Adaptive Packet Buffering Algorithm Based on Priority and Traffic Throughput for Reducing Packet Loss in Fast Handover for Mobile IPv6

Anesa Maolod Omar Al-Najeh[1], Kamalrulnizam Abu Bakar[1], Ali Safa Sadiq[1], and Kayhan Zrar Ghafoor[1]

[1] Department of Computer Science & Information Systems, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor Bahru, Malaysia

Abstract

The packet loss has become an important issue to the research community, which needs to be addressed. In FMIPv6, Packet losses are significantly related to the handover latency and buffer size used for packet buffering. In the case of increased handover latency or decreased buffer size, packet losses will be increased. To solve the problem, we propose an adaptive packet buffering (APT) algorithm based on priority of packets and traffic throughput in layer 3 (L3) were the packets are buffered by the predefined rule in the new access point during handover. This algorithm is designed to reduce packet loss in FMIPv6 and high level of throughput and low delay can be achieved through the proposed technique. To achieve a fair comparison with Adaptive Buffer Limit Tuning (ALT) algorithm, we have implemented the APT algorithm in Omnet++ along with the FMIPv6 to develop the model and the algorithm. The results of the simulation study show that the proposed algorithm can reduce the packet loss as well as the delay.

Keywords: *Packet Loss; Mobile IPv6; Fast Handover for MIPv6; Packet Buffering; APT; ALT.*

1. Introduction

Because of the increasing usage of wireless internet, both the needs and interests in the mobility management based on IP are growing to support seamless real time service. To this end, IETF (Internet Engineering Task Force) proposed Mobile IP which is a protocol supporting the mobility management based on IP [1]. In addition, Mobile IPv6 allows nodes to remain reachable while moving towards the network. The Mobile IPv6 (MIPv6) has some features that make it more efficient than Mobile IPv4 from many aspects. It sustains real-time traffic connecting the Mobile Node (MN) and Correspondent Node (CN) [3]. MIPv6 offers transparent host mobility included in IPv6 address [9][13]. The protocol guarantees the free movement of the Mobile Nodes among networks without changing its IPv6 address. However, every time a mobile node moves to other subnet, it has to update the mobile node's new Care-of-Address (nCoA) in the home agent and the correspondent node. While performing these operations, a time period during which packets are not able to be sent or received between the nodes exists.

Mobile IPv6's performance was improved through sum of mobility protocols, particular concerning handoff or handover processes where these protocols are mostly used for environments having not enough specifications like data delivery delay, packet loss, and signalization overload [3]. The primary cause of packet loss resulting in performance degradation of standard Mobile IPv6 (MIPv6) is the handover latency [15]. Therefore, when there is a handover for Mobile Node (MN), it would cause packet loss. It will be a big problem through sessions of real-time between the Mobile Node (MN) and the Correspondent Node (CN). The focus of this research will be to reduce packet loss in Fast Handover for Mobile IPv6. However, the handover latency is unacceptable in most real time services because of the well-known triangle problem [4][7].

To handle this issue, IETF has designed fast handover Mobile IPv6 (FMIPv6) [2]. It is able to reduce the packet loss rate and handover latency by performing predicted network layer handover with the use of the trigger of link layer. FMIPv6 [12] addresses the ability of a MN to send packets immediately when it detects a new subnet link, and the ability of an access router to deliver packets to a MN directly when the MN attachment is detected. Achieving these points will reduce the handover latency, by eliminating the unreachability of the MN as it will be able to receive packets during handover. FMIPv6 minimized packet loss and latency due to handover, critical for real-time services, and for establishing new communication paths to the MN at the new access router (nAR). The ambition of this protocol is to permit the MN configuration to a new care of address (nCoA), or to current care of address (cCoA) directly after a new access router's connection (nAR), before it leaves to a new sub-network. MN acquires an nCoA and registers with previous or old AR before get link to the nAR. As soon as MN leaves the current link, old AR starts forwarding traffic to nAR [2][3].

Packet buffering is able to reduce packet losses and service disruptions caused by handover [10], where the packets are buffered in the access router or point of the new subnet. It can also be performed in an access router to achieve seamless handover. However, the scalability problem may occur because access router has to manage all the connections of the mobile nodes communicating with the IP nodes residing at the outside of the subnet.

The author in [2] assumed that a New Access Router (NAR) may receive packets forwarded by the Previous Access Router (PAR) of an MN even before the MN attaches itself to the NAR. However, unless the NAR buffers the packets, loss of packets may occur. The same condition applies when a Fast Binding Update (FBU) is sent after the MN is attached to the NAR, where unless they are buffered, the packets sent to the PAR will be lost. Hence, an option is provided in the Handover Initiate message to indicate a request for packets to be buffered in the NAR. Should the PAR requests for the feature, buffering support needs to be provided by the PAR itself. However, there is a problem in the generation of a triangle routing between the previous and the new access routers as well as the Correspondent Nodes (CN).

The proposal for a scheme for an enhancement of the buffer management in the Fast Handover protocol by [7] contains two parts. It depicts that in a handoff process, buffers are used in both PAR and NAR while only the NAR buffers the packets in the original Fast Handover, thus improving the network's total utilization of buffers. In the second part, the services in a handoff process are defined into three types based on the packets' traffic characteristics to ensure that the packets are treated accordingly. Packet loss can be reduced through buffering. Nevertheless, if the number of Mobile Nodes those attached to the router increased, the efficiency of a router will be affected by the overhead on the router. Though the proposed buffer management scheme mentioned above is meant to improve the utilization of buffer for the Predictive mode for FMIPv6, there is the problem regarding the implementation of the architecture.

