무선 이동 네트워크에서의 적응적 자원 할당 방법

(Adaptive Resource Allocation Schemes in Wireless Mobile Networks)

강유화 [†] 서영주 [††] 안성옥 [†††]
(Yoo-Hwa Kang) (Young-Joo Suh) (Syungog An)

요 약 무선 네트워크 환경에서는 네트워크의 제한된 대역폭과 이동 호스트의 잦은 셀 간 이동으로 인하여, 각 호스트가 요구하는 서비스의 질(Quality of Service)을 보장하기 힘들다. 이런 무선 통신의 특성에서도, 실시간 서비스와 같은 특정 종류의 서비스의 경우에는 요구되어지는 범위 내에서 그 서비스의 질이 반드시 보장되어야 한다. 본 논문에서는 무선 이동 네트워크 환경에서 각 서비스의 서비스 율을 어떻게 조정할 것인가에 대해 논의한다. 무선 네트워크 환경에서 이동 호스트가 다른 셀로 이동하였을 경우, 부족한 대역폭을 고려하여 호스트의 서비스 율이 조정된다. 본 논문에서 각 플로우의 QoS Spec을 고려하여 서비스 율을 할당하는 알고리즘을 제안한다. 제안된 알고리즘에서는 각 플로우가 요구하는 QoS spec 값에 비례하도록 서비스 율을 정하여 부족한 네트워크 자원을 동적으로 할당함으로써, 무선 네트워크 자원을 보다 효율적으로 사용할 수 있게 한다.

Abstract In wireless networking environments, supporting guaranteed quality of service to mobile hosts is difficult due to the facts that wireless networks have limited bandwidth and mobile hosts frequently move in and out of cells. In spite of the characteristics of wireless communications, the quality of some types of services, i.e., real-time services, must be guaranteed at a certain level. When a mobile host moves into another cell, service rates for mobile hosts in wireless networks may be adjusted since wireless networks have limited bandwidths. In this paper, we propose two resource allocation algorithms in wireless mobile networks, using quality of service (QoS) specifications. For efficient use of resources of wireless networks, the proposed algorithms dynamically allocate rates of flows in proportion to QoS with limited resources.

1. Introduction

In mobile computing environments, an important issue is how to support the guaranteed quality of different types of services. Real-time service requires a bound on the delivery delay of each packet, which must be known a priori. To support real-time traffic, a certain quality of services(QoS) must be guaranteed[1,8,10,12]. In general, real-time services are separated into two types[1,8,12]. One is *guaranteed service* whose delay does not exceed a fixed bound, and the other is *predicted service* whose delay is adaptively bounded. Guaranteed service involves pre-computed worst-case delay bounds, and predicted service uses the measured performance of the network for computing delay bounds. We focus on predicted service in this paper. In addition to real-time service, *best effort service* can be delivered in the network. However, we do not consider best effort service in wireless networks because it does not require any QoS spec on the delivery delay.

In wireless networks bandwidth is limited and many mobile hosts may move in and out of cells;

---

† 정 회 원 : 한국전자통신연구원 네트워크연구실 연구원
yhkang@etri.re.kr
†† 종신회원 : 포항공과대학교 컴퓨터공학과 교수
yjsuh@postech.ac.kr
††† 종신회원 : 배재대학교 컴퓨터전자정보공학부 교수
sungohk@mail.paichai.ac.kr

논문접수 : 1999년 11월 26일
심사완료 : 2001년 8월 2일

consequently, we have to efficiently use network resources in order to provide an acceptable QoS for wireless communications[11,13]. To maximize the utilization of networks, their bandwidths are dynamically distributed to the services within the networks[8,12,14]. That is, the rates of services must be adjusted because of the limited bandwidths of wireless networks. We propose new rate allocation algorithms for wireless mobile networks by using QoS specification. Our algorithms are efficient and fair to all services according to their QoS specifications. In this scheme, we select a sufficient number of flows and change their rates considering network utilization. Therefore, our algorithms provide fair service in proportion to QoS specs, and it does not cause additional network overhead.

In Section 2 we begin with a discussion of the fairness problem to support real-time services with different QoS specs in wireless networks, and then in Section 3 we review previous published work. In Section 4, we propose our algorithms to adapt the rates of services in wireless networks, compare proposed algorithms to previous schemes, and discuss the improvements of our algorithms. Then, in Section 5 we provide simulation experiments and compare the performance of them. Finally, we conclude our paper with a review of our results and a brief discussion of future work in Section 6.

2. QoS-Based Fairness Problem

In this section, we describe the network model and then discuss the rate allocation algorithm problem to support fair services according to their different QoS in wireless mobile networks.

2.1 Network Model

We consider a cellular network architecture that is made up of cells. Each cell is covered by a base station, which is connected to the wired backbone network. The base station provides wireless network access to the mobile hosts in that cell. The bandwidth of a wired link in the backbone network is much higher than that of a wireless link. Accordingly, we assume that when a large number of mobile hosts enter into a single cell only the wireless link of the cell can become overloaded because the wired links have enough capacity to satisfy any arbitrary bandwidth requirement. We also assume that the base stations have sufficient computing power, and therefore it is more important to reduce network overhead at the expense of computations at the base stations.

