

# Dynamic Buffering Control Scheme for Mobile Handoff

A. Dutta, E. Van den berg, D. Famolari  
Telcordia Technologies  
V. Fajardo, Y. Ohba, K. Taniuchi, T. Kodama  
Toshiba America Research Inc.  
H. Schulzrinne, Columbia University

**Abstract**— In a mobile environment, as a mobile node moves from one point of attachment to another during an ongoing application session it is subjected to packet loss due to network and link layer transition. Such packet loss affects the quality of ongoing communication session such as interactive VoIP traffic and streaming media. We provide a solution to this scenario by buffering packets for the mobile node at an access router or network node near the edge of the network where mobile may be moving away from or moving towards. The buffered packets are then forwarded to the mobile node once the handoff process completes. The buffering scheme is used in conjunction with existing mobility protocols, access protocols or as an independent network or link layer mechanism. Ability to control the buffer dynamically provides a reasonable trade-off between delay and packet loss which is within the threshold limit for real-time communication. The overview and mechanisms of such schemes are described and comparisons on existing buffering schemes are also provided.

**Keywords-component; Packet Loss, Buffering, Delay, Real-Time Communication**

## I. INTRODUCTION

In a mobile environment, it is expected that a mobile node (MN) may experience a period of no network connectivity during its movement from one network to another due to layer 2 and layer 3 re-association. Therefore, packets in transit destined for the MN will most likely be lost during the handoff period (HP). The handoff period is exacerbated by the fact that the traffic path of packets for MN still goes through the previous network until such time as the MN has notified the proper entity of its new location. These update mechanisms vary for different mobility protocols, such as binding updates for Mobile IPv4/RO [15], MIPv6 [14] and Re\_INVITE for SIP Mobility [13]. In such cases, packet loss is incurred from the moment the MN leaves the previous network up to the time the appropriate entity learns of the MN's new location (HA or CN for Mobile IPv6 route optimization) and forwards the packets appropriately. This amount of delay and packet loss severely hinders the quality of interactive and streaming applications that are intolerant of delay and packet loss beyond certain threshold (3% packet loss, 150 ms end-to-end delay). This paper introduces a solution beyond the application end points by providing a per-mobile packet buffer at an access router or network entity (Buffering Node) near the edge of the

network where the mobile is moving away from or moving towards. Packets that are in transit during the handoff period get buffered in the Buffering Node (BN). When handoff completes, the buffered packets are flushed and forwarded to the MN in its new location. This approach provides zero packet loss for all packets destined for the MN that have reached the BN. The solution also describes a buffering scheme that enables the MN to have more granular control over the behavior of the BN to help reduce the overall handoff delay.

Outgoing packets sent by the MN during the handoff period can be also lost during the handoff process. In such a case, a BN can also be implemented locally in the MN to provide a buffering solution for egress packets during the handoff period. Having a BN in both the MN and the network edge provides bi-directional buffering during handoff and packet loss in both directions will be compensated for.

The BN may also be located within the access point specifically to assist an MN that performs active scanning. During active scanning on channels different from the currently associated access point the mobile can no longer receive packets from that access point. Current implementation uses MN power saving mode to signal the access point and allows it to start buffering on behalf of the MN. Implementing buffering functionality on the access point itself also provides the same functionality with better control on the buffering period and buffer size.

This paper is organized as follows. We describe the related work in Section II. Architecture of the framework and different functional components are described in Section III. Details of the buffering control protocol is described in Section IV. Section V describes tradeoff analysis of buffering delay. We provide the experimental results of our implementation in Section VI. Finally we conclude the paper in Section VII.

## II. RELATED WORK

There are two proposals [1], [2] that provide a similar buffering functionality. Both proposals define extensions to Mobile IPv4 and Mobile IPv6 protocol to support buffering in the network during a handover period. Moore et al [1] describe use of adding a P-bit in the mobility header of BU (Binding Update) and LBU (Local Binding Update) messages. The value of the P-bit indicates to the HA (Home Agent) when

to buffer (P-bit is '1') and when to forward the packets (P-bit is '0'). The approach is limited in comparison to the proposed method where dynamic negotiations can occur and multiple behaviors at end of buffering can be selected.

Khalil et al [2] describe a Mobile IPv4 buffering protocol that resembles the method proposed in this document. The closest similarity is the explicit definition of a control protocol to manage buffering duration, negotiate buffer size as well as traffic flow identification information (IP traffic filters). Within the restrictions of Mobile IPv4 environment this reference not only recommends explicit signaling to determine the duration of the buffering period but also implies a lease time to the buffer. Although the proposed method functions similarly, it works beyond the Mobile IPv4 topology and the BN's location is not limited to the FA (Foreign Agent) or HA (Home Agent).

Reference [4] is a Mobile IPv6 version of [2] and describes a fairly similar mechanism. A feature present in this draft that is not supported by the proposed method is buffering capability discovery. It takes advantage of IPv6 router advertisement to check the buffering capability of a network. However, the overall methods described in [4] and [2] are almost identical.

References [8], [9], and [10] provide alternative mechanisms to reduce packet loss without the use of any buffer management protocol and but depend heavily on the cooperation of the end clients. Most multimedia applications resort to playout buffers, FEC (Forward Error Correction) [10], RTCP-based feedback [8] and other techniques [9] in order to minimize the effects of packet loss or to fix the jitter. However, existing end-system assisted solutions may not be appropriate in a wireless medium where the bandwidth is scarce and the end hosts are placed wide apart.

In layer 2, there is an existing method that uses the power management functionality of IEEE 802.11 for avoiding packet loss while the MN (Mobile Node) is actively scanning [7]. In this method the MN signals the current access point that it is entering sleep mode and the access point attempts to buffer packets for the MN until the MN wakes up. However, this method cannot be used for buffering packets during a handover because the method assumes that the MN continues to be associated with the access point after it wakes up to stop buffering and the applicability is limited because the method does not carry additional information such as traffic flow identification information, buffer size and buffering period which might be required to meet particular QoS requirements.