The paper [15] suggested an improved fast handover schema in MIPv6 via accessing the link layer data. The fast handover protocol is tailored to reduce the packet loss through a handover. This approach can avert the packet loss in Mobile IPv6 (MIPv6). It is quite obvious that FMIPv6 has better performance under handover latency and packet loss because that fast handover protocol is tailored to eliminate packet loss and latency during handover. Two factors control number of packet lost. The first is buffer size used for packet storage for potential handover, and the second is the rate of sending, when no buffer is used like Standard Mobile IP (S-Mobile IP) the number of packet lost gets reduced, meaning that if the buffer used is big enough, no packet loss can occur.

The buffer size can be adjusted with respect to the sending rate, i.e. size increase as sending rate increases. The new link connection on the available network interface is always established the moment MN senses a new available network. It is quite clear that FMIPv6 approach functions more efficiently with respect to the handover latency and packet loss. Both handover latency and packet loss prevention are reduced under this approach utilization. The algorithm can also be tested for different mobility models in IPv6 network with neighbor data accessed. In addition, the performance can be evaluated when running fast handover for MIPv6 on Wimax networks.

Recently, to alleviate this problem, packet buffering has been implemented in the access point since it manages fewer mobile nodes than access router [4-6]. Note that the buffer space is limited in a router or access point. Therefore, the arriving packets will be lost after the buffer is full. This means that packets may be dropped regardless of the service characteristics during packet buffering for handover. For real time service such as IP telephony, for example, packet loss is often unacceptable. To satisfy the requirements of respective service, thus, the packets have to be buffered differently according to the service characteristics during handover.

There exist various approaches of packet buffering. The double packet buffering approach [7] was proposed to reduce the loss of high priority packets by buffering them in both the previous and new access router. However, the Fast Mobile IP has to be modified because additional operations are required for performing the double packet buffering. LT-Buffer (Link-Triggered Buffer) mechanism was pro-posed in [8], which utilizes the advantages of cross-layer interactions between the link layer and network layer so that packet loss can be reduced.

In this paper we propose an adaptive packet buffering tuning algorithm APT based on priority and traffic throughput to reduce packet loss in FMIPv6 when the Mobile Node (MN) move from one network to the another new network. With the proposed algorithm, the packets can interrupt the previously buffered packets or be interrupted based on the priority of its PHB during packet buffering. The evaluation of the proposed algorithm APT provided through simulated experiments in order to compare its efficiency to the existing ALT algorithm through computer simulation Omnet++.

The rest of the paper is organized as follow. In Section 2, we discuss FMIPv6 and ALT algorithm. The adaptive packet buffering tuning algorithm (APT) is proposed and the packet loss occurrence possibility of different PHB packet is analyzed in Section 3. The performance of the algorithm and results are evaluated in Section 4. Finally, we conclude the paper in Section 5.

2. Related Works

2.1 Fast Handover Mobile IPv6 with AP Packet Buffering

FMIPv6 was proposed by IETF to care the drawbacks of Mobile IP. FMIPv6 is able to great reduce the handover latency by performing predicted handover through the link layer trigger. However, the mobile node cannot avoid the link switching latency since most of the mobile nodes are attached to only one access point. Such latency may cause packet losses. As a result, the packets are buffered during handover in FMIPv6 [2] to minimize packet losses.

Packet buffering needs to be properly implemented to achieve seamless handover. Packet buffering in the access router has the scalability problem since an access router has to manage all the mobile nodes communicating with the nodes. To alleviate this problem, it is better to implement packet buffering in the access point managing fewer mobile nodes than access router [4-6].

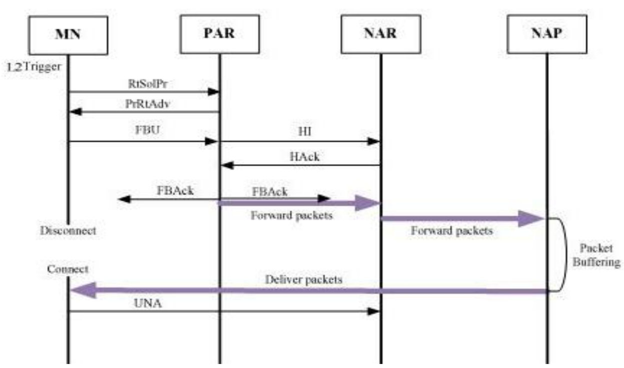

Fig. 1 Fast Handover Mobile IPv6 with AP packet buffering.