2.2 Rate Allocation Algorithm Criteria

When the requirements of all the services within a link in a network are more than its capacity, a sufficient number of services are dropped or adapted because many services are adaptive in nature and operate over a wide range of bandwidths. It is assumed that each application has an associated discrete set of throughput levels and that they can operate at any of those bandwidths; however, a higher throughput level will have a better QoS. Adaptive service provides QoS within bounds, e.g., minimum bandwidth and maximum bandwidth, which means that flows can be accepted so long as their lower bounds can be satisfied. For such services it is possible to overcome the link overload condition by reducing the bandwidth of individual flows, which is called *rate adaptation*. To increase network utilization and provide better service, excess resources need to be distributed effectively among flows. Two important criteria of rate adaptation are *fairness* and *network overhead*[14]. Fairness is related to which flows are affected and how they are affected, whereas network overhead is related to the amount of messages that inform senders and receivers of their rates. There is tradeoff between fairness and network overhead. To support better fairness of flows, the number of flows whose rates are changed must increase and the network overhead must also increase.

During a handoff, the rates of flows in the link have to be allocated fairly according to their QoS specs. It cannot be regarded as fair to allocate the same rates to flows that have different QoS specs. We consider another important criterion: *QoS specification* in rate adaptation. To support fairness in rate adaptation, we have to allocate proper rates to flows according to their different QoS.

In this paper, we use following notations:

- QoS spec – $[B^i_m, B^i_x]$:
  minimum bandwidth and maximum bandwidth for flow i in the network
- B^i_c : current bandwidth of flow i in the network

We characterize the fairness property of a rate allocation scheme using the following parameters.

- *Mean Service Ratio*: Mean of var (B^i_c/B^i_x)
  for flow i over all flows in the network
- *Variance of Service Ratio*: Variance of var (B^i_c/B^i_x)
  for flow i over all flows in the network
  where var (B^i_c/B^i_x) is the variance of the normalized bandwidth of flow i over its duration

Service ratio is defined by the ratio of the current bandwidth used by a host over the maximum bandwidth required by a host. For each flow, the variance of service ratio during the total duration is calculated. The Mean Service Ratio is the average value of variances of service ratios for all flows. Variance of Service Ratio is the variance of service ratio variances of all flows, and small value of Variance of Service Ratio indicates that the network offers relatively balanced service rates to all flows.

We determine the network overhead of a rate allocation scheme using the following two parameters.

- *Average Time Counts*: average number of flows whose rates are changed per second
- *Average Handoff Counts*: average number of flows whose rates are changed per handoff

3. Previous Work

In a fixed network, a *max-min optimality* criterion is used for rate allocation[2]; the total bandwidth of each of these links is shared equally among all flows passing through that link.

In a wireless mobile network, Lu[8] used a max-min optimality criterion as the dynamic rate allocation algorithm when mobile users move from one cell to another. In this scheme, which is invoked frequently in mobile networks, the available bandwidth is fairly allocated to flows every time a handoff occurs or a flow terminates. Although fairness is ensured at all times among the flows within a cell with this scheme, the network overhead may be very high.

Talukdar[14] characterizes the fairness property of a rate allocation scheme in terms of its impact on the *aggregate quality* received by a flow in a wireless mobile network. Three rate adaptation schemes are described in[14]: *Minimum Adaptation*, *Fair Adaptation* and *Average-Fair Adaptation*. The Average-Fair Adaptation scheme considers the aggregate quality of a flow to be the overall quality of service it has received over its duration. This scheme allocates the minimum QoS requirement or the maximum QoS requirement to a small number of flows in rate adaptation. Therefore, it does not support fairness among all flows with different QoS at all times.

Each algorithm proposed in[14] uses two quantities: *availability* and *demand*. 'Availability' is the maximum amount of bandwidth the flow can release without being dropped, that is, the difference between its current bandwidth and its minimum bandwidth. 'Demand' is the maximum amount of additional bandwidth the flow may request to improve its QoS, that is, the difference between its current bandwidth and its maximum bandwidth. Each link maintains two data structures: *avail_list* and *degraded_list*. In the avail_list, the flows in the link are kept sorted in the non-increasing order of their availability. In the degraded_list, the flows operating below their maximum bandwidth levels are kept sorted in the non-increasing order of their demands.

3.1 Minimum Adaptation (MA)

When an existing flow enters into a cell from another, the Minimum Adaptation scheme drops, allocates, or degrades the entering flow. If the total bandwidth requirements of all flows including the newly entered flow may exceed the link capacity and the total minimum bandwidth requirements of them can be provided by the link, the *degrade algorithm* is used. To degrade the rates of flows, the degrade algorithm first computes the bandwidth deficit, and then the flows in the avail_list are reduced to their minimum bandwidth levels in the non-increasing order of their availability until there is no bandwidth deficit.

When a flow moves out of a cell or terminates, the *upgrade algorithm* checks if there are any flows operating below their maximum bandwidth levels. To upgrade the rates of flows, the upgrade algorithm first computes the excess available bandwidth of the link, and then it increases the bandwidth levels of the flows in the degraded_list as much as possible in the non-increasing order of their demands.