The existing proposals are tightly coupled with specific mobility management protocols. In contrast, the proposed method can work with any mobility management protocol by allowing the protocol used for buffering control to be defined as a separate protocol. In the existing proposals, location of buffering node is limited to mobility agents such as home agent and mobility anchor point. In contrast, the proposed

method provides more flexibility on location of buffering node. In the existing proposals, forwarding of buffered packets to the mobile node after completion of the handover period depends heavily on the forwarding behavior of the mobility agent of the coupled mobility management protocol. In contrast, the proposed method defines its own tunnel establishment mechanism used for forwarding buffered packets to the mobile node to provide perfect independence of mobility management protocols. In the proposed method, detailed queuing and forwarding mechanisms for the buffering packets as well as detailed behavior in erroneous situations are defined, while such details are missing in the existing proposals.

### III. ARCHITECTURE

There are two major components required for the buffering scheme, a BN (Buffering Node) and a buffering control protocol (BCP). The BCP signals events and parameters between the MN and the BN. It is a reliable protocol composed of request (BReq) and answer (BANs) pairs with re-transmission capabilities. The protocol details are discussed in Section IV. In this architecture, there is a clear separation between buffering and forwarding that enables buffered packets to be forwarded to any node when the buffers are being flushed. In the case of mobility protocols, forwarding is performed regardless of whether tunneling is involved or not. The buffering control protocol can be used in the following manner.

- Between an end-host and its access router
- Between an end-host and its access point. In this case the buffering control protocol may be defined at either L2 or L3

The buffering model is best described using traffic flows going towards the MN. Prior to handoff, pre-handoff traffic reaches the MN via the previous network. Once MN decides to move to the new network it explicitly signals the buffering node of its intent. Buffering node then starts to buffer traffic destined for the MN that is still being forwarded through the previous network.

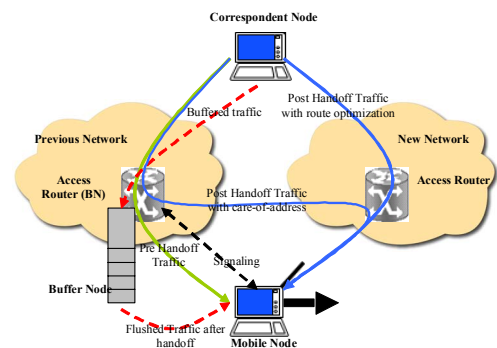


Figure 1. Buffering Model with Access Router as BN

Once handoff completes, MN signals BN (explicit or implicit) to flush the buffered packets and forwards it to the MN in the

new network. However, it is important to note that the location of the BN is not limited to this topology. The BN can reside on any node that requires temporary buffering support.

As an example, previous access router, next access router and the mobile can all have the buffering capability. We describe below the characteristics of the buffering node (BN) and interaction with other elements in the network.

A buffering node (BN) is a network entity that intercepts packets destined for one or more MN and temporarily buffers them in order to avoid packet loss while the MN is in the handoff process. To do this the BN has to be able to classify packets for each MN, buffer the packets and forward the buffered packets appropriately after handoff completes. In addition, each of these steps is signaled by a MN that requests buffering service.

In relation to MN, BN is a passive entity. MN uses the BCP to request buffering service from the BN. The protocol also communicates buffering request result and error conditions including signaling errors. BN only responds to request (sends a BAns for each BReq received) but never generates them.

The BN must be able to classify traffic and allocate buffers dynamically for each MN asking for service. Classification of the traffic that must be buffered is dictated by the MN when it sends a BReq[*initial*]. [*initial*] specifies the signal type (initial) and signal parameters sent by the MN including traffic classification information. The MN may be interested only in buffering certain types of traffic as opposed to all traffic destined for the MN. If classification information is not provided by the MN the BN should buffer all traffic for the MN.

The size of the buffer allocated by BN can also be negotiated by the MN in BReq[*initial*]. The MN may have prior knowledge of the length of the handoff period and traffic rate and thereby be able to compute a reasonable value for buffer size (bsz). However, the BN can impose limit on this value to conserve memory resources. Therefore, the BN dictates the final value for the bsz. This final value is communicated back to the MN by the BN as part of BAns[*initial*]. If the value imposed by the BN is unacceptable to the MN then the MN may withdraw its request for buffering service by signaling a BReq[*stop*] to the BN indicating its intent to terminate the buffering service.

The MN may also indicate in the Breq[*initial*] if the bsz it proposes is the amount it absolutely requires and it cannot accept any less. The MN can indicate this by setting mandatory flag in Breq[*initial*] parameters. In the case that the BN cannot accommodate this requirement then the BN should immediately terminate the service and communicate the error back to the MN via BAns[*initial*]. In such a case the BN should not even keep state information regarding the service.

Although the value of the bsz is set during the initial negotiation, it can also be dynamically adjusted during the course of buffering service.

## A. Buffering Service Duration

Since the buffering service is a transient service, buffering begins once BN has received a BReq[*initial*] from an MN and negotiated values have been established. State information regarding the service is then maintained by the BN. If for some reason the BN is not able to accommodate the initial request, it may communicate this failure to the MN as part of the BAns[*initial*]. At the least, the buffering service should encompass the handoff period for which the MN is not able to send or receive IP packets. The termination of the buffering period is the end of service (EOS) duration which can have a positive or negative outcome. A buffering period with a positive outcome can be determined using both *time-limited* and *explicit signaling* method. Both methods are mutually exclusive but complementary since one can be used to preempt the other. A buffering period with a negative outcome occurs when error conditions are met.

### A.1.1 Time-limited buffering

The BN and MN can negotiate a time-limited timeout value (bp) to determine the EOS. Expiration of the timeout prior to an explicit receipt of BReq[*stop*] constitutes a positive EOS and enforcement of existing flushing policy (FP). A large bp value can cause the buffer to overflow. In such a case, buffer overflow contingencies should be performed. The length of the timeout should encompass the handoff period for which the MN is not able to send or receive IP packets.

### A.1.2 Explicit signaling

An explicit receipt of BReq[*stop*] prior to expiration of a previously established bp value constitutes a positive EOS and enforcement of existing FP. Prior BReq[*ext*] signals may also have carried a new FP value that overrides any existing policy. In such a case, the new FP is enforced. The duration of the explicit signal encompasses the handoff period for which the MN is not able to send or receive IP packets. Explicit signaling may be used together with the time-limited buffering. It can also be used to accommodate dynamic behavior of the network such as a change in the buffer size requirement.

