PROACTIVE HOT-ZONE FLOW CONTROL FOR SEAMLESS STREAM HANDOFF UNDER MOBILE MULTICAST NETWORKS*

Sheau-Ru Tong (srtong@mail.npust.edu.tw) and Sheng-Hsiung Yang
Department of Management Information Systems
National Pingtung University of Science and Technology
Taiwan, R.O.C.

Abstract- *This paper considers a scenario of broadcasting stream media over a mobile multicast networking environment, where a server employs the wired Internet multicast to replicate and distribute stream media to a set of access points (APs), from which mobile hosts (MHs) intercept the stream through wireless unicast links. An analytical framework is introduced for characterizing the traffic dependency between AP and MH by taking into account the issues impacting the seamless stream handoff (e.g., AP out-of-synch, handoff latency and path-setup delay). Based on it, a proactive hot-zone flow control scheme is proposed, which employs a simple threshold-based policy for regulating the traffic flow between AP and MH prior to handoff. The simulation results reveal that the proposed scheme can significantly reduce the playback hiccups while consuming limited buffer space, compared with the one without any special attention. Particularly, deployment of a few-second-video-length buffer in AP and MH is capable of achieving the seamless stream handoff most of the time subject to fairly loose constraints of the AP synch error (ranging ±3 sec) and handoff latency (around 1 second).*

1. Introduction

Ubiquitous support of Internet access has been made possible by the rapid progress in the WWAN technologies (e.g., GPRS, UMTS, and CDMA2000), handheld devices (e.g., cellar phone, PDA, and notebook) and evolving mobility protocols (e.g., MIP and SIP) [1]. It enables mobile hosts (MHs) to gain the access to a wide-spectrum of multimedia information services, such as TV broadcast, news report, tourist service and commercial advertisement. The broadcast of stream media (e.g., video and audio) is one of the fundamental services utilized by these services and can be efficiently supported by IP multicast with mobility extension [2]. In principle, a multicast tree in the wired Internet is constructed, from a source to a group of access points (APs), each of which governs a region/cell. The media stream sent by the source is delivered to respective APs through the data replication done at each branch node of the tree. A MH utilizes the wireless unicast link to connect to the AP associated with the region that the MH is currently residing and intercepts the on-going stream for playback. As the MH roams from one region to the other, a handoff process is invoked for its switching the stream interception from the old AP to the new AP. We call such stream handoff is *seamless* if the MH's playback can be continued without any "hiccup". However, the challenges come from several sources: (1) First, a stream received by different APs may not be perfectly synchronized, because the data paths from the server to each AP may involve different nodes, links and traffic conditions and thus incur different network delays. This phenomenon is known as the *out-of-synch problem* [3]. (2) Second, the handoff process normally causes non-negligible signaling delay, over which we may experience some data loss or rough transmission. (3) Finally, if the MH connects to an AP, which is currently not a member of a multicast group, a long setup delay may be required for that AP to establish a new subscription. All these issues potentially cause stream discontinuity and should be carefully dealt with to ensure the seamless stream handoff. Most of the previous works primarily focus on strategic solutions for shortening handoff delay by reducing signaling overheads [5][6][7][8] or minimizing data loss [3][4][9]. But few analytical models have ever been proposed for designing the stream flow control for handoff. In this paper, we propose an analytical framework for characterizing the traffic dependency between AP and MH by taking the above timing issues into account, and derive a necessary condition for the seamless stream handoff. A *proactive hot-zone flow control* is then proposed, which determines the timing for triggering different events and buffer requirements, and employs a simple threshold-based policy for regulating the traffic flow between AP and MH, relying on the buffering status of the next AP, prior to handoff. The simulation results reveal that the proposed scheme can significantly reduce the playback hiccups while consuming very limited buffer space, compared with the one without any special attention. Particularly, deployment of a few-second-video-length buffer in AP and MH is capable of achieving the seamless stream handoff most of the time subject to fairly loose constraints of the AP synch error (ranging ±3 sec) and handoff latency (around 1 second). The rest of the paper is organized as follows. Section 2 gives the assumptions and formulation of the problem. Section 3 presents the proposed scheme. Section 4 provides the performance evaluation. Section 5 draws the final conclusion.