3.2 Fair Adaptation (FA)

The Fair Adaptation scheme tries to allocate each flow rate in a fair way during the handoff. When a flow enters a cell, this scheme drops, allocates or degrades the rates of all flows in the cell. To degrade, it first allocates the minimum bandwidth level to each flow, and then the excess available bandwidth of the link is equally distributed to all flows.

When a flow moves out of a cell or terminates, this scheme fairly upgrades the rates of all flows in the cell.

3.3 Average-Fair Adaptation (AFA)

The Average-Fair Adaptation scheme uses the average bandwidth of currently active flows in a cell to determine which flows should be upgraded or degraded when flows handoff or terminate. Avail_list and degraded_list are kept sorted on a composite key including *relative average bandwidth* as a primary key for each flow. Relative average bandwidth is the total bandwidth consumed by a host up to the present

This algorithm selects a sufficient number of flows from the avail_list, which is kept sorted in the non-increasing order of the flows' relative average bandwidths or degraded_list, which is kept sorted in the non-decreasing order of the flows' relative average bandwidth. This scheme either decreases the bandwidths of the selected flows until there is no deficit bandwidth, or it increases them to the maximum possible bandwidths.

Minimum Adaptation tries to minimize the number of flows whose rates are changed. Thus, this scheme unfairly allocates each flow rate during a handoff. However, its network overhead is low. Fair Adaptation creates higher network overhead than the minimum algorithm because it fairly distributes network bandwidth to all flows during handoff. Average-Fair Adaptation is similar to the minimum algorithm. It selects flows with larger or smaller average bandwidth first, so the overall bandwidth allocation to flows having a long duration will have better fairness characteristics than the minimum algorithm. In the average-fair algorithm, the number of flows whose rates are changed is small compared to that in the fair algorithm so network overhead is lower.

4. Rate Allocation Algorithms

Fair Adaptation distributes available bandwidth of the link to each flow equally; consequently, it cannot support various kinds of services that have different QoS specs. To distribute the available bandwidth to each flow by QoS specification, we allocate the rate of each flow in proportion to its QoS spec, which guarantees fair allocation by QoS specification at any moment.

To reduce network overhead, the number of changed flows must be as small as possible. Minimum Adaptation changes the rates of a small number of flows that are selected by their availability. A flow's availability is the absolute value of the maximum bandwidth that it can release (that is, |current bandwidth minimum bandwidth|). It is not the percentage of the flows' different bandwidth ranges. Accordingly, availability cannot be a key to allocate rates fairly to flows that have different QoS specs.

Average-Fair Adaptation changes the rate of each flow by using the relative average bandwidth of a flow, which guarantees fairness over a long period of time. We do not need to calculate the average bandwidth for a long time, however, because we fairly allocate flows rates in proportion to their QoS specs during handoff. To support fairness by using QoS specs, it is sufficient to compute the ratio of the current bandwidth to the maximum bandwidth. This ratio is the relative value of the QoS specs of each flow, so it guarantees fairness at all times among the flows over their duration even though flows have different QoS specs.

To better support fairness and reduce network overhead, we propose the *Average-Available Rate Allocation* algorithm and the *Target-Based Rate Allocation* algorithm. Our algorithms allocate rates in proportion to the QoS specs to sufficient flows in the list of the link using new selected keys.

In the rate adaptation schemes proposed in[14], flows have bandwidth requirements and levels of operation and they operate at any of the bandwidth levels. These schemes change the bandwidth levels of the flows in rate adaptation. In our scheme each mobile host has both a minimum and maximum bandwidth requirement, as a new QoS specs that is not composed of the bandwidth levels. Each flow can operate at any bandwidth between its minimum and maximum bandwidth level, which is allocated according to its QoS spec in rate adaptation. The maximum bandwidth level is required to reduce the overhead caused by increased number of rate adaptations. If the maximum bandwidth level is not defined, a mobile host in a cell can use the full bandwidth of the cell, and then each time other mobile hosts request bandwidths, rate adaptations should be performed, which increases overhead.

Notation

- B_t^i : target bandwidth of flow i in the network
- $A_c = C - \sum_i B_c^i$: available bandwidth of a link for flows running at current rates
- $A_m = C - \sum_i B_m^i$: available bandwidth of a link for flows running at minimum rates
- C: network capacity
- N: number of flows in the link
- avail_list: sorted in non-increasing order of availability for all flows within the link
- Availability is defined for each flow as

$$A^i = B_c^i - B_m^i, \text{ for flow i in the link}$$

4.1 Average-Available Rate Allocation (AAA)

In this scheme, each link has an *avail_list* in which flows in the link are kept sorted in the non-increasing order of their availability, which is the criterion for selecting flows for upgrading or degrading in rate adaptation. The availability is the difference between minimum bandwidth and current bandwidth in the condition that services have similar amounts of bounded delay; therefore, it supports better fairness by using QoS specs on all flows in the link at any time, regardless of their different QoS specs.

When a new flow i is created or flow i moves into a cell, this scheme first checks if the currently available bandwidth(A_c) of the link is enough to provide the minimum bandwidth requirement of flow i. If possible, flow i is allocated as much bandwidth as possible. Otherwise, this scheme checks if the minimum bandwidth requirements of all flows including flow i can be provided by the link. If they can, the rates of flows in the link are degraded; if not, flow i is dropped.