Table 1: The description of handover messages

Message	*Description*
RtSolPr	Router Solicitation for Proxy Advertisement
PrRtAdv	Proxy Router Advertisement
HI	Handover Initiate
HAck	Handover Acknowledge
FBU	Fast Binding Update
FBACK	Fast Binding Acknowledgement
UNA	Unsolicited Neighbor Acknowledgement

Figure 1 shows the procedure of fast handover with AP packet buffering, and Table 1 lists the description for handover messages, respectively. When a Mobile Node (MN) moves to a New Access Router (NAR), the signal power gets decreased and then link trigger occurs from the link device of the Mobile Node (MN). When the Mobile Node (MN) receives the link trigger, it predicts handover. Then, it sends 'RtSolPr' message to the Previous Access Router (PAR), which sends 'PrRtAdv' message including the information of the New Access Router (NAR) to the Mobile Node (MN). The Mobile Node (MN) makes new Care-of-Address (nCoA) using the information, and sends 'FBU' message to the Previous Access Router (PAR). Upon receiving the 'FBU' message, the Previous Access Router (PAR) sends 'HI' message to the New Access Router (NAR). When the New Access Router (NAR) receives the 'HI' message, it performs DAD (Duplicate Address Detection) and initiates the tunneling between the Previous Access Router (PAR) and itself. After that, the New Access Router (NAR) sends 'Hack' message to the Previous Access Router (PAR) to inform the result. If the new Care-of-Address (nCoA) made by the Mobile Node (MN) already exists in the New Access Router (NAR), the 'Hack' message includes a new Care-of-Address (nCoA) made by the New Access Router (NAR). The packets arriving at the Previous Access Router (PAR) are forwarded to the New Access Router (NAR) through the tunnel, and the New Access Router (NAR) delivers the packets to the New Access Point (NAP). Then, the packets are buffered in the New Access Point (NAP). The Previous Access Router (PAR) sends 'FBAck' message to the Mobile Node (MN), and the New Access Router (NAR) replies the 'FBU' and 'HAck' message. The Mobile Node (MN) reduces the handover latency because these operations are performed before link disconnection. After completing the link switching, the New Access Point (NAP) delivers the buffered packets to the Mobile Node (MN), and the Mobile Node (MN) informs the New Access Router (NAR) to complete the handover by sending the 'UNA' message to the New Access Router (NAR).

Packet losses are closely related to the handover latency and buffer size used for packet buffering. In the case of increased handover latency or decreased buffer size, packet losses will be increased.

2.1 Adaptive Buffer Limit Tuning (ALT)

ALT algorithm [11] objective is to simultaneously achieve both low delay and high throughput efficiency. As it is difficult to select a fixed size for a buffer which can suit the range of conditions of a network, observations done shows that a strategy to be considered is the usage of adaptive buffer sizing as shown in Figure 2.

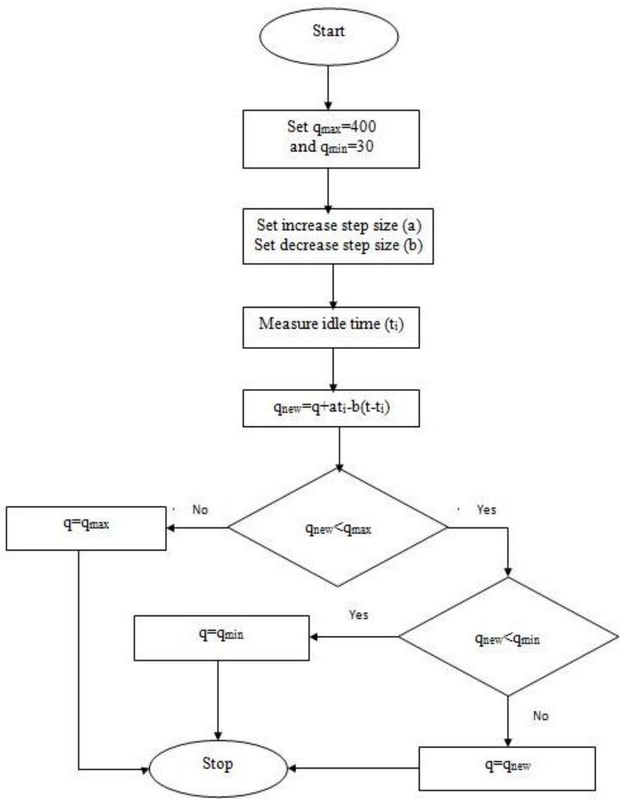

Fig. 2 An Adaptive Buffer Limit Tuning (ALT) Algorithm.

Researchers have shown that a wireless station's own service rate can be measured by the station itself through observation where t_S represents the time in which the packets take to arrive at the head of the queue of the network interface and t_e being the time the packets are transmitted successfully (t_e takes into account the time the corresponding MAC ACK is received). These measurements are implementable in real devices with minor burden of computation. At time t, with T_{serv} (t) as the inter-service time, exponential smoothing is used for the calculation of the mean inter-service time:

$$T_{serv} = \alpha T_{serv} + (1-\alpha)(t_e - t_S); \quad (1)$$

whereas $\alpha = 0.999$.

With the use of the known measurements, an adaptive strategy was proposed where the target queuing delay is T. A buffer size, Q, is then selected in accordance to the formula:

$$Q = min(T/T_{serv}, Q_{max}); \quad (2)$$

whereas Q_{max} set to = 400 packets.

This will cause the size of the buffer to be decreased with the fall of the service rate or increased with the rise of the rate. Therefore, an approximated delay in the constant queuing can be maintained to T *seconds*. As a result, the buffer size can be regulated thus, equaled to the BDP in accordance the variety of service rates. However, packet losses are closely related to the handover latency and buffer size used for packet buffering. In the case of increased handover latency or decreased buffer size, packet losses will be increased.

To account for the impact of the stochastic nature of the service rate on buffer size requirements, the researchers modified this update rule to

$$Q = min(T/T_{serv} + a, Q_{max}) \quad (3)$$

where a is an over- provisioning amount to accommodate short-term fluctuations in service rate. Based on the measurements, they had found that a value of a equal to 400 packets works well across a wide range of network conditions.