2. Basic Assumptions and Problem Formulation

The mobile multicast network of interest is described as follows. A video server broadcasts a video stream (video channel) to a (multicast) group of APs through a wired multicast channel, where each AP is currently serving the stream to one or more MHs. Each MH connects to and intercepts the stream from the AP designated to the region it is residing through a wireless unicast channel. A *mobile switching center* (*MSC*) is also deployed in the wired network for monitoring connections and carrying out the handoff decision about where and when a MH should be handed over to. Without loss of generality, in the following discussion, we focus on the case that a MH intercepts a video channel from a multicast group of APs.

We assume a video is packetized into a sequence of packets with a fixed packet size of p and numbered by 1, 2, 3 and so on. It is sent as a constant-bit-rate (CBR) stream with a packet rate of R using multicast. (In the following discussion, "packet" is the basic data unit.) The same CBR stream will be reconstructed at each AP. Each AP owns a buffer queue which always keeps a *window* of B_{AP} number of the most recently received packets. A MH also has a buffer queue which can accommodate up to B_{MH} number of received packets and consumes them at the same CBR. Next, we define the flowing address functions and characterize the traffic dependency between a MH and the new and old APs.

Definition 1 **(*Address functions*)*:***

$f_{MH}(t)$: the packet id that a MH is consuming for decoding at time t.
$f_{AP}(t)$: the packet id that a MH is receiving from the server at time t.

* This work is supported by the National Science Council through a grant NSC 88-2219-E-020-001.

0-7803-9332-5/05/$20.00 ©2005 IEEE

$f_{net}(t)$: the packet id under transmission (AP→MH) at time t.
$f_{AP}^*(t) \geq f_{AP}(t)-B_{AP}$: the min packet id an AP buffers at time t.
$f_{MH}^*(t)=f_{MH}(t)+B_{MH}$: the max packet id a MH could buffer at time t.

Usually the propagation delay over the wireless link is relatively short compared with the packet transmission time, so here we assume it is virtually zero. In that case, clearly $f_{net}(t)$ must fall between $f_{AP}(t)$ and $f_{MH}(t)$. Furthermore, $f_{net}(t)$ should never run behind $f_{AP}^*(t)$ and beyond $f_{MH}^*(t)$. As a result, we obtain following *address constraint* associated with an AP-MH connection.

$$f_U(t) \geq f_{net}(t) \geq f_L(t), \qquad (1)$$

where $f_U(t)=\min\{f_{AP}(t), f_{MH}^*(t)\}$ and $f_L(t)=\max\{f_{MH}(t), f_{AP}^*(t)\}$. The seamless stream handoff from one AP-MH connection to the other can be interpreted as constructing a feasible (continuous) f_{net} which satisfies these two address constraints at the same time. Figure 1 gives a visual illustration of this idea where a MH comes across two adjacent APs. We use AP_i and AP_{i+1} to indicate the current and next APs, respectively. Also, the associated functions are indexed by corresponding AP id. The *feasible* areas (highlighted with grey color) refer to the ones bounded by the address constraints with signal ranges covered by each AP. f_{net} successfully crosses over two feasible areas and causes no interruption to f_{MH}, so the stream handoff is seamless. Note that over the handoff period f_{net} incurs a "*flatten*" segment. This is because the rough transmission results in poor sustainable throughput. Also two APs may have different buffer sizes and receive the same stream with some time offsets due to the out-of-synch problem, so two feasible areas usually don't match perfectly (or even disjoin at all). If the infeasibility of f_{net} is foreseeable prior to handoff, we may call for adjustment on the AP_{i+1}'s buffer space to make f_{net} feasible. To implement this idea, a *proactive* strategy is taken as follows.