## A.2 Flushing Policy

The BReq[*initial*] as well as subsequent BReq sent by the MN dictates the policy that the BN should enforce against any buffered packets once EOS is met. The flushing policy (FP) enforced by the BN is one of forwarding, dropping the buffered packets and dropping the buffered packets with sending explicit error notification to the MN using the BCP or

an out-band mechanism such as ICMP. BReq[initial] from the MN should contain the default FP enforced by the BN once buffering service ends without receipt of any subsequent BReq[ext] overriding the default FP. Subsequent BReq[ext] containing a new FP overrides any previously established FP.

### A.3 Packet Forwarding

When the existing FP forwards the buffered packets after a positive EOS, the BN must be able to know the current CoA of the MN when it starts flushing the buffered packets. The current CoA of the MN may be the same as that is registered in the BN or may be different if the MN has changed its CoA after registering with the BN. To make the BCP work with any mobility management protocol, the BCP itself has a mechanism to indicate the BN about the current CoA of the MN when the BN starts flushing the buffered packets. This is achieved by including the current CoA of the MN in the BReq[stop] message. In the case of time-limited buffering where BReq[stop] is not signaled, the BCP should include the CoA in BReq[ext] signal prior to expiration of the bp. The BN should use the last signaled CoA if more than one BReq[ext] is sent by the MN that has differing CoA.

Once the BN knows the current CoA of the MN when it starts flushing the buffered packets, forwarding of the buffered packets is performed in the following way.

If the MN and BN are on the same IP subnet, the buffered packets are sent locally to the MN by using the Address Resolution Protocol (ARP). Otherwise, if the MN and BN are on different IP links, the buffered packets are sent to its next hop router towards the MN and will be routed to the MN using IP routing. In the latter case, the buffered packets are carried in an IP tunnel between the BN and the current CoA of the MN. If this IP tunnel does not exist prior to performing explicit signaling to stop buffering, it is established by the explicit signaling via a BReq[stop] and BAns exchange or bp expiration. The BN must maintain a mapping between the current CoA and the CoA of the MN in the previous network as part of the IP tunnel establishment. IP routing of the buffered packets can then result in forwarding of the packets through the IP tunnel. In some cases where this IP tunnel is established via a mobility management/optimization protocol such as FMIP6 [3], the tunnel will be used for forwarding the buffered packets instead of creating another tunnel.

### A.4 Location of BN in the Network

Although the BN can be placed at any node that is on the communication path between the MN and the CN, it is recommended to place it either at the previous access router to which the MN is attached before the handoff as shown in Figure 1 or at the new access router to which the MN is attached after the handoff as shown in Figure 2.

In the case where the BN is located on the new network (Figure 2), it may be assumed that some pre-authorization and pre-provision is done prior to MN's movement. The pre-provisioning procedure may allow the BN to determine the address of the MN even prior to moving without the aid of any mobility protocols. In all cases however, whether the BN is located in the previous or new network, we can adapt the same architecture and interaction between BN and MN does not change.

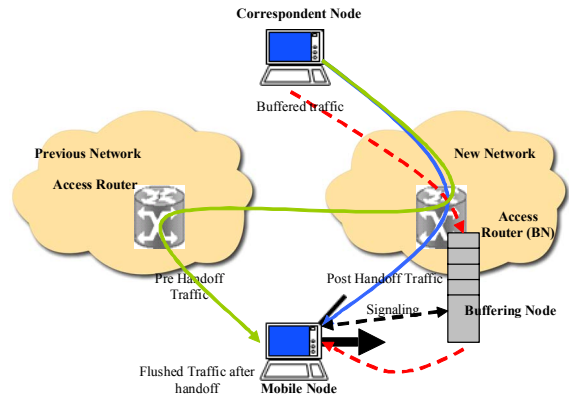


Figure 2 : Packet Buffering with Pre-authentication

Another possible location of the BN is within the MN itself as shown in Figure 3. In this case, the BN serves as a buffering service to outgoing packets for the MN during the handoff period. This usage is mutually exclusive and complements the BN locations previously suggested. With a BN present in the MN, bi-directional buffering can be achieved. As with previous locations, the architecture of the BN and its interaction with the MN does not change.

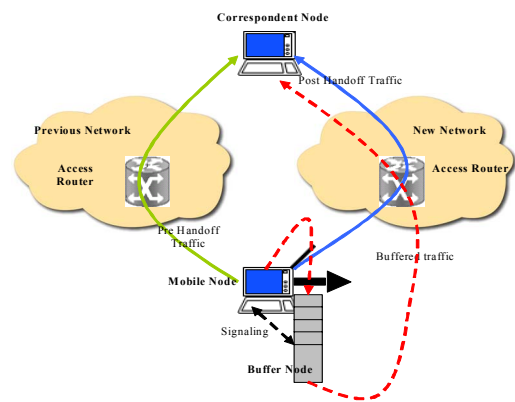


Figure 3. Buffering Node (BN) located in MN for outgoing packets

The presence of Buffering Node functionality in the MN also helps in case of ping-pong effects during handoff. Ping-pong

effects occur, e.g., when the MN fails to complete handoff in the new network and re-associates itself to the previous network. In such a case, the BN acts as recovery mechanism for packets that may have been lost during the failed handoff attempt. When buffering is performed at the MN for outgoing traffic, the forwarding behavior of the buffered packets is determined solely by the destination address of each packet. Also, no external signaling is needed for buffering outgoing packets at MN. Rejai et al [11] discuss buffering outgoing traffic to provide congestion control, however it does not specifically deal with a mobility scenario.

The BN may also be located within the previous access point to assist in buffering packets when the MN is actively scanning for finding a new access point or testing connectivity to a new access point.

#### A.5 Error condition

Error conditions that occur during buffering service which force the buffering service to terminate should constitute a negative EOS. Any existing FP may be enforced if the BN attempts to perform limited recovery. In general, however, error conditions should be treated fatal and all resources and state information allocated to the service should be released to the system.

### B. Requirements for Buffer Control Protocol

We define here some of the requirements for the Buffer Control Protocol.