The effectiveness of this simple adaptive algorithm is illustrated. Here they plotted the throughput percentage and smoothed RTT of download flows as the number of download and upload flows is varied. It can be seen that the adaptive algorithm maintains high throughput efficiency across the entire range of operating conditions. This is achieved while maintaining the latency approximately constant at around 400ms. The latency rises slightly with the number of uploads due to the over provisioning a to accommodate stochastic fluctuations in service rate, .demonstrates the ability of the adaptive algorithm to respond quickly to changing network conditions. At time 200s the number of uploads is increased from 0 to 10 [11].

3. The Proposed Algorithm

The main objective in this research is to reduce a packet loss in FMIPv6 based on adaptive packet buffering tuning algorithm (APT) algorithm by using packet buffering tuning in scanning (probe) phase. This allows the MN to predict the potential handovers during its movement to obtain the right handover decision then the packet loss will be reduced. As the result of that, this research will focus on how to optimize the FMIPv6 functionality by apply APT algorithm. Thus, the packet loss happening during the handover process by use the buffering technique will be reduced.

3.1 Proposed APT Algorithm

In this section the design of the proposed adaptive packet buffering tuning algorithm based on priority and traffic throughput (APT) has been discussed.

The FMIPv6 supports packet buffering to reduce packet loss during handover. Nevertheless, when the MN moves to the congested access point (AP), the packets may be dropped during packet buffering at the new access point (nAP) because of finite buffer space applying simple packet buffering. More sophisticated packet buffering is required to support differentiated QoS for the packets during handover. Figure 4 shows the flowchart of adaptive packet buffering tuning technique based on priority and traffic throughput (APT).

In addition, high level of throughput and low delay can be achieved through the proposed technique. The buffer should not be left empty for a long period of time to ensure that the utilization of the link is efficient. The link idle time can be reduced through increasing the buffer size. However, shorter buffer ensures lower delays. Therefore, if the buffer is rarely left empty, the size should be decreased in order to achieve low delay. In contrast, should there be a long period of time where the buffer is left empty, the size should be increased to achieve high throughput.

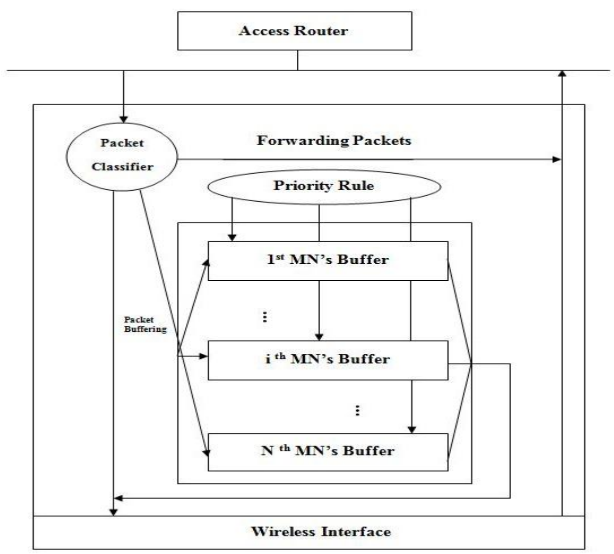

Fig. 3 The Structure of an AP with APT Algorithm.

Figure 3 shows the structure of an AP with the proposed APT algorithm. Here one buffer is assigned per mobile node (MN), rather than maintaining only one buffer per protocol connection to resolve the possible scalability problem. In general, a mobile node (MN) maintains several connections because several applications are supported. Simultaneously, one buffer is managed per connection; a mobile node (MN) has to manage more than one buffer. Managing one buffer per mobile node (MN) is simpler than one buffer per connection. Moreover, it is easy to support a new APT approach that is extension to ALT algorithm.

The proposed algorithm in this paper adaptive packet buffering tuning algorithm based on priority and traffic throughput as seen in Figure 4 in pink colour.

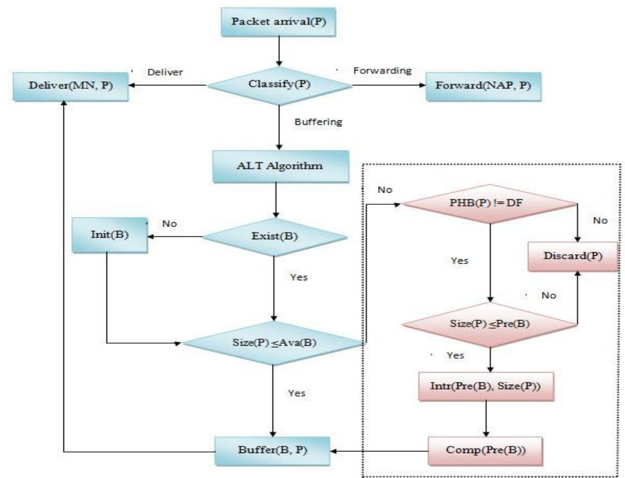

Fig. 4 The Flowchart of Adaptive Packet Buffering Algorithm APT.

This algorithm starts with step *Packet arrival(P)* is forwarded from the previous access point (PAP) and arrives at the buffer of the new access point (NAP). A *Classify(P)* classifies the incoming packet *P* into Deliver, Forwarding, or Buffering. Deliver means that the packet *P* is delivered to the mobile node (MN). *Deliver(MN, P)* denotes that the packet, *P*, is delivered to mobile node (MN). Forwarding means that the packet *P* has to be forwarded to a new access point (NAP) because the mobile node (MN) performs handover. *Forward(NAP, P)* denotes that the previous access point (PAP) forwards packet, *P*, to a new access point (NAP). Buffering means that the packet *P* forwarded from the previous access point (PAP) has to be buffered at a new access point (NAP).