3. Proactive Hot-Zone Flow Control

Basically we try to get a media stream activated in AP_{i+1} before handoff. As shown in Figure 1, we divide the entire handoff process into three subsequent phases: (1) *query*, (2) *hot-zone* and (3) *handoff*, where t_{query}, $t_{hot\text{-}zone}$ and $t_{handoff}$ indicate the starting time of respective phases and t_{end} the end time of the process. Normally the regular stream transmission is carried out by some streaming protocol (e.g., RTP) with a CBR of R. Once in the query phase, MSC chooses the next AP (i.e., AP_{i+1}) for handoff according to an *AP consolidation policy* (to be described shortly) and requests AP_{i+1} to join the multicast group if it is a nonmember. When the stream starts to arrive at AP_{i+1}, MSC estimates a threshold h which depends on the address constraint associated with AP_{i+1} and indicates the packet id that MH should receive up to before entering the handoff phase, and adjusts AP_{i+1}'s buffer space if necessary. Next, in the hot-zone phase, the regular stream transmission is switched to the hot-zone transmission, where AP_i tries to catch up with that threshold for the traffic sent to MH. In the handoff phase, the actual handoff operation is performed and the stream transmission may be deteriorated or turned off. After the handoff phase, AP_{i+1} proceeds the regular stream transmission from the location stopped by AP_i. In the following, we first describe how to determine the threshold h and AP for achieving the seamless handoff, followed by deciding the timings of triggering respective phases.

3.1 Seamless Stream Handoff Property

For f_{net} to be continuous, the stream transmission stopped at t_{end} should be picked up by AP_{i+1} at some time $t_{hook} \geq t_{end}$. In other words, there must be a packet id l such that $f_{U(i)}(t_{handoff}) \geq l \geq f_{L(i)}(t_{handoff})$ and $f_{U(i+1)}(t_{hook}) \geq l+\theta \geq f_{L(i+1)}(t_{hook})$, where θ is the traffic amount obtained during the handoff phase. The following theorem describes the necessary condition for h to preserve such a property.

Theorem 1 (Threshold policy & seamless handoff): Suppose we employ a threshold policy that the packet id received by MH at the end of the hot-zone phase is no less than h (i.e., $f_{net}(t_{handoff}) \geq h$), where $h=f_{L(i+1)}(t_{end})$. Then the stream handoff is seamless, if $f_{U(i)}(t_{end})-f_{L(i+1)}(t_{end}) \geq \Delta t_{handoff} \times R$, where $\Delta t_{handoff}$ is the time interval of the handoff phase.

Proof: It is given that $f_{U(i)}(t_{end})-f_{L(i+1)}(t_{end}) \geq \Delta t_{handoff} \times R$. Also $f_{U(i)}(t_{handoff})=f_{U(i)}(t_{end})-\Delta t_{handoff} \times R$. We thus have $f_{U(i)}(t_{handoff}) \geq f_{L(i+1)}(t_{end})=h$. By enforcing the threshold policy, we mean that $f_{net}(t_{handoff})=l$ should fall in $[\max\{ f_{L(i)}(t_{handoff}), h\}, f_{U(i)}(t_{handoff})]$. Next, suppose θ amount of the traffic is obtained during the handoff phase, so at time t_{end}, $f_{net}(t_{end})=l+\theta$. Also, $f_{L(i+1)}(t_{end}) = h \leq l \leq l+\theta$. As time goes on, $f_{L(i+1)}$ keeps on increasing at a rate of R and f_{net} remains still. Eventually, we will find a time t_{hook} ($\geq t_{end}$) at which $f_{U(i+1)}(t_{hook}) \geq l+\theta \geq f_{L(i+1)}(t_{hook})$. Therefore, the seamless stream handoff is guaranteed. ■

For instance, as indicated in Figure 1, at t_{end}, $f_{U(i)}(t_{end})-f_{L(i+1)}(t_{end}) \geq \Delta t_{handoff} \times R$, so the threshold policy can derive a feasible f_{net} crossing over two feasible areas.

3.2 AP Consolidation Policy

There could be couple candidate AP_{i+1}'s , with their signal ranges overlapped with that of AP_i, that a MH can be handed over. They may or may not be a member of the multicast group and also have different buffer sizes and network delays (the server-AP). In some cases, the seamless stream handoff cannot be realized unless we expand $B_{AP(i+1)}$. (B_{MH} is usually quite small, so room for its expansion is very limited.) The *AP consolidation policy* is to find the "best" AP_{i+1}, described as follows. Suppose the MSC continuously monitors APs and keeps track on the address constraints associated with connections. At t_{query} the MSC has a list of candidate APs in hand and tries to choose one of them. It first considers the minimum-workload (e.g., the minimum number of active streams or clients in service) and member AP which can realize the seamless stream handoff directly. If none, it further considers the minimum-workload and member AP that needs to do the buffer expansion to realize the seamless stream handoff. If none again, it chooses the minimum-workload and nonmember AP to issue a subscription. Once the stream starts to arrive, the necessity of the buffer expansion is further checked and applied if necessary. If none again, the minimum-workload AP is finally chosen from the rest of APs. In this case, a hiccup is unavoidable. The corresponding pseudo codes for testing buffer expansion are listed in Table 1, where all possible combinations of the address constraints of AP_i and AP_{i+1} are checked to see whether the seamless stream handoff can be done or not by applying Theorem 1. If "OK" is returned, the job can be done. If "EXPAND" is returned, the job can be done only when $B_{AP(i+1)}$ is expanded up to $B^*_{AP(i+1)}$. If "REJECT" is returned, the job fails due to the fact that $f_{L(i+1)}$ is dominated by f_{MH}, and expanding $B_{AP(i+1)}$ cannot solve the problem. (A proof of the test can be found in [11].)