#### B.1 Priority of Buffered Packets

The priority of the buffered packet should be maintained against any newly arrived post-handoff packets in order to avoid out-of-order delivery of packets. Once a positive EOS is met and BN has to forward the buffered packets, it must make certain that the buffered packets be forwarded to MN before any newly arrived packets destined for the MN. This is necessary in order to maintain the sequence of the packets being sent to the MN. Methods to accomplish this are mostly implementation specific and are dependent of mechanisms available in the system. Common methods are as follows.

##### B.1.1 Additional Forwarding Queue in the BN

An additional forwarding queue may be implemented in addition to the BN's main buffer. The forwarding queue may be used to store newly arrived post-handoff packets destined for the MN while the BN's buffer is being flushed. This has the effect of having two buffers used in sequence. Flushing of the buffers is exclusive with the main buffer having higher priority over the forwarding queue. As an example, systems can take advantage of traffic control queues (i.e. linux traffic control module) to implement queue priority.

#### B.1.2 System Receiver Queue

Some systems implement an interface receiver queue (network driver rx queue) to buffer incoming packets before being forwarded by the system. The BN can utilize this feature by pre-empting the processing of packets stored in this queue. The BN can pre-empt the system and flush its buffers before the system begins routing packets stored in the receiver queue. One caveat is that the receiver queue holds all incoming packets for that interface. So pre-empting the processing of the receiver queue would generally delay delivery of all packets and not just packets for the MN.

#### B.2 Flushing and Forwarding Rate on EOS

The flushing and forwarding rate in the BN after a positive EOS should be greater than the receive rate of the newly arrived post-handoff packets. If this is not the case, then there is greater possibility that the queue holding the post-handoff packets will be filled before the BN's buffer is empty. This will result in overflow of post-handoff queues and eventually packet loss. The flushing and forwarding can be done exclusively by the BN if the BN pre-empts all other actions in the system (i.e. spin-locking). This can guarantee that flushing and forwarding will occur prior to any other forwarding. However, the BN should take great care with this approach since a lengthy pre-emption may have adverse effect on the systems health. In all cases, the solution is implementation specific.

#### B.3 Maximum Buffering Service Duration

Maximum limits to the service duration should be administratively configured in the BN by defining a maximum size of the buffer as well as maximum duration of bp value. These limits should consider the local resources available to the BN. The limits should be considered non-negotiable values.

#### B.4. Communication Link to MN

It is assumed that signaling (BCP) between MN and BN exists before and after handoff.

### IV Buffering Control Protocol (BCP)

In this section we provide the overview of Buffering Control Protocol.

The BCP is a control protocol used by the MN to request buffering services at the BN. It is a simple and reliable messaging system composed of request and answer signal pairs. The BCP may be defined as a new protocol or as extensions to existing protocols such as PANA, SIP and Mobile IP(v4/v6) or defined as extensions to link layer protocols. In Mobile IPv6 for example, it may be possible to

define a new mobility option in Binding Update/Acknowledgement message exchange that carries the BCP in TLV format. In PANA, it is possible to define BCP AVP's that can be appended to the PUR/PUA message exchange. Other methods may be employed as long as the requirements of the BCP signaling can be accommodated. In all cases, delivery and encoding of BCP signals may become specific to each protocol that carries BCP.

### A. Protocol Signals

As a rule, request signals are sent from the MN to the BN and answer signal are sent by BN to MN in response to a request signal. The BN should never generate a request signal. Request signals carry parameters regarding the request and answer signal contains result codes. Reliability is supported by using transmission timeouts, re-transmission and error handling behavior. The BCP uses the following signal pairs.

#### A.1 BReq[initial] and BAns[initial]

These signals are always the first to be exchanged between MN and BN. It is used to establish the buffering service. These signals have the following format.

BReq[initial] = { id, bp, tc, bsz, p }  
 id – MN Id used to uniquely identify the MN to the BN. This can be the source address or MAC address of the MN.  
 bp – Buffering Period  
 tc – Application specific traffic to be classified and buffered  
 bsz – Suggested buffer size to be allocated  
 p – FP for EOS, valid values are drop, forward or drop with signal  
 flag { m } – Request flags  
 m – if set bsz is mandatory and cannot be negotiated

BAns[initial] = { id, bp, bsz, rcode }  
 id – MN Id used to uniquely identify the MN to the BN. This can be the source address or mac address of the MN.  
 bp – Buffering period for this service  
 bsz – Buffer size allocated for this service  
 rcode – Result code provided by BN

#### A.2 BReq[ext] and BAns[ext]

These signals are exchanged after establishing buffering service and before or after the MN's handoff period. They are used to extend the parameters of the buffering service.

BReq[ext] = { id, seq, bp, bsz, p, coa }  
 id – MN id sent in the BReq[initial]  
 seq – Signal sequence number  
 bp – Additional buffering period, maybe zero (0)  
 bsz – Additional buffer size, maybe zero (0)

p – new FP for EOS, valid values are drop, forward or drop with signal  
 coa – current CoA of the MN

BAns[ext] = { id, seq, bp, bsz, rcode }  
 id – MN id sent in the BReq[initial]  
 seq – Signal sequence number, must match BReq[ext]  
 hp – New buffering period for this service  
 bsz – New buffer size allocated for this service  
 rcode – Result code provided by BN

#### A.3 BReq[stop] and BAns[stop]

These signals are exchanged to stop the buffering service.

BReq[stop] = { id, p, coa }  
 id – MN id sent in the BReq[initial]  
 p – Termination FP for EOS, valid values are drop, forward or drop with signal  
 coa – current CoA of the MN

BAns[stop] = { id, rcode }  
 id – MN id sent in the BReq[initial]  
 rcode – Result code provided by BN

### B. Service Attributes

The BCP also creates service attributes (state information) within the BN. These attributes should include the following.

- MN Id (id)
- Buffering period (bp)
- Negotiated buffer size (bsz)
- Traffic classification (tc) parameter
- FP, current EOS flushing policy
- Last extension request sequence number
- Current MN CoA
- Previous MN CoA

The attributes should be allocated during the request phase. The values of the attributes are updated by the BN upon receiving valid request signals or other local events. The attributes lifetime is limited to the duration of the service. If a positive or negative EOS is met, the BN should release resource occupied by these attributes.