Here between two steps (*Classify(P)* and *Exist(B)*) there is ALT algorithm. And the algorithm has some steps which are as following. Set the initial queue limit the (q) lies between the minimum and maximum values (q_{min} and q_{max}). The maximum buffer limit q_{max} set as to 400 packets, and the minimum buffer limit q_{min} set as to 30 packets.

Then, Set increase step size (*a*) means that increase the buffer size to be (*a*), and set decrease step size (*b*) means that decrease the buffer size to be (*b*). Where (*a*, *b*) are design parameters. Every time (*t*) second Measure the idle time (t_i) means every second gauge the idle time and busy time in the observation or update interval.

Respectively, calculate the value of new q (q_{new}) by $q_{new}=q+at_i-b(t-t_i)$ quotation. If t too large, will likely miss bursty changes, and if t too small, can therefore yields repeated or similar observations if the traffic patterns are varying slowly. Here choose the safer option $t=1$ second is used.

Next step is $q_{new}<q_{max}$ checks if the new buffer queue (q_{new}) is smaller than the maximum buffer limit (q_{max}). Otherwise, $q=q_{max}$ means that the value of q equals to value of the q_{max}. The $q_{new}<q_{min}$ step checks if the new buffer queue (q_{new}) is smaller than the minimum buffer limit (q_{min}), $q=q_{min}$ means the value of the q equals to value of the q_{min}. Otherwise, $q=q_{new}$ means that the value of q equals to value of the q_{new}.

After that, *Exist(B)* checks if the buffer space is available in the *MN*. Otherwise, *Init(B)* initiates buffer space, *B*, in the *MN*. *Size(P)* and *Ava(B)* calculates the size of packet and available buffer space, respectively. If the size of the arrival packet is smaller than the available buffer size, packet *P* is buffered in the buffer *(Buffer(B, P))*. Otherwise, the buffer is overflowed.

The steps following are in the pink colour, it means the APT approach. In the overflowed buffer, the packet is able to bypass lower priority packets previously buffered or be bypassed by higher priority packets according to the priority of PHB (Per-Hop-Behavior). PHB(P) identifies the PHB of packet, *P*. if the PHB of the packet is DF (Default Forwarding), it is preempted. Otherwise, *Discard(P)*, it means the packet *P* is discarded.

Pre(B) checks the available space which is able to be preempted by the packet. Then, the checked space is compared with that of the packet by using this step *Size(P)≤Pre(B)*. *Intr(Pre(B), Size(P))* interrupts the packets having lower priority than packet, *P*, as much as the size of it. It means that step checks if the packet has lower priority than packet, *P*, then that packet interrupted as much as the size of the packet, *P*.

Comp(B) compresses the buffer space, *B*. Empty space may exist between the packets because of interruption operation. *Comp(B)* is able to prevent any change of the sequence of the packets due to interruption operation. After this step, the packet *P* is buffered in the buffer *B* *(Buffer(B, P))*.

The APT approach can provide differentiated QoS for fast handover in the DiffServ domain by applying preemption rules according to the Per-Hop- Behavior (PHB) of the packets. Table 2 shows the rule adopted for the preemptive priority. Expedite Forwarding EF PHB has the highest priority since it has time and loss sensitive characteristics and the strict QoS requirements. Assured Forwarding AF PHB has elastic QoS requirements for maximum available QoS guarantee. Hence, the AF PHB has the second priority. Default Forwarding DF PHB does not support any QoS to the packets. Therefore, it has the lowest priority.

Table 2: The rule adopted for preemption

Priority	***(PHB)***	***Can interrupt others?***	***Can be interrupted?***
1	(EF)	YES	NO
2	(AF)	YES	YES
3	(DF)	NO	YES

3.2 Packet Loss Occurrence Possibility

This research analyzes the packet loss occurrence possibility of the proposed adaptive packet buffering tuning algorithm based on priority and traffic throughput. The proposed algorithm is modeled using an *M/M/c/c* queue. The parameters used in the model are as follows.

λ_i: Arrival rate of the ***i*** PHB packet
μ_i: Service rate of the ***i*** PHB packet
$\rho_i=(\lambda_i/\mu_i)$: Traffic load of ***i*** PHB packet, ***i*** =EF, AF, and DF.
c and ***k*** are parameters design for average service time.

The loss occurrence possibility of packets is calculated using the Erlang loss formula for *M/M/c/c* queue (Gross, et al., 2008). The EF PHB packet is never interrupted by AF or DF PHB packet. Therefore, loss occurrence possibility of EF PHB packet is as follows.

$$L_{EF} = \frac{\rho_{EF}^{c}/c!}{\sum_{k=0}^{c} \rho_{EF}^{k}/k!} \tag{4}$$

The EF and AF PHB packet are not interrupted by the DF PHB packet. Hence, the loss occurrence possibility of EF and AF PHB packet is as follows.

$$L_{EF,AF} = \frac{\rho_{EF,AF}^{c}/c!}{\sum_{k=0}^{c} \rho_{EF,AF}^{k}/k!} \tag{5}$$