Table 1: Pseudo codes of the buffer expansion test

	Procedure AP_BUFFER_EXPANSION(AP_i, AP_{i+1})
1	**If** $f_{U(i)}(t_{end})-f_{L(i+1)}(t_{end}) \geq \Delta t_{handoff} \times R$ //seamless handoff test.
2	**then if** $f_{U(i+1)}(t_{end}) \geq f_{L(i+1)}(t_{end})$// AP_{i+1}'s address constraint.
3	**then** return(OK)// no buffer expansion needed.

else if $f_{AP(i+1)}(t_{end}) \geq f_{MH}(t_{end})$
  then $B^*_{AP(i+1)} = f_{AP(i+1)}(t_{end}) - f_{U(i+1)}(t_{end})$
    return(EXPAND)// buffer expansion.
  else return(REJECT)// infeasible.
else if $f_{U(i+1)}(t_{end}) \geq f_{L(i+1)}(t_{end})$ //AP_{i+1}'s address constraint test.
  then if $f_{L(i+1)}(t_{end}) == f_{AP(i+1)}(t_{end}) - B_{AP(i+1)}(t_{end})$
    then $B^*_{AP(i+1)} = f_{AP(i+1)}(t_{end}) - f_{U(i)}(t_{end}) + \Delta t_{handoff} * R$
      return(EXPAND)// with buffer expansion.
    else return(REJECT)// infeasible.
  else if $f_{AP(i+1)}(t_{end}) \geq f_{MH}(t_{end})$
    then $B^*_{AP(i+1)} = \max\{ f_{AP(i+1)}(t_{end}) - f_{U(i)}(t_{end}) + \Delta t_{handoff} * R, f_{AP(i+1)}(t_{end}) - f_{MH}(t_{end}) - B_{MH} \}$
      return(EXPAND)// buffer expansion.
    else return(REJECT)// infeasible.
End

3.3 Phase Timings

For individual phases to have sufficient time to process their jobs, their triggering times should consider the worst-case scenario. In the query phase, since the MSC continuously monitors the connections (i.e., address functions), the buffer expansion test can be quickly performed for those member APs. Such computing time is relatively short. But the MSC may also ask a non-member AP to join the multicast group to service the stream. In that case, the MSC has to wait for the stream to arrive at that AP to obtain the address functions. Such a multicast subscription delay is usually relatively long, and the maximum of it, Δt_{ms}, (assumed to be known) should be allocated for the query phase. In the hot-zone phase, the worst cast is that the threshold policy may ask AP_i to fill up the MH's buffer from emptiness, so a time interval of B_{MH}/R_{max} is required, where $R_{max} \geq R$ is the maximum sustainable network throughput (assumed to be known). In the handoff phase, the worst handoff latency $\Delta t_{handoff}$ (assume to be known) is required. So the total time interval for the handoff process equals $\Delta T_{handoff} = \Delta t_{ms} + B_{MH}/R_{max} + \Delta t_{handoff}$.