The *degrade algorithm* first computes the *average availability* of all flows including the newly entered flow in the link and then compares it to the availability in the avail_list. The number of flows whose availability is greater than the average availability of the link is approximated to be half of all the flows in the link. To reduce the network overhead, we find the average availability multiplied by 1.8. The value of 1.8 in degrade algorithm (also, the value of 0.2 in upgrade algorithm) is used for the following reason. If we assume that the probability of the availability of all flows served in the link is distributed uniformly from 0 to 1 and then about ten percent of all flows are supposed to have values higher than this. As we adapt the percentage of changed flows by multiplying the average availability by various values, we can control the network overhead in the wireless network. If the service ratio of a flow is greater than this value, then we put it in the *reduced_set* with its minimum bandwidth. Otherwise the rate of a flow is not changed. Flows whose availability is over this value are supposed to be running at high rates relative to other flows in the link We find the total availability (the available bandwidth except the total current bandwidths of flows not in the reduced_set) of the link and flows in the reduced_set, and then allocate each flow in the reduced_set a reduced rate in proportion to its maximum bandwidth. The degrade algorithm is shown in Figure 1.

When flow *i* moves out of a cell or terminates in a cell, we first check if there are any flows running at below their maximum bandwidth. If so, the *upgrade algorithm* is executed, otherwise it is not executed. In the upgrade algorithm, if the increased rate of a flow is higher than its maximum rate, we allocate its maximum rate instead of its adapted rate. The upgrade algorithm first checks the available bandwidth of the link in which all flows, except flow *i*, are running at their minimum rates. Then this algorithm computes the average availability of the link. To reduce the network overhead, we find twenty percent of the average availability. About ten percent of all flows are supposed to have values less than this. Then we compare the availability of each flow in the

```
$A_r = 0$;   $n = 0$;   $B_x = 0$;

// input a new flow $i$ into avail_list
$N = N + 1$;        // number of flows including new flow $i$
$A_m = A_m - B_m^i$;  // recaculate $A_m$
$A_a = A_m / N$;    // average availability of the link
$A_a = A_a * 1.8$;

// for flow $i$ whose availability is greater than $A_a$
while ($A^i > A_a$)     // $i \in$ flows in avail_list
// flow $i$ is put into reduced_set
{
    $n = n + 1$;        // number of flows in reduced_set
    $B_x = B_x + B_x^i$; // $\sum_i B_x^i$ ,for i $\in$ flows in reduced_set
}

// find the total availability of unchanged flows
// for $A^i \le A_a$ ,$i \in$ flows in avail_list
for (i = $n$ ; i < $N$ ; i++)        $A_r = A_r + A^i$;

// total availability of the link for changed flows
$A_t = A_m - A_r$;

// allocate rate to flows in reduced_set
for (i = 0 ; i < $n$ ; i++)
{
    $Rate = B_m^i + A_t * \frac{B_x^i}{B_x}$;   // $i \in$ flows in reduced_set
    if ($Rate > B_x^i$)         $B_c^i = B_x^i$;
    else                        $B_c^i = Rate$;
}
```

Fig. 1 Degrade Algorithm AAA

```
$A_u = 0$;   $n = 0$;   $B_x = 0$;

// remove a handout flow $i$ from avail_list
$N = N - 1$;        // number of flows except flow $i$ in the link
$A_m = A_m + B_m^i$;  // recaculate $A_m$
$A_a = A_m / N$;    // average availability of the link
$A_a = A_a * 0.2$

// find the total availability of unchanged flows
while ($A^i \ge A_a$)
{
    $A_u = A_u + A^i$;    // $i \in$ flows in avail_list
    $n = n + 1$;         // number of unchanged flows
}

// for flow $i$ whose availability is less than $A_a$
// flow $i$ is put into upgraded_set
for (i = $n$ ; i < $N$ ; i++)
    $B_x = B_x + B_x^i$;  // $\sum_i B_x^i$ ,for $i \in$ flows in upgraded_set

// total availability of the link for changed flows
$A_t = A_m - A_u$;

// allocate rates to flows in upgraded_set
for (i = 0 ; i < $N - n$ ; i++)
{
    $Rate = B_m^i + A_t * \frac{B_x^i}{B_x}$;   // $i \in$ flows in upgraded_set
    if ($Rate > B_x^i$)         $B_c^i = B_x^i$;
    else                        $B_c^i = Rate$;
}
```

Fig. 2 Upgrade Algorithm AAA

avail_list with this value. If the availability of each flow is less than this value, then we put a flow in the *upgraded_set* with its minimum bandwidth. Flows whose availability is greater than this value are supposed to run at relatively high rates within the available bandwidth of the link. Consequently, we exclude those flows from the upgraded_set and do not change their rates. We find the excess availability (the excess available bandwidth up to the current bandwidth of flow *i*) of the link and its flows in the upgraded_set and allocate a rate to each flow in the upgraded_set in proportion to its maximum bandwidth requirement. Figure 2 shows the upgrade algorithm.