Whereas $\rho_{EF,AF} = (\lambda_{EF} + \lambda_{AF})/(\mu_{EF} + \mu_{AF})$. Using Eq. (4) and (5), the L_{AF} value is calculated using the formula below.

$$L_{EF,AF} = \frac{\lambda_{EF}}{\lambda_{EF}+\lambda_{AF}} L_{EF} + \frac{\lambda_{AF}}{\lambda_{EF}+\lambda_{AF}} L_{AF} \quad (6)$$

The loss occurrence possibility of AF PHB packet is

$$L_{AF} = \frac{1}{\lambda_{AF}} \{(\lambda_{EF} + \lambda_{AF})L_{EF,AF} - \lambda_{EF}L_{EF}\} \quad (7)$$

Likewise, the loss occurrence possibility of DF PHB packet can be derived as follows.

$$L_{DF} = \frac{1}{\lambda_{DF}} \{(\lambda_{EF} + \lambda_{AF} + \lambda_{DF})L_{EF,AF,DF} - (\lambda_{EF} + \lambda_{AF})L_{EF,AF}\} \quad (8)$$

4. Performance Evaluation

We evaluate the performance of the proposed adaptive packet buffering algorithm (APT) by Omnet++ computer simulation. The network model adopted for simulation is illustrated in Figure 5.

4.1 Simulation Setup

We illustrate here the simulation settings that have been used to evaluate the efficiency of proposed APT algorithm to reduce packet loss in FMIPv6. Moreover, to evaluate the effects of different wireless interference levels on the various performance metrics such as packet loss, delay, and throughput all the experiments have be done under different transmission powers. Furthermore, to efficiency and capability of the proposed algorithm in this paper, it is to reduce packet loss in FMIPv6 as much as possible.

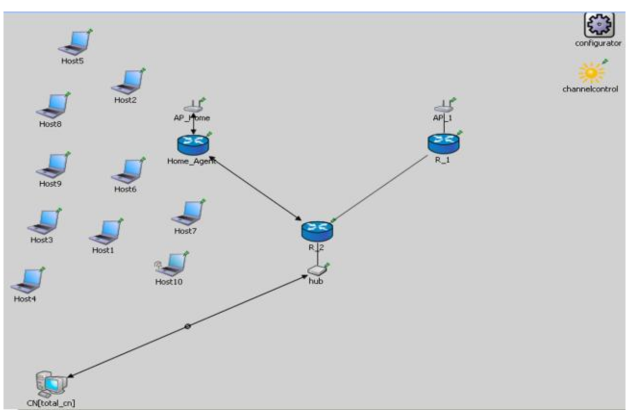

Fig. 5 The Network Model used for Simulation.

In the Figure 5 above, the mobility environment will be defined in this simulation as 700m x 500m with two Accesses Points (AP) when each AP run over IEEE 802.11/b standard uses 2.45 GHz, since this standard the coverage area for each AP will be as a 120 meters (300 feet) distance between each AP (largest market penetration of any WLAN standard, with commercial products available since 1999). Here all the mobile nodes (MNs) have Fast Mobile IPv6, and they are in the same subnet of the DiffServ (Differentiated Services) domain. The corresponding node (CN) transmits one packet in every 10ms to all the MNs.

Additionally, the PHB of each packet is randomly decided with the same probability. The packet size and the buffer queue size of the mobile node (MN) are set to be 512 bytes and from 400 packets to 30 packets, respectively. The proposed algorithm (APT) assumes that there is no packet loss due to transmission link errors. The bandwidth between routers is 11Mbps, and between a router and access point (AP) is 1Mbps. The interval speed in this network model is 1m/s to 2m/s. The propagation delay between the routers is 1ms and 3ms between a router and AP. To cause a buffer overflow, 10 MNs are assumed to move to the same new access point (NAP) at the same time. The simulation parameters in this research proposed algorithm and the values are as shown in the Table 3.

Table 3: Simulation Parameters and Values

No	***Simulation Parameters***	***Value***
1	Topology range	700mx500m
2	Number of Mobile Nodes	10 MNs
3	Interval Speed	1 to 2 m/sec
4	Number of Correspondent Nodes	1 CN
5	Number of Home Agents	1 HA
6	Number of routers	3
7	Number of Access Points	2 APs
8	Coverage Area for each AP	120m
9	Packet size	512 bytes
10	Buffer queue size	Between 30 to 400 packets
11	Bandwidth	11Mbps
12	Simulation time	700sec
13	Mobility Model	Random Waypoint Mobility Model (RWP)
14	Simulation tool	OMNET++

4.2 Simulation Results

In order to examine and evaluate the impact of proposed technique that it captured the traffic packet dropped which occurred when MN moved from one subnet to another one during simulation time. It can be observed from Figure 6 the proposed APT algorithm far superior than ALT algorithm in terms of ability to decreases the packet loss during the simulation scenario.

Figure 6 shows the traffic packet loss. With the proposed APT algorithm the packet loss started at 0.15 packets per second at the beginning and it constantly decreased to occur no packet loss during the simulation time (700 sec). However, with the ALT algorithm, the packet loss started nearly to 0.3 packets per second at the beginning and it gradually decreased until 0.03 at the time 700 sec. It reveals that the proposed APT algorithm effectively supports differentiated QoS for the packets according to PHB, and significantly reduces the packet loss rate compared with ALT algorithm. The improvement that could be obtained through using APT algorithm in terms of packet loss in comparison with the ALT algorithm is 49% improvement.