Next, we want to measure the elapse time from the present time t_c to MH's leaving the signal range of an AP, denoted by ΔT_{left}. This job is also done by the MSC. Figure 2 shows that the signal range's radius is R_{AP}, and a MH is moving at a velocity of v with a direction indicated by θ_1. The distance between the MH and AP, denoted by R_{MH}, can be computed by using the RSSI model [10]. Using the R_{MH} and θ_1, we can obtain $\theta_2 = \sin^{-1}(R_{MH}\sin\theta_1/R_{AP})$. Thus D_{MH} can be written as

$$D_{MH} = D_{AP} - R_{MH}\cos\theta_1 = R_{AP}\cos\theta_2 - R_{MH}\cos\theta_1. \quad (2)$$

Therefore, $\Delta T_{left} = D_{MH}/v$. Since ΔT_{left} is estimated based on the current status of MH and may become inaccurate as MH changes the velocity or direction, so it should be periodically updated. Meanwhile, for avoiding ping-pong effect, it is desirable to postpone the handoff process as late as possible. That is, the MSC will not start the handoff process until $\Delta T_{left} \leq \Delta T_{handoff}$. This thus gives $t_{query} = t_c + \Delta T_{left} - \Delta T_{handoff}$, $t_{hot\text{-}zone} = t_c + \Delta T_{left} - \Delta T_{handoff} + \Delta t_{ms}$, $t_{handoff} = t_c + \Delta T_{left} - \Delta T_{handoff} + \Delta t_{ms} + B_{MH}/R_{max}$ and $t_{end} = t_c + \Delta T_{left}$.

4. Performance Evaluation

We simulate an environment where 20 APs, say $AP_1, AP_2, \ldots, AP_{20}$, are lined up along a road with equal spacing. Each has a signal range of 1.35 km with certain overlaps with the neighboring APs. The video stream has a bit rate of 1.5Mbps. Given p=1KB, R is around 200 pps. Also let $R_{max}=1.5R$. In each random experiment, a MH moves on the road from the beginning to the end at a velocity of 60Km/hr. (The time for its staying in an AP region is greater than $\Delta T_{handoff}$.) The time offset of the stream perceived by an AP with respect to that perceived by AP_1 is referred to as the *AP synch error*, denoted by γ. γ is uniformly distributed over $\pm\phi$, where ϕ is referred to as the *synch error range*. We assume the *hard* handoff in a sense that no packets get transmitted during handoff, and the MH can only receive packets from the old AP before handoff or from the new AP after handoff. The proposed hot-zone flow control is applied during handoff. Δt_{ms}, $\Delta t_{hot\text{-}zone}$ and $\Delta t_{handoff}$ are assumed to be 1,200ms, 2,000ms and 1,000ms, respectively. We compare our scheme with a *primitive* scheme, where no special flow control is employed (i.e., only a CBR stream with a R rate is running between AP and MH.) B_{AP}^{upper} either means the buffer upper-bound that B_{AP} can be expanded up to under the hot-zone mode or B_{AP} used under the primitive mode. Metric of interest is the hiccup experienced by playback, which can be characterized by two factors: *starvation* o_{starve} (the number of packets that MH starves for playback) and *skip* o_{skip} (the number of packets skipped since the last received packet) (Figure 3).

Figure 4 and Figure 5 show the values of γ, o_{starve} and o_{skip} incurred by each AP of a particular experiment under the primitive and hot-zone modes, respectively, given ϕ=1,000ms, B_{MH}=5,600ms, and B_{AP}^{upper}=1,400 ms. For clarity, we translate the number of packets into video clip length (ms) and show o_{skip} and o_{starve} in positive and negative values, respectively. As it can be seen, in the primitive mode the fluctuation of γ incurs significant o_{starve} and o_{skip} in each AP. This problem is however prominently improved in the hot-zone mode due to the hot-zone flow control. Figure 6 further shows the average of o_{MH} (=$o_{starve}+o_{skip}$) incurred by an AP, taken from 100 random experiments, under various (B_{AP}^{upper}, B_{MH}) configurations for two different modes subject to ϕ=1000ms. In the primitive mode, more than 1,500ms o_{MH} still remains even when (B_{AP}^{uppe}, B_{MH}) grows up to (5,000ms, 10,000ms). On the other hand, in the hot-zone mode, o_{MH} is rapidly declined as B_{AP}^{upper} or B_{MH} increases. We observe a boundary curve on the B_{AP}^{upper} vs. B_{MH} plane, and the configurations beyond it lead to null o_{MH}. Figure 7 summarizes such boundaries for ϕ=1,000ms, 2,000ms and 3,000ms. It is clear that the curves are leveraged as ϕ increases. This is because more serious synch error requires more buffer space (in both MH and AP) to smooth out hiccups. Tradeoff exists between B_{AP}^{upper} and B_{MH}. Up to certain critical B_{AP}^{upper}, increasing B_{MH} does not help for decreasing B_{AP}^{upper}. (For instance, B_{AP}^{upper} becomes a constant of 2,800ms, 4,500ms or 5,400ms when B_{MH} is beyond 1,200ms, 1,600ms or 2,600ms when ϕ=1,000ms, 2,000ms or 3,000ms, respectively.) This is because the address constraint is primarily dominated by f_{AP} and f_{AP}^* when B_{MH} grows beyond those values. In other words, those critical points offer cost-effective configurations in a sense of minimizing the AP buffering load.