### C. Service Phases

In general, the BCP can divide the BN service into three phases. A buffering service must complete these phases in sequence.

#### C.1 Service Request Phases

The service request phase is the initial phase used to establish a buffering service for an MN. It spans a single

BReq[*initial*]/BAAns[*initial*] signal exchange between MN and BN to negotiate the buffering requirements and determine if the BN can fulfill those requirements. If the BN is able to accommodate the MN then state information is allocated and the buffering of MN traffic starts immediately. The BN then sends a BAAns[*initial*] to the MN with result code (rcode) indicating success. If the BN has made adjustments to hint values provided by the MN and is able to provide the service, the buffering service is also started with the BAAns[*initial*] including an rcode indicating an error. If the BN will be unable to provide service then no state information is allocated and the BN sends a BAAns[*initial*] with the appropriate error code. The MN sends the following parameters in the initial BReq[*initial*].

#### C.1.1 Buffering Period

Since the MN may have prior knowledge of the length of the buffering period, which may be an estimated value of the handover period, the MN should send this value to the BN. The BN should use this value for "bp" if it does not exceed a locally configured maximum limit. If the value is below the maximum limit the BN should set the rcode to indicate success in the BAAns[*initial*]. If it exceeds the maximum limit the BN can either deny the buffering request or use the maximum limit as the bp value. In either case the BN should notify the MN of its decision by setting the rcode to indicate an error in the BAAns[*initial*]. The MN has the option of terminating the service if the BN uses a maximum value that the MN perceives to be unsuitable. In such a case, the MN can proceed immediately to the termination phase.

#### C.1.2 Buffer Size (bsz)

The MN can suggest a specific bsz to be used by the BN during the service. This is possible since the MN may have prior knowledge of the application being used and consequently the expected traffic rate and channel conditions. In combination with the handoff period, the MN may be able to calculate a reasonable bsz needed to accommodate the classified traffic. However, the bsz provided by the MN is only a hint to the BN. The BN has the final authority on the actual bsz to be used based on locally available resources. If the BN is able to accommodate the bsz hinted by the MN, it should set the rcode indicating success in the BAAns[*initial*]. If the BN is unable to accommodate the hint value, it should use the locally configured maximum limit and notify the MN of its decision by setting the rcode to indicate an error in the BAAns[*initial*]. The MN has the option of terminating the service if the bsz issued by the BN is unsuitable. In such a case, the MN can proceed immediately to the termination phase.

#### C.1.3 Traffic Classification (tc)

By default, all traffic destined for the MN will be buffered by

the BN during the buffering period. However, the MN may provide parameters regarding the type traffic that needs to be buffered. This helps in case the MN is interested in buffering only certain types of traffic (tc). The format of the tc parameter is implementation specific and the MN maybe required having prior knowledge of this implementation.

#### C.1.4 Flushing Policy

The FP sent by the MN dictates the policy that the BN should enforce with the buffered packets once EOS is met. Currently supported policy is to forward or drop the packets. An FP value in BReq[*initial*] specifying that the BN drop the buffered packets on any EOS may seem inappropriate since the service becomes meaningless. However, there may be cases where the MN decides to enforce two different policies depending on the EOS conditions met. As an example, an MN may wish to drop all packets when time-limited buffering EOS is met and forward all packets when explicit signaling EOS is met. In such a case, the default FP should be set to drop all packets and subsequent BReq[*stop*] that signals a positive EOS should carry a FP that forward all packets.

The FP supplied in BReq[*initial*] may also be set to null. In such a case, the FP used should be a locally configured value.

#### C.2 Buffering Phase

The buffering phase spans the period when traffic classification and buffering occur. This begins immediately after the BN responds with a BAAns[*initial*] in the request phase indicating that buffering service is being provided and ends when an EOS condition is met. During this phase, the BN and MN can exchange one or more extension request (BReq[*ext*]/BAAns[*ext*]). Each exchange completes an extension request negotiation. Extension request allows the MN to re-negotiate the attribute values that has been established during the request phase or previous extension request exchanges. It allows the buffering service to be extended. A BReq[*stop*]/BAAns[*stop*] exchange may also be performed in this period to abort the buffering service. In such a case, it is expected that the FP supplied with the BReq[*stop*] enforces a policy of dropping all buffered packets.

Care should be taken when performing extension request since the signal exchange is done before or after the handoff period. The MN should synchronize the signal exchange with the handoff period to make sure network connectivity is available during the signal exchange.

In contrast to BReq[*initial*], the parameters provided by BReq[*ext*] are to be added to existing attribute values. Except for the FP, the resulting new values will be subject to the same limits and conditions imposed on the initial values. The MN sends the following parameters in every BReq[*ext*].

### C.2.1 Buffering Period

The MN may notify the BN of additional handoff delay in the BReq[ext] signal. The handoff value provided in BReq[ext] is added to the existing handoff attribute value and subjected to the limits and conditions specified. This process is repeated for each new BReq[ext] received by the BN. In the case that the MN is not interested in adjusting this specific value, it may set this parameter to zero (0).

### C.2.2 Buffer Size (bsz)

The MN may provide a hint to the BN of how much more bsz is required for the extension request. The bsz value provided in BReq[ext] is added to the existing bsz attribute value and subjected to the limits and conditions specified in Sec 3. This process is repeated for each new BReq[ext] received by the BN. In the case that the MN is not interested in adjusting this specific value, it may set this parameter to zero (0).

### C.2.3 Flushing Policy

The MN may use BReq[ext] to establish a new default FP for the current buffering service. The BN should enforce the new FP when an EOS is met. If the FP supplied in BReq[ext] is null then it is an indication to the BN to maintain the existing FP attribute value. If the existing attribute value is null then the BN should set the FP attribute value to a locally configure value.

### C.2.4 CoA Information

The BReq[ext] should also be used to update the CoA of the MN to facilitate forwarding (Sec 2.1.5). Carrying the CoA on BReq[ext] is necessary in the case that the time-limited timeout pre-empts explicit signaling to meet a positive EOS. Having a CoA parameter in BReq[ext] also makes the BCP and BN architecturally independent of the circumstances it is being used in. As an example, a BN residing on a MN (Figure 3.) can notify the BN of its previous CoA in case of ping-pong effects.

