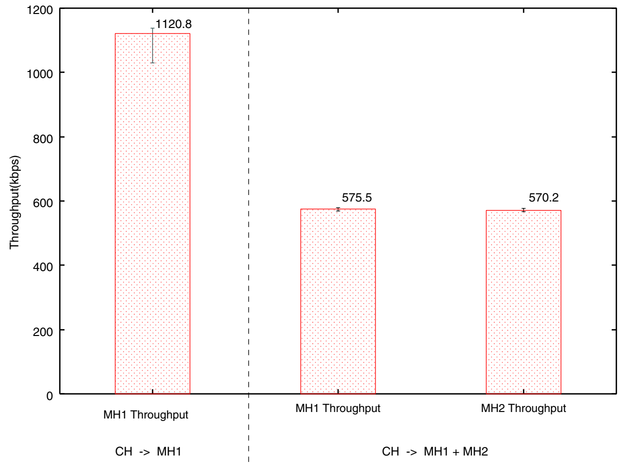

Figure 15. The throughput of the UDP flow while both channels have good condition

In both measurements, CH sends UDP data continuously at the maximum rate over 2-Mbps WaveLAN. The bar of "CH -> MH1" in Figure 15 shows the average value of the throughput of Case 1 together with the supplementary maximum and minimum values when the average SNR was approximately 24.5 dB. On the other hand, the bar of "CH -> MH1 + MH2" indicates the average value of the throughput of Case 2 when the average SNR was approximately 24.5 dB. Note that the sum of the throughput in Case 2 is almost equal to the throughput in Case 1.

Next, we compare the throughput when a bandwidth-based packet scheduling is performed with that when an FIFO packet scheduling is at a base station. We conducted ten trials of the two following situations.

- Case 3: measurement of the throughput from CH to MH1(good condition) and MH2(bad condition) with no bandwidth control.
- Case 4: measurement of the throughput from CH to MH1(good condition) and MH2(bad condition) with bandwidth control.

In both experiments, the average values of SNR were observed to be approximately 24.5 dB at MH1 and 5.9 dB at MH2. For bandwidth controlling, we applied CBQ [5] contained in ALTQ [2] software to a packet scheduling in the base station. We set weight of MH1 is 1 and MH2 is 0.15 statically in Case 4.

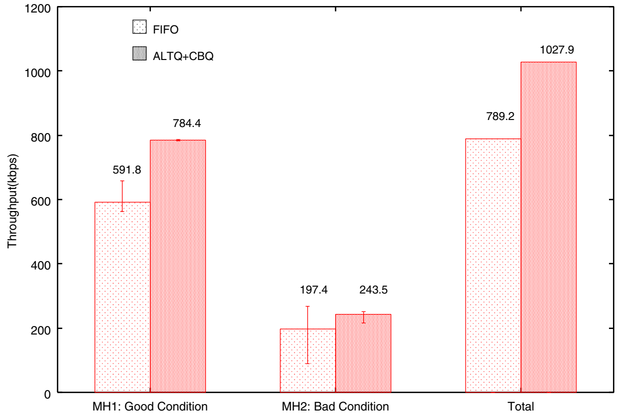

Figure 16. The comparison of the UDP Throughput between FIFO and CBQ

Figure 16 shows the results of the both experiments. As seen in the figure, the throughput from CH to MH1 in Case 3 is almost the same as that in Case 2. However, the throughput from CH to MH2 in Case 3 is worsened. Also the throughput is also deteriorated in Case 4, the amount of the deterioration is mitigated. Furthermore, the throughput from CH to MH1 is improved in Case 4 compared with Case 3. As a result, the total throughput with bandwidth control by CBQ becomes 23 % larger than that without bandwidth control. An application with bandwidth control is also useful to reduce the fluctuation of the throughput as seen in Figure 16.

It is also seen that the total throughput in Case 4 is only 90 % of that in Case 2. This is due to the overhead incurred by the bandwidth-controlled packet scheduling. But it can be said that the bandwidth-controlled packet scheduling is effective when there is a bad wireless environment despite of its overhead.

6.2. Comparison between G/B scheme and Piece-wise linear scheme

We compared the performance of SPS-GB and that of SPS-PW. Figure 17 and Figure 18 show snapshots of transmission rate of UDP flows from CH to MH1 and MH2 at AP, respectively. AP transmits the equal share or the maximum and the minimum share to MH1 and MH2 in Figure 17. As seen in the figures, SPS-PW provides finer tuning of bandwidth allocation.