5. Conclusion

In this paper, we present a proactive flow control scheme, termed the hot-zone flow control, for implementing the seamless stream handoff in mobile multicast networks. Theory for the seamless stream handoff is derived, which leads to an AP consolation policy and a simple threshold policy which accounts for the buffering status of the next AP. We demonstrate that the proposed scheme is able to significantly reduce the hiccups in playback, while consuming only limited buffer space, compared with one without any special attention.

6. References

[1] N. Banerjee, etal., Mobility Support in Wireless Internet, IEEE Wireless Communications, Oct 2003, pp. 54-61.

[2] H. Gossain, C. Cordeiro and D. Agrawal, Multicast: Wired to Wireless, IEEE Communication Magazine, June 2002, pp. 2-9.

[3] J. Lai and W. Liao, Mobile Multicast with Routing Optimization for Recipient Mobility, IEEE Transactions on Consumer Electronics, Vol. 47, No. 1, Feb. 2001, pp. 199-206.

[4] Y. Pan, etal., An End-to-End Multipath Smooth Handoff Scheme for Stream Media, IEEE JSAC, Vol. 22, No. 4, May 2004, pp. 653-663.

[5] R. Ramjee, etal., HAWAII: A Domain-Based Approach for Supporting Mobility in Wire-area Wireless Networks, IEEE/ACM Trans. Networking, Vol. 10, June 2002, pp. 396-410.

[6] S. Sharma, N. Zhu, and T. Chiueh, Low-Latency Mobile IP Handoff for Infrastructure-Mode Wireless LANs, IEEE JSAC, Vol. 22, No. 4, May 2004, pp. 643-652.

[7] I. Wu, etal., A Seamless Handoff Approach of Mobile IP Protocol for Mobile Wireless Data Networks, IEEE Transactions on Consumer Electronics, Vol. 48, No. 2, May 2002, pp. 335-344.

[8] W. Ma and Y. Fang, Dynamic Hierarchical Mobility Management Strategy for Mobile IP Networks, IEEE JSAC, Vol. 22, No. 4, May 2004, pp. 664-676.

[9] A. Helmy, M. Jaseemuddin, and G. Bhaskara, Multicast-Based Mobility: A Novel Architecture for Efficient Micromobility, IEEE JSAC, Vol. 22, No. 4, May 2004, pp. 677-690.

[10] B. McLarnon, VE3JF, VHF/UHF/Microwave Radio Propagation: A Primer for Digital Experimenters, ARRL and TAPR Digital Communications Conference, Oct. 1997.

[11] S. Tong and S. Yang, A Proactive Flow Control Strategy for Supporting Seamless Stream Handoff under Mobile Multicast Networks, Technical Report, MIS Dept, Nat'l Pingtung U. of Sci. and Tech.

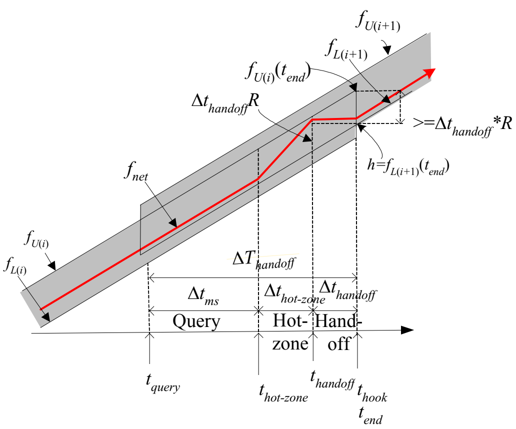


Figure 1: The seamless stream handoff process illustrated by the address constraints associated with two AP-MH connections.