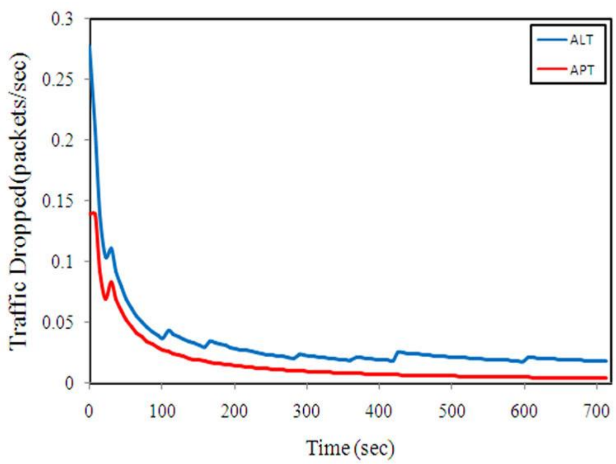

Fig. 6 Traffic Packet Loss during the simulation time for 700sec in ALT algorithm and APT algorithm.

Figure 7 shows the average of throughput in AP during the simulation time, and the average throughput in AP is increased in APT algorithm better than ALT algorithm. In both algorithms, from zero second until 100 seconds no throughput takes place. In APT algorithm, from time 100 sec it increased to reach its peak at 16000 bits per second, and then it dropped sharply at time 700 sec to reach 8000 bits per second. On the other hand, in ALT algorithm, in 100 sec the throughput increased to reach its peak at 9000 bits per second to be followed by a sharp drop at 6000 bits per second at time 700 sec. Additionally, in spite of the drop that took place from 16000 to 8000 bits per second, it is still considered that APT algorithm has high throughput compared to the ALT algorithm. The improvement that could be obtained through using APT algorithm in terms of average throughput in AP in comparison with the ALT algorithm is 50.86% improvement.

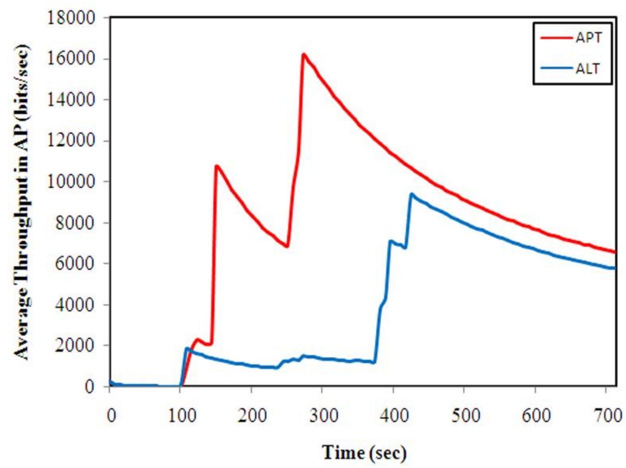

Fig. 7 Average Throughput in AP during the simulation time for 700sec in ALT algorithm and APT algorithm.

Figure 8 reveals the delay time in real-time traffic during the simulation time for 700sec in ALT algorithm and APT algorithm. In both algorithms, at the beginning, no delay took place in real-time application until second 100.The delay of APT algorithm shows fluctuation from second 110 to second 400. At that point, the delay time in real-time traffic increased slightly followed by a very slight drop almost at the second 410. From that time upwards, the delay of real-time traffic reached the highest value of delay at 0.4 ms, followed by a decreased until the second 700. However, at 100 sec, the real-time delay increased significantly at second 150 to reach 0.4 ms, followed by a slight drop until second 250 in ALT algorithm. At that time, the delay dramatically increased to be followed by a sharp drop at time 650sec; after that, the delay in real-time application increased significantly to reach its highest value of delay (0.6 ms) at second 700. As a result, the highest value of delay time in real-time traffic (0.4 ms) in APT algorithm was less than the highest value of delay time in real-time traffic (0.6 ms) in ALT algorithm. The improvement that could be obtained through using APT algorithm in terms of delay time in real time traffic in comparison with the ALT algorithm is 41.38% improvement.

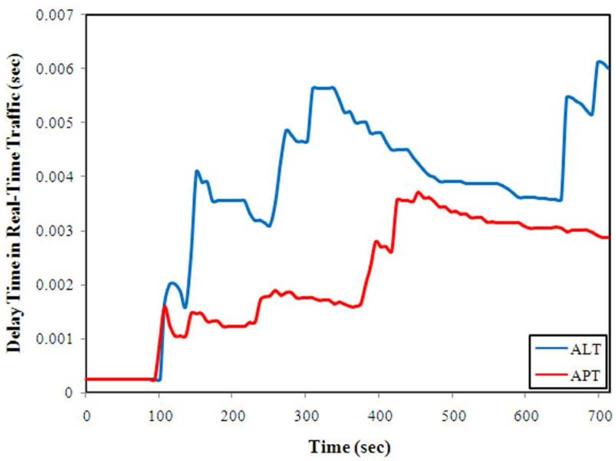

Fig. 8 Delay Time in Real-Time Traffic during the simulation time for 700sec in ALT algorithm and APT algorithm.