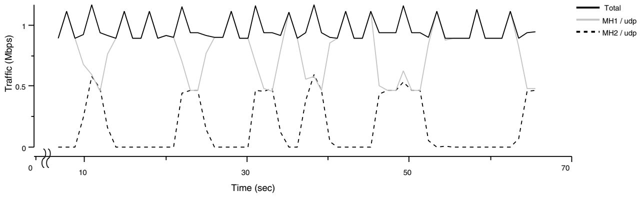

Figure 17. Snapshot of SPS-GB

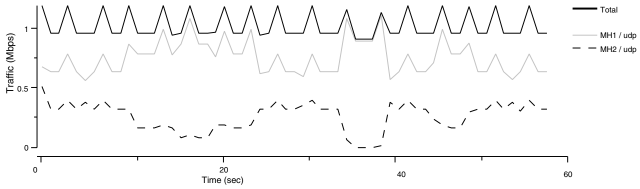

Figure 18. Snapshot of SPS-PW

We conducted ten trials of the 5 following situations.

- Case 5: measurement of the throughput from CH to MH1(good condition) and MH2(bad condition) with no bandwidth control.
- Case 6: measurement of the throughput from CH to MH1(good condition) and MH2(bad condition) with SPS-GB (GB1: $D = 3$) .
- Case 7: measurement of the throughput from CH to MH1(good condition) and MH2(bad condition) with SPS-GB (GB2: $D = 6$) .
- Case 8: measurement of the throughput from CH to MH1(good condition) and MH2(bad condition) with SPS-GB (GB3: $D = 9$) .
- Case 9: measurement of the throughput from CH to MH1(good condition) and MH2(bad condition) with SPS-PW (PW)

where D is threshold of weight to change.

In all the above these measurements, the average values of SNR were observed to be approximately 25 dB at MH1 and 6.3 dB at MH2. Case 5 is set to identical to Case 3 for reference. In Case 6-8, we apply SPS-GB. GB1-3 have a different threshold each other. Also in Case 9, we apply SPS-PW. PW has 4 thresholds, $D1 = 3, D2 = 5, D3 = 7, D4 = 9$. We set these value of threshold based on the result of the experiments.

The results of the these experiments are shown in Figure 19 and 20. Figure 19 shows the average values of throughput of FIFO, GB1, GB2, GB3 and PW with the supplementary maximum and minimum values. Figure 20 shows the total throughput of each scheme and the bandwidth consumed at MH1 and MH2. The throughput fluctuate compared with Case 3 because these schemes are adapting dynamical to wireless conditions.

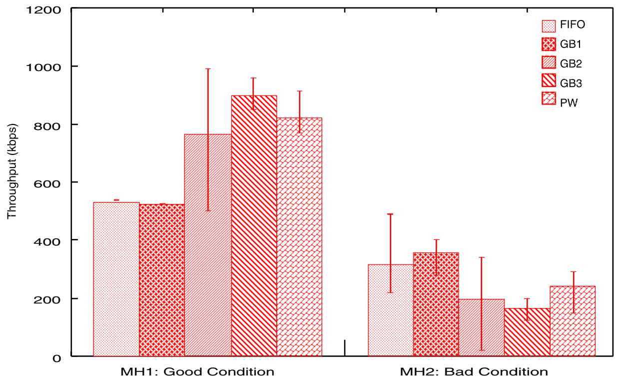

Figure 19. Throughput of UDP flow and dispersion at MH1 and MH2

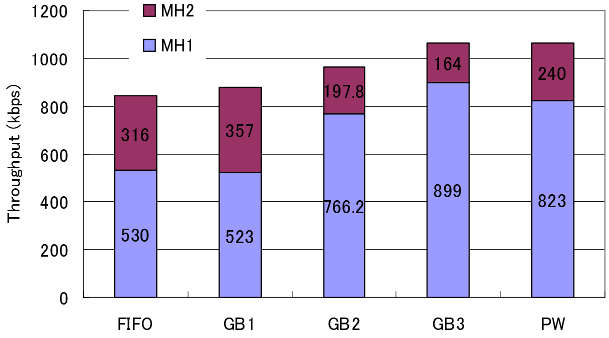

Figure 20. Total throughput of each scheme and detail

Let l_{eq} denote the value of l when w equals to 0.5 . We set l_{eq} of GB1 to lower than those of GB2 and PW. For this reason, GB1 does not limit the bandwidth in the case of the SNR is somewhat a low value. As a result, GB1 achieves the same performance of FIFO in Figure 20.

We set l_{eq} of GB2 identical to PW. However, the throughput of GB2 fluctuates more compared with PW. Therefore GB2 does not provides fairness to between MH1 and MH2. Furthermore GB2 does not achieve the total throughput that PW attains.