4.2 Target-Based Rate Allocation (TBA)

In the above algorithm we change rates of flows in the reduced_set or upgraded_set based on their availability compared with the average availability of the link. If each flow has very different bandwidth range between the minimum bandwidth and the maximum bandwidth, the difference of the availability of each flow is very high. To compare the availability of flows, they need to have similar QoS specs because availability is the difference between the current bandwidth and the minimum bandwidth. When all services have similar amounts

```
$B_m = 0;$    $B_x = 0;$    $B_c = 0;$    $n = 0;$

// for flow $i \in$ flows in the link including a new flow
for ($i = 0, i < N, i++$)
{
// target bandwidth is the 70 percent of the maximum
    $B_t^i = 0.7 * B_x^i;$

// for flow $i$ whose current bandwidth is greater than its target
    if ($B_c^i > B_t^i$)          // flow $i$ is put into reduced_set
    {
        $n = n + 1;$              // # of flows in reduced_set
        $B_x = B_x + B_x^i;$      // $\sum_i B_x^i$
        $B_m = B_m + B_m^i;$      // $\sum_i B_m^i$
    }
    else                          // for flow $i \in$ unchanged flows
        $B_c = B_c + B_c^i;$      // $\sum_i B_c^i$
}

// find the total availability of the link for changed flows
$A_t = C - (B_m + B_c);$

// allocate rate to flow $i \in$ flows in reduced_set
for (i = 0 ; i < n ; i++)
{
    $Rate = B_m^i + A_t * \frac{B_x^i}{B_x};$
    if ($Rate > B_x^i$)       $B_c^i = B_x^i;$
    else                      $B_c^i = Rate;$
}
```

Fig. 3 Degrade Algorithm-TBA

```
$B_m = 0;$    $B_x = 0;$    $B_c = 0;$    $n = 0;$

/ / / for flow $i \in$ flows in the link except a handout flow
for ($i = 0, i < N, i++$)
{
// target bandwidth is the 30 percent of the maximum
    $B_t^i = 0.3 * B_x^i;$

// for flow whose current bandwidth is less than its target
    if ($B_c^i < B_t^i$)          // flow $i$ is put into upgraded_set
    {
        $n = n + 1;$              // # of flows in upgraded_set
        $B_m = B_m + B_m^i;$      // $\sum_i B_m^i$
        $B_x = B_x + B_x^i;$      // $\sum_i B_x^i$
    }
    else                          // for $i \in$ unchaged flows in the link
        $B_c = B_c + B_c^i;$      // $\sum_i B_c^i$
}

// find the total availabilty of the link for changed flows
$A_t = C - (B_m + B_c);$

// allocate rate to flow $i \in$ flows in upgraded_set
for (i = 0 ; i < $n$ ; i++)
{
    $Rate = B_m^i + A_t * \frac{B_x^i}{B_x};$
    if ($Rate > B_x^i$)       $B_c^i = B_x^i;$
    else                      $B_c^i = Rate;$
}
```

Fig. 4 Upgrade Algorithm-TBA

of bounded delivery delay, the Average-Available algorithm guarantees fairness among all flows with different QoS. If services have different amounts of bounded delay, the Average-Available algorithm does not guarantee fairness well.

We propose a new key to select changed flows in the rate adaptation. The key is the *target bandwidth* of flows in the link, which is defined as an arbitrary percent of its maximum bandwidth. It is the relative value of the maximum bandwidth; therefore, Target-Based Rate Allocation supports better fairness among flows with different QoS at any moment.

The *degrade algorithm* (Figure 3) compares the current bandwidth of all flows to their target bandwidth in the link. In this scheme, we define the target bandwidth as 70 percent of the maximum bandwidth of each flow. If the current bandwidth of a flow is greater than the target bandwidth, then we put it in the *reduced_set* with its minimum bandwidth. Otherwise the rate of a flow is not changed. Flows whose current bandwidth is over its target bandwidth are supposed to be running at high rates relatively to other flows in the link. We find the total availability of the link and flows in the reduced_set, and then allocate each flow in the reduced_set a reduced rate in proportion to its maximum bandwidth.

The *upgrade algorithm* (Figure 4) first checks the available bandwidth of the link in which all flows, except flow *i*, are running at their minimum rates. In this algorithm we compare the current bandwidth of each flow in the link with its target bandwidth defined as the 30 percent of the maximum bandwidth. If the current bandwidth of each flow is less than its target bandwidth, then we put a flow in the *upgraded_set*. Flows whose current bandwidth is greater than its target bandwidth are supposed to run at relatively high rates within the available bandwidth of the link. Consequently, we exclude those flows from the upgraded_set and do not change their rates. We find the excess availability of the link and its flows in the upgraded_set and allocate a rate to each flow in the upgraded_set in proportion to its maximum bandwidth.

5. Simulation Results

In this section, we compare the performance of the Average-Available Algorithm and Target-Based Algorithm with that of the Fair Adaptation algorithm and the Average-Fair Adaptation algorithm[14]. Because the FA and the AFA are the rate allocation algorithms that cause the lowest network overhead to support fair services in wireless networks. We use the following performance parameters: *Mean Service Ratio* and *Variance of the Service Ratio* for the fairness characteristics and *Average Time Counts* and *Average Handoff Counts* for the network overhead characteristics.