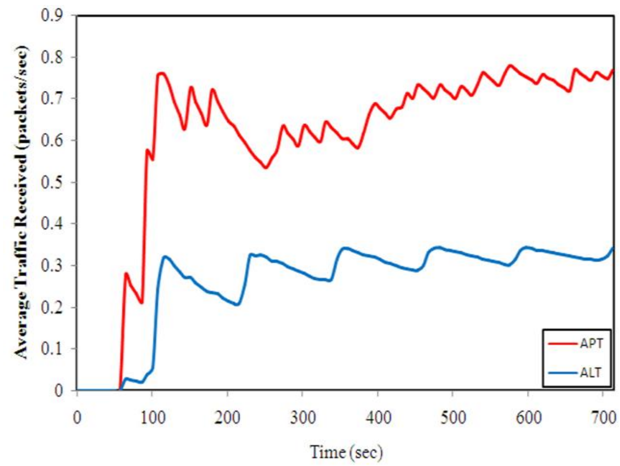

Fig. 9 The Average Traffic Received during the simulation time for 700sec in ALT algorithm and APT algorithm.

Figure 9 represents the average traffic received during the simulation time for 700sec in both algorithms ALT algorithm and APT algorithm. At the beginning, no traffic received took place until second 60 in both Algorithms. In this respect, the traffic received increased followed by a slight drop. After increasing significantly, the traffic received has reached to almost 0.6 packets per second at time 120 sec. As a result, the traffic received in APT algorithm increased followed by a number of fluctuations to stop at 0.8 packets per second at second 700. In contrast to APT algorithm, the traffic received in ALT algorithm shows a little increase and remained constant almost at 90 sec. After that, the traffic received increased sharply at 0.3 packets per second to be followed by a number of fluctuations to stop almost at 0.4 packets per second at second 700. To sum up, the traffic received in APT algorithm is better than the one received in ALT algorithm. The improvement that could be obtained through using APT algorithm in terms of average traffic received in comparison with the ALT algorithm is 56.96% improvement.

5. Conclusion and Future Work

In this paper we have proposed an adaptive packet buffering tuning algorithm APT to support differentiated QoS in the DiffServ domain. The forwarded packets are buffered by the defined rule during handover. We designed the structure of AP supporting the proposed adaptive packet buffering tuning algorithm APT. Moreover, we performed Omnet++ computer simulation to validate the performance of the adaptive packet buffering tuning algorithm APT. From the simulation results, we found that adaptive packet buffering tuning algorithm APT effectively supports differentiated QoS for the packets according to PHB, and significantly reduces the packet loss rate compared with ALT algorithm. In future work, it is recommended that the caching techniques at the Home Agent (HA) or Foreign Agent (FA) will be proposed in order to reduce the packet loss in wireless networks over IEEE 802.11/b standard. This proposed algorithm will be applied on different scenarios of mobility with various mobile densities.

References

[1] D. Johnson, C. Perikins, and J. Arkko, "Mobility support in IPv6, " IETF RFC 3775, June 2004.

[2] R. Koodli, "Mobile IPv6 Fast Handovers", IETF RFC 5568, July 2009.

[3] S. M. Zaki, "Reducing Real-Time Traffic Packet Loss in Mobile IPv6 Using two Tier Buffer", Master of Science (Computer Science), Faculty of Computer Science and Information Systems, University Technology Malaysia, 2008.

[4] S. Lee et al., "A Handoff Packet Marker for DiffServ in Mobile IPbased Networks with Packet Buffering", IEEE Trans. Consumer Electronics, vol. 55, no. 2, May 2009, pp. 438-446.

[5] S. Lee et al., "A Packet Forwarding Controller for Mobile IP-based Networks with Packet Buffering", IEEE Trans. Consumer Electronics, vol. 55, no. 3, Aug. 2009, pp. 1344-1350.
[6] K. Hur et al., "Priority forwarding for improving the TCP performance in mobile IP based networks with packet buffering", Computer Communications, vol. 30, no. 6, March 2007, pp. 1337-1349.
[7] W. Yao, and Y. Chen, "An Enhanced Buffer Management Scheme for Fast Handover Protocol", in Proc. 2004 IEEE Int. Conf. Distributed Computing Systems Workshop, Jan. 2004.
[8] H. Lu, X. Tie, P, Hong, "A Novel Buffer Mechanism for Fast Handovers in Mobile IPv6", in Proc. 2007 IEEE Int. Conf. Wireless and Optical Communications Networks, July 2007.
[9] Y. Zhao, and G. Nie, "A Scheme for Reducing Handover and Packet Loss in Heterogeneous Mobile IPv6 Networks", International Conference on Future Computer and Education, 2009.
[10] R. Caceres and V. Padmanabhan, "Fast and scalable handoffs for wireless networks", Proc. of ACM Mobicom '96, Nov. 1996, pp. 56-66
[11] T. Li, and D. J. Leith, "Adaptive Buffer Sizing for TCP Flows In 802.11e WLANs", IEEE Communications Letters, to appear, Irish Research Council for Science, Engineering and Technology and Science Foundation Ireland Grant 03/IN3/I396, 2008.
[12] R. Koodli, "Fast Handovers for Moblie IPv6", Internetdraft, 2007.
[13] X. Wu, and G. Nie, "Performance Analysis and Evaluation of Handover in Mobile IPv6", International Symposium on Intelligent Ubiquitous Computing and Education, 2009.
[14] D. Gross et al., Fundamentals of Queueing Theory, 4th ed. Wiley, 2008.
[15] M. Alnas, I. Awan, and R.D.W. Holton, "Performance Evaluation of Fast Handover in Mobile IPv6 Based on Link-Layer Information", The Journal of Systems and Software, 2010, pp. 1-7.