### C.3 Termination Phase

Termination phase begins when an EOS is met and ends when all resource allocated for the buffering service is released to the system. When a positive EOS is met and forwarding policy is enforced, the BN should follow the forwarding requirements described in Section 2 of the document. In case of explicit signaling, a BReq[stop]/BAns[stop] is exchanged between MN and BN to terminate a buffering service. The MN should synchronize with the handoff period to make sure that there is network connectivity during the signal exchange. An explicit BReq[stop] can also be used by the MN to abort the buffering service. The service is aborted MN signals a BReq[stop] with a FP that enforces the packets to be dropped. BReq[stop] has the following parameters. In addition, if the BN has

determined to perform forwarding of the buffered packets to the MN, and the MN and BN are on different IP links, and there is no tunnel between the BN and MN, the BN also creates a new tunnel to the MN.

### C.3.1 Flushing Policy

During the termination phase, the BN should enforce the FP supplied in BReq[stop]. If the FP supplied in BReq[stop] is null then the BN should enforce the existing FP attribute value. A FP which specifies forwarding should follow the forwarding requirements in Section 3.

### C.3.2 CoA Information

If BReq[stop] signal carries a CoA information, it should take override previous CoA information carried in BReq[ext] and should be used when forwarding the buffered packets (Sec 2.1.5). The BReq[stop] can also be used as a signal to establish an IP-IP tunnel in case the new CoA resides in the IP network different from its home network.

## D. Protocol Signal Flow

The protocol flow shown in Figure 4 provides a general sequence of the signal exchanges as it relates to the MN handoff, traffic classification and buffering.

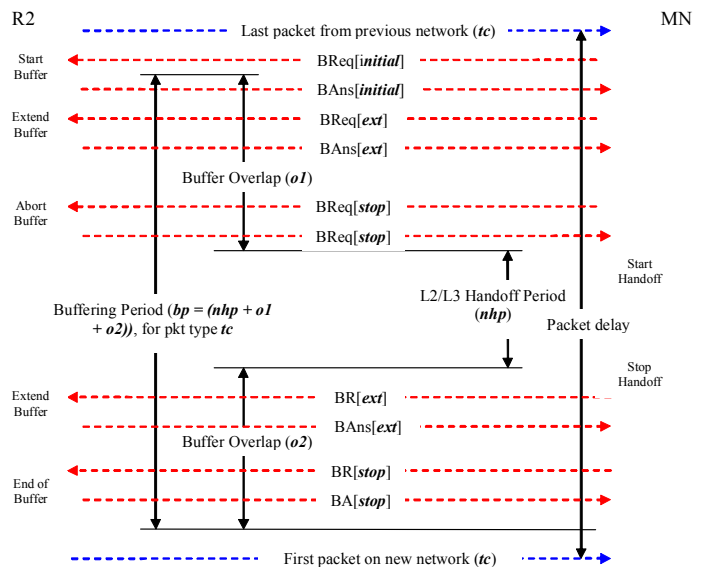


Figure 4. Protocol flow for BCP

The variables  $o1$  and  $o2$  define the period where valid MN and BN can occur when network connectivity maybe present during the buffering phase.

## E. Protocol Reliability

The BCP is a lock-step protocol which requires completion of an existing signal exchange before another signal exchange

can be initiated. This simplifies the state transition in the BN and rigidity in re-transmission.

BCP uses transmission timeouts and re-transmission to detect failure in signaling. Only request signals are re-transmitted. If an MN sends a request and does not receive an answer within a specified timeout period, it should re-transmit the request. An error condition occurs in the MN if no answer is received after a maximum number of re-transmission has occurred. Both the transmission timeout and maximum number of re-transmission should be a reasonable value to minimize overall delay. Answer signals sent by the BN should not be re-transmitted. The BN should respond with the same answer signal to any re-transmitted request. If connectivity with the MN is lost completely, the BN can rely on the hp value to terminate the buffering service.

#### F. Discovering BN availability in a network

An MN may attempt to send a BReq[*initial*] to a network entity in both the existing network and the new network it is moving towards to discover the availability of buffering service. The MN can use the lack of response from the network entity to indicate which network does not provide buffering service. The lack of response should be defined as the time when the MN reaches the maximum number of re-transmission attempts without receiving an answer. If an MN receives no response from both networks then the service is not present. If the MN receives response from both networks then the MN should choose which network will provide the service and abort the request in the network it did not choose. Alternatively, the availability of the buffering service can be provided by existing mechanisms such as DNS, DHCP or IEEE 802.21-based Information Service (IS).

### 3.7 Security Considerations

The BCP does not provide inherent security. It relies on security provided by lower layer, IP layer (i.e., IPsec) or parent protocol that carries it.

## V. BUFFERING DELAY TRADEOFF ANALYSIS

End-to-end delay of any specific packet and the delay between last packet in the previous point of attachment and first packet in the new point of attachment are most important. Buffering mechanism while reduces the packet loss, it also introduces additional delay to the both of the above parameters. An in-handoff packet that would have got lost otherwise gets buffered in the buffering node for a certain period of time that is determined by the handoff delay and time taken to flush the packets from the queue at the new point of attachment. Yemini [18] provides a tradeoff analysis between delay and packet loss. It also stresses the fact that as soon as the threshold of buffer is exceeded any newly arriving packets cause the first packet of the queue to be lost. Although this paper focuses on

the queue at the sender side it could generally be applicable to the general theory of the buffering protocol described in this paper. In case of explicit signal buffering, buffering period is equivalent to the total handoff period and additional time taken to flush the buffer after the handoff has taken place. Total number of packets stored in the network buffer depend on the buffer length, transmission time and packet generation rate at the source. Packets arrive in the buffer at a regular interval. But when these packets are flushed out of the buffer after the handoff, all the buffered packets are flushed out at the same time without any inter-packet gap. Although this avoids packet loss, the mobile is subjected to a spike since these packets arrive on the mobile almost instantaneously. The in-handoff packets that are buffered in the edge router are subjected to an increased amount of delay compared to pre-handoff and post-handoff packets. But each consecutive in-handoff packet is subjected to a different amount of delay since these packets spend different amount of time in the buffer. Later packets are subjected to lesser amount of delay compared to the packets that got in first. End-to-end delay for in-handoff packets, delay between last packet in the old network and first packet in the new network and total number of packets affected due to handoff are some of the metrics of interest to support real-time communication. Packet generation rate at the source, handoff period, time taken to signal the buffer to flush, packet transmission time are some of the parameters that affect the optimal buffer length at the router. While the overall buffering period is influenced by the handoff delay, it affects end-to-end delay, number of packets delayed, and the jitter. However, the jitter observed due to buffering at the router node can be compensated by the playout buffer at the mobile. In the next section, we provide the experimental results showing how different amount of traffic rate and buffering delay may affect some of these performance metrics.