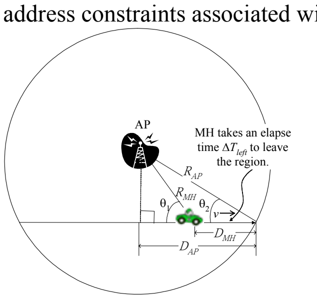


Figure 2: A MH is moving at a velocity v with a direction indicated by θ_1.

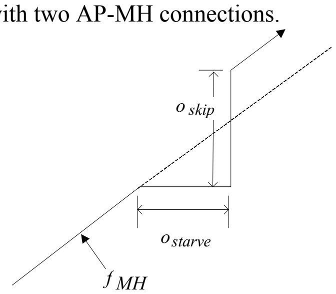


Figure 3: Illustration of hiccup factors: o_{starve} and o_{skip}.

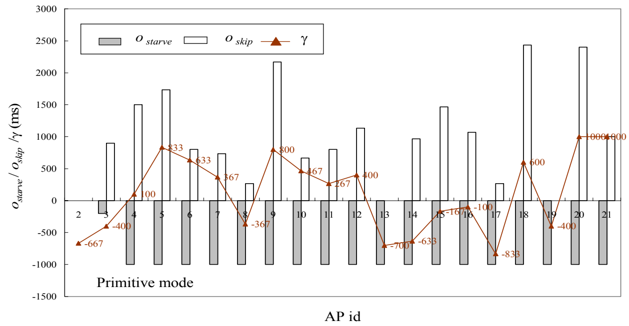


Figure 4: The values of γ, o_{starve} and o_{skip} incurred by each AP under the primitive mode (ϕ=1,000ms, B_{MH}=5,600ms, B_{AP}^{upper}=1,400ms).

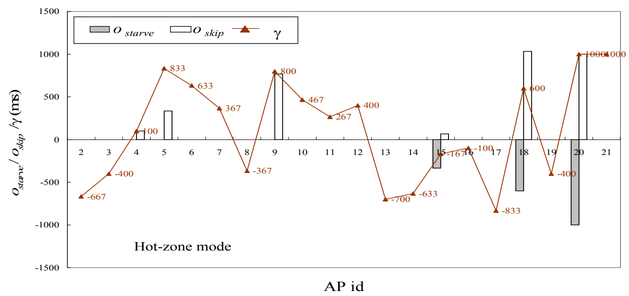
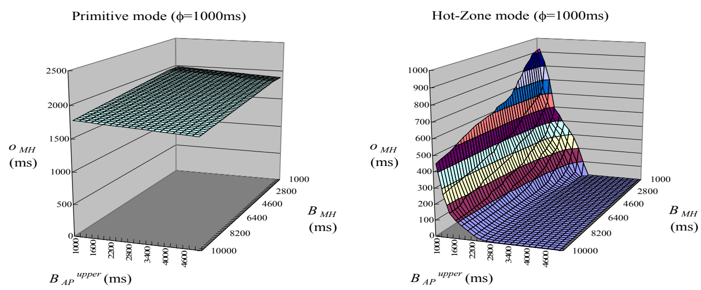


Figure 5: The values of γ, o_{starve} and o_{skip} incurred by each AP under the hot-zone mode (ϕ=1,000ms, B_{MH}=5,600ms, B_{AP}^{upper}=1,400ms).



(a) (b)

Figure 6: Means of o_{MH} incurred by an AP under various (B_{AP}^{upper}, B_{MH}) configurations under (a) the primitive mode (b) the hot-zone mode subject to ϕ=1,000ms.

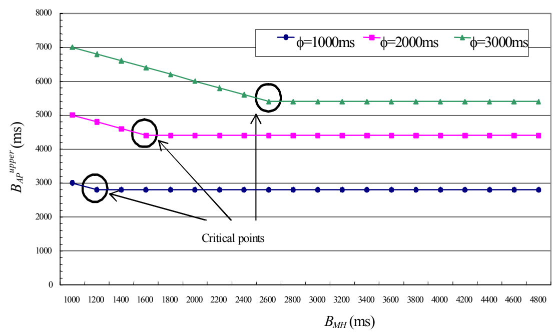


Figure 7: Boundaries of the configurations with null o_{MH} (seamless-handoff) subject to ϕ=1,000ms, 2,000ms and 3,000ms.