5.1 Simulation Model

We use a cellular network configuration to evaluate(Figure 5) and compare each algorithm's performance [14]. There are ten wireless cells, which are covered by base stations, and a wired network to which each of these base stations is connected by a link of infinite bandwidth. The wireless link at each cell has a bandwidth of *C* Mbps. In our system, a mobile host can move into any of its two neighboring cells. Each flow originates at a node in the wired network or a mobile host in a cell and the unicast data flows stream between the node and the mobile host through the base station of that cell.

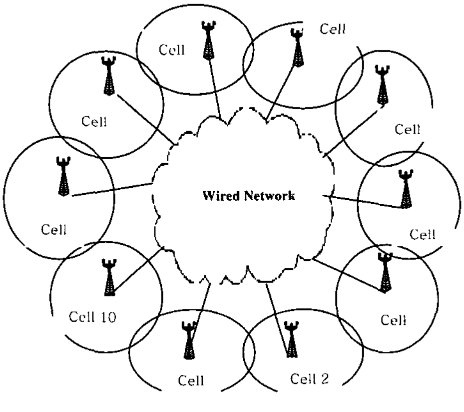

Fig. 5 Network Configuration for Simulation

We measured the values of the performance parameters by varying the mean mobility rate while keeping all other parameters fixed. We used the parameters and their values associated with the flows and host mobility as follows:

- Mean Flow Arrival Rate(λ_a): 3.0 flows/second (Poisson Distribution)
- Duration of a flow: 300 seconds
- Cell Capacity(C): 10 (Mbps)
- Mean Mobility Rate(λ_b): moves/second (Poisson Distribution)

  Then, Mean Cell Stay Time: 1/Mean Mobility Rate(Exponential Distribution)

We performed two types of experiments: one when the flows have different bounded delays, and the other when the flows have similar bounded delays. That is, the experiment 1 reflects the situation that there are various types of real-time services with different QoS specs and the experiment 2 reflects the situation that there are similar types of real-time services with QoS specs in the wireless networks. We examine which algorithm is efficient in the two different situations to support real-time services.

5.2 Simulation Experiment 1

In this experiment, flows have different QoS specs and different bounded delay. We used three types of service classes based on the minimum bandwidth requirement and the maximum bandwidth requirement.

- Service Class 1: [64 Kbps, 256 Kbps]
- Service Class 2: [256 Kbps, 1024 Kbps]
- Service Class 3: [1024 Kbps, 2048 Kbps]

Figures 6 and 7 show the *Mean Service Ratio* and *Variance of Service Ratio*, respectively, for the four rate allocation algorithms. These parameters characterize the fairness property in rate allocation. To compute these parameters, we first calculate the variance of the service ratio (current bandwidth to maximum bandwidth) of each flow for its duration. Then we compute the average variance of the service ratio (Mean Service Ratio) and the variance of the service ratio variance (Variance of Service Ratio) for all flows in the network. The small values of these parameters mean all flows suffer similar rate variations and they are served fairly according to their QoS specs. In Fair Adaptation (FA), the Mean Service Ratio is small; however, the Variance of Service Ratio is high. FA distributes rates to all flows equally then flows have similar rates, which causes large variations in the service ratio of all flows with different QoS specs. Consequently, FA does not support the different QoS specs of flows. Average-Available Adaptation(AAA) has the similar characteristics of FA. Average-Fair Adaptation (AFA) allocates a small number of flows their minimum bandwidths or maximum bandwidths. In Figures 6 and 7, AFA has the highest two values, which means the variations of the service ratio of different flows differ widely. AFA also does not support fairness among flows with different QoS. Target-Based Adaptation (TBA) has the lowest Mean Service Ratio and Variance of Service Ratio values, which means that the variations of the rates are very small. In TBA, all flows have rates in proportion to their QoS, so it supports various QoS specs of flows at all times. TBA has small Mean Service Ratio and Variance of Service Ratio values continuously even though mobility rate increases, which means that TBA supports QoS fairly regardless of mobility rate. TBA is also very sufficient in networks where each flow has different QoS specs.

In Figures 8 and 9, we compare the network overhead characteristics generated to support the fairness of flows by the four rate allocation algorithms. *Average Time Counts* and *Average Handoff Counts* denote the number of flows whose rates are changed per second and per handoff, respectively. With the increase in mobility rate, handoff rate increases. The number of changed flows per handoff remains almost invariant of the mobility rate. High mobility rate causes more rate allocation and a greater number of flows whose rates are changed. Therefore, the network overhead increases with mobility rate. FA has two parameter values that are much higher than those of any of the other algorithms. AFA and TBA have very low network overhead. AAA has a similar network overhead to these two algorithms even though the three algorithms have minor differences in the values of their two parameters. Accordingly, TBA has the lowest network overhead, but has good fairness characteristics by using QoS specs.

5.3 Simulation Experiment 2

In this experiment, flows have similar bounded delays even though their QoS specs are different. We again used three types of service classes based on minimum bandwidth requirement and maximum bandwidth requirement.

- Service Class 1: [16 Kbps, 528 Kbps]
- Service Class 2: [526 Kbps, 1166 Kbps]
- Service Class 3: [1024 Kbps, 1792 Kbps]

Figures 10 and 11 show the *Mean Service Ratio* and *Variance of Service Ratio* as a consequence of mobility rate, respectively. FA has low Mean Service Ratio and Variance of Service Ratio values by changing rates of most flows in the link, which causes very high network overhead. AFA has the highest two-parameter values, which means some flows suffer wide variation of rates, whereas some flows suffer smaller variation in AFA. Therefore AFA does not support different QoS. AAA and TBA have low Mean Service Ratio and Variance of Service Ratio values, which differ greatly from AFA. Both AAA and TBA support different QoS.