## VI. TESTBED AND PERFORMANCE EVALUATION

We have implemented parts of this Buffering Control Protocol to reduce the packet loss for MPA assisted handoff. We have experimented with both the “*time limited buffering*” and “*explicit signaling buffering*” for different traffic rates.

The current solution is implemented using kernel queue module that hooks into linux netfilter’s [12] QUEUE handler. The new module is called ip\_mparb (IPv4 MPA router buffer). This has the following advantages:

- Packet classification is done by iptables so the module is much simpler. It will simply rely on iptable’s packet classifier with the ip\_mparb as the target.
- Implementation is efficient since packets are routed to the module in sk\_buff objects so no copying is done. ip\_mparb simply queue’s the sk\_buff’s without modification.
- Implementation is very fast since ip\_mparb is a kernel level module that becomes part of the ip routing stack. No additional socket mechanism required.

- Easily meets the requirement of maintaining packet sequence since all packets that must be buffered have to pass through this module. So when buffered packets need to be flushed, they can be transmitted first prior to allowing newly arrived packets to be transmitted.
- ip\_mparb can use any queuing discipline we require. At the moment, a simple FIFO queue is used.

Table 1: Results of Time Limited Buffering

Traffic Rate	X (ms)	Y (ms)	PD (ms)	Packet Loss	Packet Buffered
70 pkts/Sec	0	30	27	0	19
	0	20	29	0	3
	0	10	12	0	1
80 pkts/sec	0	30	28	0	3
	0	20	33	0	3
90 pkts/sec	0	30	69	0	3
	0	20	46	0	3
	0	10	11	3	1
100 pkts/sec	0	30	69	0	10
	0	20	46	0	4
	0	10	11	3	1

User level interaction is limited to simple control events so we can use existing user level commands that can pass control events to kernel modules. In the following paragraph we present some of the results. Initially we used SIP as the signaling protocol and RAT as the media. We found no packet loss while using RAT at 60 packets/sec. We wanted to see how buffering may affect the packet loss and delay by increasing the packet rate. We increased the packet generation rate to 70 packets/sec through 90 packets/sec.

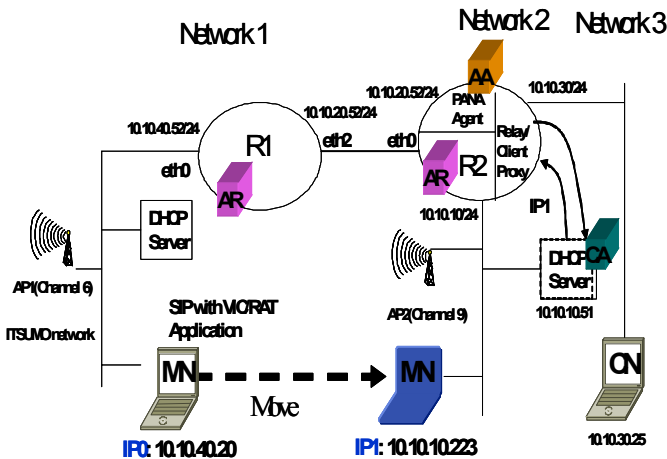


Figure 5: Experimental Test-bed

Figure 5 shows the experimental test-bed where this buffering scheme has been experimented. Figure 6 shows an overview of the ideal timing sequence for time limited buffering as

experimented in the test-bed. All variables shown are time variables. Table 1 and Table 2 show values of average packet delay, average packet loss and average number of packets buffered using both time limited and explicit buffering approaches. The most ideal scenario is to reduce  $o1$  and  $o2$  to zero though this is not possible for all practical purposes. A negative value for  $o1$  and/or  $o2$  will result in packet loss.

This means that  $hp$  is not encompassed within  $y$ . Based on the experiments,  $o1$  and  $o2$  can be fine tuned using  $x$  and  $y$  where  $y$  is based on  $hp$  and  $x$  is based on average round trip time. In addition, another alternative is the use of an explicit flush message instead of fine tuning  $y$  (noted as PUR if  $y=0$  in the figure above). The experimental results are based on packet generator's rate of 70, 80, 90 and 100 packets/sec. The rate is based on a value that is greater than the codec rate of RAT used in the MN (60 packets/sec). Also, R2 switch over process (deleting the tunnel, updating the ARP cache etc.) always occurs during  $o1$  immediately after R2 begins buffering. This switch over period is very small (average about 0.300 to 0.500 ms) so it is not considered in the figure above. All values are based on averages where the average is taken from at least 3 test samples. Experiment results are based on modified "madwifi" driver with Netgear 802.11b WiFi cards in the MN. Modification involves only optimization with no functional changes.

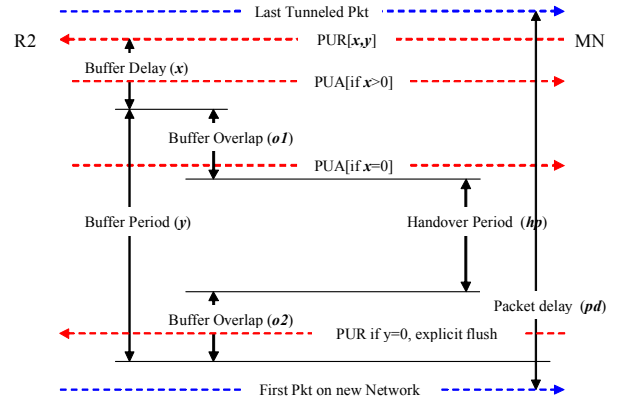


Figure 6: Protocol Flow with PANA as BCP

Table 1 summarizes the average results of first four samples that use "madwifi" driver and netgear wifi card with time-limited buffering approach. Four different packet generation rate are experimented. The average  $hp$  is 10-16 ms. This includes L2 and L3 related delay that happens in sequence. Average L2 delay is 5-8 ms followed by average L3 delay of 5-8 ms. L3 related delay is the delay associated with assigning the previously obtained IP address to its physical interface. Bulk of the handoff delay is avoided because of many of the handoff operation done proactively. The buffering node in the experiment is the next access router that is discovered before the mobile moves to the new network. Because the  $hp$  value is very small,  $y$  can be fine-tuned to its minimum of 10 ms without incurring packet loss. Value of "x" has been kept to

zero as it introduces additional delay. When we used RAT as the media agent, it generates an average 60 pkts/sec, there is no discernible loss in the audio sample and almost always have no packet loss.