GB3 has the almost same performance as PW. However, GB3 reduces the bandwidth of MH2 because we set l_{eq} of GB3 to higher than those of GB2 and PW. If all the packets to the mobile host in a bad radio condition are dropped, we can improve the total throughput. Instead, the bandwidth to these hosts are overly reduced.

Each G/B scheme differ in the performance depending on the threshold. The fluctuation of the throughput is mitigated in PW compared with G/B schemes because PW applies fine-grained mapping between the throughput and the fluctuation of SNR by conducting dynamic adaptation. Moreover we can expect that PW is useful in improving throughput when the mobile host moves. Finally, from the discussion above, we can conclude that PW archives the best balance of performance and fairness in the five schemes.

7. Conclusion and Future Work

In this paper we have presented a new packet scheduling scheme for wireless links based on receiver's SNR, SPS. In SPS, outgoing packet flows are scheduled such that the flows with higher SNR values have larger weights in scheduling. Unlike conventional schemes, SPS accommodates a flexible mapping between weight and SNR. We also presented measured results of the relationship between throughput and SNR, the design and the implementation of SPS. In our experiments, SPS has shown its effectiveness in enhancing total throughput for hosts with good and bad wireless connectivity over a wireless LAN. Our experimental results over WaveLAN demonstrate that bandwidth control by CBQ outperforms a scheme without bandwidth control by 23 %. Thus we can conclude that PW archives the best balance of performance and fairness in the five schemes.

We plan to implement a packet scheduling scheme including the importance-value scheme. We have not addressed how a flexible scheduling is performed utilizing information about SNR. The SPS scheduler can allocate bandwidth to each flow based on per-flow importance values as well as SNR. The per-flow importance value can be assigned such that a flow requiring higher real-time delivery such as an audio-data flow has a higher value. The SNR value of the k-th host, $S(k)$ is modulated to $S^*(k)$ as follows:

$$S^*(k) = (1-a)S(k) + aI(k),$$

where $I(k)$ and a represent the importance value for the k-th host and a coefficient of balancing, respectively. We plan to implement SPS taking the importance value into account.

As another future work, the relationship between the frequency of feedback and the performance should be analyzed. We must consider how wireless hosts move for this analysis.

Acknowledgments

We would like to thank the members of RT-HDI projects for their advice. We would also like to thank Soko Aoki, Kaori Ishii and the anonymous reviewers for valuable comments.

References

[1] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. Tripathi, "Enhancing throughput over wireless LANs using channel state dependent packet scheduling," *IEEE INFOCOM '96,* Apr 1996.

[2] K. Cho, "A framework for alternate queuing: Towards traffic management by PC-UNIX based routers," *USENIX 1998 Annual Technical Conference,* June 1998.

[3] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queuing algorithm," *ACM SIGCOMM'89,* 1989

[4] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transaction on Networking,* 1(4):pages 397-413, Aug 1993.

[5] S. Floyd, and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Transactions on Networking,* 3(4): pages 365-386, Aug 1995.

[6] J. Gomez, Andrew T. Campbell, and H. Morikawa, "The Havana framework for supporting application and channel dependent QoS in wireless networks," ICNP '99, 1999.

[7] IEEE 802.11, "International standard for Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," 1999.

[8] S. Lu, T. Nandagopal, and V. Bharghavan, "Fair scheduling in wireless packet networks," *ACM SIGCOMM'97,* Aug 1997.

[9] T. Nandagopal, S. Lu, and V. Bharghavan, "Unified architecture for the design and evaluation of wireless fair queuing algorithms," *ACM MOBICOM'99,* Oct 1999.

[10] Network performance benchmark, Information Network Division Hewlett-Packard Company, Jan 2000, http://www.netperf.org/

[11] T. S. Ng, I. Stoica, and H. Zhang, "Packet fair queuing algorithms for wireless networks with location dependent errors," *IEEE INFOCOM'98,* March 1998.

[12] P. Ramanathan and P. Agrawal, "Adapting packet fair queuing algorithms to wireless networks," *ACM MOBICOM'98,* Oct 1998.

[13] M. Srivastava, C. Fragouli, and V. Sivaraman, "Controlled multimedia wireless link sharing via enhanced class-based queuing with channel-state-dependent packet scheduling," *IEEE INFOCOM'98,* March 1998.

[14] I. Stoica, H. Zhang, and T. S. Eugene Ng, "A hierarchical fair service curve algorithm for link-sharing, real-time and priority service," *ACM SIGCOMM'97,* 1997.

[15] B. Tuch, "Development of WaveLAN, an ISM band wireless LAN," *AT&T Technical Journal,* 72(4): pages 72-33, July/Aug. 1993.