Figures 12 and 13 show the Average Time Counts and Average Handoff Counts as a consequence of mobility rate, respectively. In this experiment, the performance characteristics of the four schemes are similar to those in experiment 1. FA has significantly higher network overhead than any other algorithms. AFA, AAA and TBA have similar network overhead and the network overhead is very low relative to that of FA. Accordingly, AAA and TBA have low network overhead, but support good fairness to flows. However, the calculation overhead of AAA is lower than that of TBA. In AAA, it is easy to calculate the average availability and there is no need to maintain target bandwidth of each flow. Therefore, AAA is more efficient when flows have similar bounded delays with their different QoS specs in wireless networks.

6. Conclusion

In wireless networks, bandwidth is limited and many mobile hosts move from cell to cell. Accordingly, network resources must be used efficiently. To maximize the utilization of a network, its bandwidth must be dynamically allocated to different services within the network. We have proposed new algorithms for rate allocation (Average-Available Rate Allocation and Target-Based Rate Allocation) in wireless mobile networks using QoS specifications, where the rates of services are adjusted in accordance with the limited bandwidth of the wireless network. Fair Adaptation allocates similar rates to all flows; therefore, FA causes high network overhead and does not support various QoS specs. Average-Fair Adaptation allocates the minimum or maximum bandwidth to a small number of flows. Although AFA causes light network overhead, AFA does not support various QoS specs at all times. AFA only supports the QoS specs of each flow over its duration using relative average bandwidth. AAA and TBA are efficient and fair to all flows because the two algorithms allocate the rates of flows in proportion to their QoS specs during the handoff. Consequently, AAA and TBA always guarantee the fairness of flows having different QoS specs. In consideration of the limited bandwidths of wireless networks, AAA and TBA have light network overhead in rate allocation because they only change the rates of a small number of flows. Consequently, AAA and TBA adequately support various QoS specs fairly in wireless mobile networks. AAA guarantees different QoS specs of flows that have similar bounds on the delivery delay well. TBA selects changed flows by using the target bandwidth of each flow in the link, and thus TBA guarantees better fairness among flows with different QoS even though each flow has a different bound on delivery delay, and mobility rate increases.

To allocate the resources of networks to flows efficiently, the rate allocation algorithm does not cause network overhead and supports fairness to flows in proportion to their QoS specs. But there is tradeoff between network overhead and fairness of rate allocation. As the number of flows whose rates are changed increases, so does the amount of network overhead needed to support better fairness of flows. To reduce network overhead, we change the rates of ten percent of the flows in AAA and the rates of flows whose current bandwidth are greater or less than their target bandwidth in TBA. Through simulations, we find that the network overhead of AAA and TBA is low, and AAA and TBA support better fairness by allocating rates of flows in proportion to their QoS specs.

We are performing more simulations to support various QoS specs in wireless mobile networks. We are also planning to investigate the factors that affect the fairness characteristics of rate allocation algorithms.

References

[1] D. Clark, S. Shenker, and L. Zhang, Supporting Real-Time Applications In An ISPN: Architecture and Mechanism, Proceedings SIGGOM '92.

[2] A. Chanry, An Algorithm for Rate Allocation in a Packet-Switching Network with Feedback, Laboratory for Computer Science Technical Report TR-601, MIT, Cambridge, MA, USA, April 1994.

[3] D. A. Levin, I. F. Akydiz, and M. Naghshineh, The Shadow Cluster Concept for Resource Allocation and Call Admission in ATM-Based Wireless Networks, in Proceedings of the 1st Annual International Conference on Mobile Computing and Networking, pp. 62-74, November 1995.

[4] A. Bakre, and B. R. Badrinath, Handoff and system support for indirect TCP/IP, in Proceedings of the 2nd Usenix symposium on mobile and location independent computing, pp. 11-24, April 1995.

[5] A. Charny, D. Clark and R. Jain, Congestion Control with Explicit Rate Indication, Proceedings ICC '95, June 1995.

[6] S. Floyd and V. Jacobson, Link-sharing and Resource Management Models for Packet Networks, IEEE/ACM Transaction on Networking, Vol. 3, No. 4, August 1995.

[7] S. Jamin, P. B. Danzig, S. Shender, and L. Zhang, A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks, Proceedings SIGCOMM '95, 1995.

[8] S. Lu and V. Bharghavan, Adaptive resource management algorithms for indoor mobile computing environments, Proceedings of ACM SIGCOMM'96.

[9] K. Lee, Adaptive Network Support for Mobile Multimedia, in Proceedings of the 1st Annual International Conference on Mobile Computing and Networking, pp. 62-74, November 1995.

[10] Suresh Singh, Quality of Service Guarantees in Mobile Computing, Journal of Computer Communications, Vol.19, 1996.

[11] A. K. Talukdar, B. R. Bandrinath, and A. Acharya,, On Accommodating Mobile Hosts in an ISPN, in the Proceedings of the IEEE INFOCOM '97, April 1997.