Table 2: Results from Explicit Buffering

Rate	X (ms)	Y (ms)	PD (ms)	PL	Avg. PB
70 pkts/sec	0	N/A	36.62	0	2.5
80 pkts/sec	0	N/A	42.14	0	3
90 pkts/sec	0	N/A	44.30	0	3.5
100 pkts/sec	0	N/A	44.50	0	4

Table 2 shows the experimental results under the same environment (madwifi driver and netgear wifi card) but using explicit buffering approach. The average *hp* is 10-16 ms. Using explicit signaling between MN and R2 to flush the buffer, it is guaranteed that no packet loss will occur compared to using a value of *y* that has the possibility of having  $\alpha < 0$  resulting in packet loss. The price is additional delay. As an example, when using an “ideal” *y* value at 70 pkts/sec the avg delay is only 12 ms as compared to explicit signaling which is 36 ms. Similar to the case of time-limited buffering, experimental results using RAT as a media did not produce any discernible loss in the audio and is guaranteed to have no packet loss.

Detailed break-down of delay for an experiment with 100 packets/sec traffic rate, 1024 bytes of packet size and explicit signaling is given below. Total handoff (handover) delay is about 12.5 seconds. Layer 2 (L2) delay with “madwifi” driver is about 4.8 ms, and L3 configuration time is about 0.5 ms (L3). Processing delay (a) for buffer request at PAA =5.699 ms, switch over period at PAA (b) is 0.46 ms that includes tunnel setup and ioctl calls. Processing delay at MN (c) to send stop request is 6.788 ms. Processing delay (d) to flush the packets at the buffer is 4.626 ms. Flushing period (e) at PAA is 0.205 ms. Thus the total handover delay (hp)  $D = (L2/L3/c) + d + e$ . Thus it appears from the above that the handover period is a fraction of the total packet delay incurred. Explicit signaling method adds to the total packet delay because of the delay associated with the flushing where as time limited signaling increase the probability of packet loss.

## VII. CONCLUSION

We have presented the specification of a buffer control protocol that helps reduce the packet loss for a mobile that is subjected to rapid handoff. This buffer control protocol is generic enough to work along with any mobility management protocol. It allows the buffering node to be placed at any part of the network including the mobile node itself. We present

the experimental results of an implementation that is part of MPA framework. Dynamic buffering approach in conjunction with MPA helps reduce the packet loss to almost zero while keeping the delay bound within a limit that is acceptable for real-time communication. We infer from these experimental results that there is a strong relationship between the handoff delay and buffering period. Dynamic buffering method allows us to reduce the packet loss at the cost of additional delay.

## REFERENCES

- [1] N. Moore, J. Choi, B. Pentland, "Tunnel Buffering for Mobile IPv6", draft-moore-mobopts-tunnel-buffering-00.txt, July 2004, work in progress
- [2] M. Khalil, H. Akhtar, E. Oddaura, C. Perkins, A. Cerpa, "Buffer Management for Mobile IP", draft-mkhalil-mobileip-buffer-00.txt, October 1999, work in progress
- [3] R. Koodli, "Fast Handovers for Mobile IPv6", RFC 4068
- [4] G. Krishnamurthi, R. Chalmers, C. Perkins, "Buffer Management for Smooth Handovers in Mobile IPv6", draft-krishnamurthi-mobileip-buffer6-00.txt, July 2000
- [5] Chul-Ho Lee, Dongwook Lee, JongWon Kim, "Seamless MPEG-4 Video Streaming over Mobile IP-enabled Wireless LAN", Network Research Workshop 2004/18<sup>th</sup> APAN meeting
- [6] C. Perkins, K-Y. Wang, "Optimized smooth handoffs in Mobile IP", Proceedings of IEEE Symposium on Computers and Communications, July 1999
- [7] Pejman Roshan, Jonathan Leary, "802.11 Wireless LAN Fundamentals", Cisco Press, ISBN 1587050773, December 2003.
- [8] J. Rosenberg, H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction", RFC 2733, December 1999
- [9] C. Perkins, O. Hodson, "Options for Repair of Streaming Media", RFC 2354, June 1998
- [10] Bremen J., Wenger S., Sato N., et. Al., "Extended RTP Profile for RTCP-based Feedback (RTP/AVPF)", draft-ietf-avt-rtcp-feedback-11.txt, August 2004
- [11] Reza Rejaie et al. RAP: An End-to-end Rate-based Congestion Control Mechanism for Real-time Streams in the Internet", March 1999
- [12] B. Aboba and J. Wood, "Linux netfilter Hacking HOWTO", <http://www.netfilter.org/Documentation/HOWTO>, Rev 1.14, July 2002.
- [13] Elin Wedlund, Henning Schulzrinne, "Mobility Support using SIP" in IEEE/ACM Multimedia Conference WOMOM 1999.
- [14] D. Johnson et al, Mobile IPv6, RFC 3375, IETF
- [15] C. Perkins et al, Mobile IPv4 RFC 3344, IETF
- [16] A. Dutta, T. Zhang, Y. Ohba, K.Taniuchi, H. Schulzrinne "MPA assisted Optimized Proactive Handoff Scheme", ACM Mobiquitous, 2005
- [17] D. Forsberg et al, "Protocol for Carrying Authentication for Network Access (PANA)", draft-ietf-pana-pana-07 (work in progress), December 2004.
- [18] Yechiam Yemini, "A bang-bang principle for Real-time transport protocols", ACM SIGCOMM, 1983.