[12] S. Lu, Kang-Won Lee and Vaduvur Bharghavan Adaptive Service in Mobile Computing Environments, IFIP International Workshop on Quality of Service '97.

[13] A. K. Talukdar, B. R. Bandrinath, and A. Acharya, Integrated Services Packet Networks with Mobile Hosts: Architecture and Performance, in the Proceedings of the IEEE INFOCOM'97.

[14] A. K. Talukdar, B. R. Bandrinath, and A. Acharya, Rate adaptation schemes in networks with mobile hosts, MOBICOM'98. The fourth annual ACM/IEEE international conference on Mobile computing and networking.

[15] T. Kwon, Y. Choi, C. Bisdikian and M. Naghshineh, Call admission control or adaptive multimedia in wireless/mobile networks, Proceedings of first ACM international workshop on Wireless mobile multimedia (WOWMOM '98), pp. 111-116, Oct. 1998.

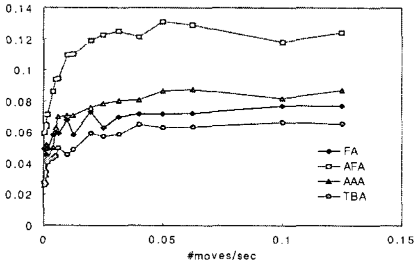

Fig. 6 Mean Service Ratio Experiment 1

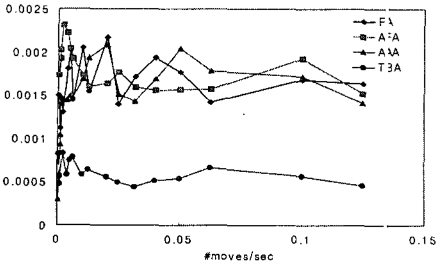

Fig. 7 Variance of Service Ratio Experiment 1

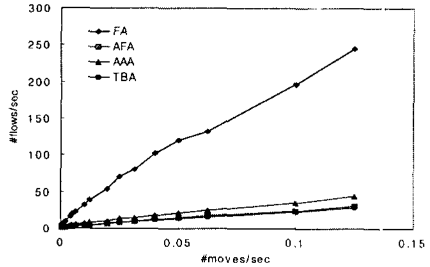

Fig. 8 Average Time Counts Experiment 1

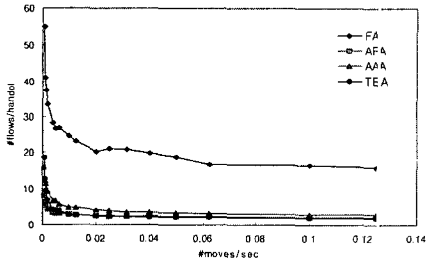

Fig. 9 Average Handoff Counts Experiment 1

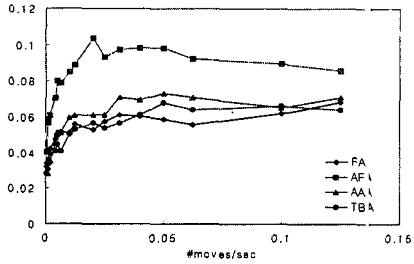

Fig. 10 Mean Service Ratio Experiment 2

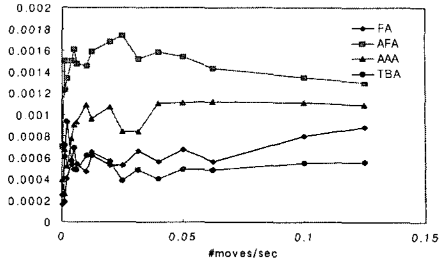

Fig. 11 Variance of Service Ratio Experiment 2

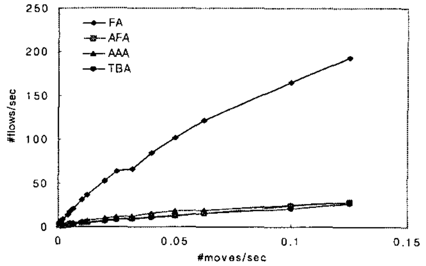

Fig. 12 Average Time Counts Experiment 2

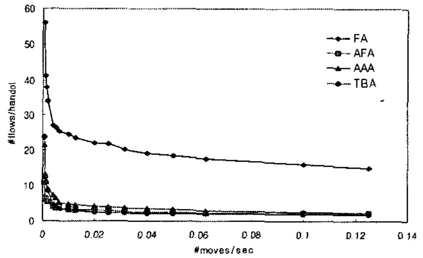

Fig. 13 Average Handoff Counts Experiment 2

강 유 화

1996년 영남대학교 컴퓨터공학과 졸업(학사). 1996년 ~ 1998년 LG정보통신 연구원. 2000년 포항공대 정보통신대학원 정보통신학과 졸업(석사). 2000년 ~ 현재 한국전자통신연구원(ETRI) 연구원. 관심분야는 Mobile Computing, Routing Protocol, QoS.

서 영 주

정보과학회논문지 : 정보통신
제 28 권 제 1 호 참조

안 성 옥

정보과학회논문지 : 정보통신
제 28 권 제 1 호 참조