# Realization of Multiple Access Interface Management and Flow Mobility in IPv6

Umar Toseef, Asanga Udugama, Carmelita Goerg
Communication Networks Group
University of Bremen
28359 Bremen, Germany
+49 (421) 218-[ 8665 | 8665 | 2277 ]
[umr | adu | cg ]@comnets.uni-bremen.de

Changpeng Fan, Frank Pittmann
Nokia Siemens Networks
GmbH & Co. KG
13623 Berlin, Germany
+49 (30) 386-[36361 | 29387]
[changpeng.fan | frank.pittmann]@nsn.com

## ABSTRACT
Internet capable mobile or portable devices are already a modern commodity while it is becoming more and more common that such devices are hosts to more than one wireless network interface. The aim of this work is to show from a user's perspective how such a portable device may make best use of this property by using multiple wireless and wired network interfaces simultaneously. This would incline that the intelligent control logic can distribute active flows across the available network interfaces and that it is also able to seamlessly transfer them between the network interfaces in mid-session without interruption. Focus of this work is on the inclusion of user preferences in the decision process, recognizing that future telecommunication systems may include also network conditions and operator preferences.

## Categories and Subject Descriptors
C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design - *Network communications, Wireless communication*

C.2.2 [**Computer-Communication Networks**]: Network Protocols - *Protocol architecture (OSI model), Protocol verification.*

## General Terms
Management, Performance, Reliability, Experimentation, Security, Verification

## Keywords
Flow Management in IPv6, Optimized bandwidth resources usage, Traffic filtering, 3GPP, Non-3GPP.


*Mobilware'08,* February 12-15, 2008, Innsbruck, Austria.


## 1. INTRODUCTION
This work is carried out within the project ScaleNet (Scalable, efficient and flexible Network) that is partly sponsored by the German Ministry for Education and Research within the framework 'Networks of Tomorrow'. ScaleNet is the inter-working between 3GPP and non-3GPP networks considering mobility beyond terminal mobility and other network services while efficiently supporting applications and other services, hence, aiming at inter-access system service delivery in cooperation with efficient, optimized mobility management between the accesses.

As a part of ScaleNet activities, this paper details the realization of multiple access interfaces and support of flow mobility over these network interfaces. Multiple access interface realization provides mobile node with several network interfaces to connect to the Internet. However mobile node is restricted to use only one network interface to send and receive traffic from Internet and the rest of the interfaces act as backup for the mobile node to strengthen its connectivity to the Internet. Flow management functionality, as discussed in this work, takes the advantage of available multiple network interfaces and allows mobile node to direct its traffic flows to any of the available network interface. This results in management of available bandwidth resources to satisfy user and network operator's policies. Moreover, with the help of flow management services, a mobile node can dynamically shift its traffic flows from one network interface to another without interruption as its preferences change or some of the interfaces to the Internet are no more active and new interfaces are available for use.

This paper details the concept work of flow management, implementation and the analysis of results taken when using Mobile IPv6 as a mobility management protocol.

## 2. MULTIPLE ACCESS INTERFACE REALIZATION

### 2.1 Concept
As networking technology evolves, more and more different types of network access technologies become available. Each network access technology has its advantages and disadvantages that make it an attractive choice in a particular situation or scenario. For example IEEE technologies like WLAN (i.e., 802.11a/b/g) is

cheap and has high bandwidth but provides small area of coverage, limited security capabilities, no seamless mobility support and hence is suitable as bit pipe for only local (or static) use. On the other hand 3GPP technologies like UMTS or GPRS have a wider coverage area but due to their low bandwidth and high service cost (because of sophisticated network services like security and seamless mobility support) they are suitable for high speed outdoor use where WLAN is not available and is not an appropriate choice. Therefore it is a natural choice for a user to have hardware support for multiple access technologies so that it can connect to the Internet through any available or suitable network access technology. Moreover user will also like to keep its active connections alive while disconnecting from one access network and connecting to another network. This facility can be provided to the user by using a mobility protocol. Considering IEEE networks the IP protocol stack has also been upgraded (e.g. Mobile IPv6[1]) to provide mobility support so that a user can make seamless handover from one network to another. 3GPP standardization is also on its way to evolve the current network architecture[8] to support mobility between 3GPP and non-3GPP (e.g., IEEE networks) networks and may base on Mobile IP.

There is another aspect of having hardware support for multiple access technologies which is the user capability to connect to multiple access networks simultaneously. The chances to have multiple access network services available are pretty good in urban areas where coverage of different access technologies overlap and hence give a user the possibility to connect to Internet through more than one interfaces. Although currently available mobility protocols help users stay always connected by making handover from one access network to another, it is not clear what should be the behavior of a mobile device that is connected to multiple access networks. In standard Mobile IPv6, a user is not allowed to have multiple active network interfaces simultaneously. However there are recently some proposals in IETF working groups under discussion that deal with how mobility protocols can be modified to allow mobile devices to use multiple network interfaces simultaneously. This task is referred to as multiple care-of addresses (MCoA) registration in Mobile IPv6 terminology.

## 2.2 Operation

[3] is a draft from IETF MONAMI6[12] working group that proposes extensions in both Mobile IPv6 as well as in NEMO Basic Support Protocol [2] for MCoA registration. This extension propounds a new identification number called Binding Unique Identification (BID) number for each binding cache entry to accommodate multiple binding registrations. A unique BID number is assigned to each network interface or care-of address bound to a single home agent. Mobile node must send this BID number in each binding update message for the receiver to distinguish between bindings corresponding to the same home agent. In other words home address is used to identify a mobile node while BID number is used to identify multiple bindings (i.e. network interfaces) registered by the mobile node.

In order to transport BID and related information between mobile node and home agent/correspondent node an extension to mobility option header has been proposed. This sub-option is called BID sub-option and is included in binding update, binding acknowledgement and binding refresh request messages. It is also possible to associate a priority with each binding registration to specify user preferences about the use of active network interfaces as well as to determine the default binding. Default binding implies the care-of address that will be used to send and receive traffic at the mobile node. At any time instant a binding with the highest priority will be considered as the default binding.

Figure 1 shows a mobile node (MN) that is connected to two foreign access networks and therefore acquires two care-of addresses (CoA), one for each network interface. MN now decides priority and BID for each CoA and sends binding update message to its home agent (HA) with BID sub-option included in it. HA has extended its binding cache entry structure so that it can store multiple CoAs against one home address (HoA). So HA receives two binding updates and registers each CoA of MN against its home address (HoA) in its binding cache. When HA forwards traffic to the MN, it will check its binding cache entry for that MN and will pick up a CoA with the highest priority to use it as the destination address to send packets to MN.

In Figure 1 mobile node acquires two care-of addresses at its two interfaces one from each foreign network. However it is also possible for a node to acquire multiple care-of addresses over one network interface connected to a single physical network. An example of the latter case is a network where multiple prefixes are announced on the link to which mobile node is attached. In this case several global addresses will be configured on this network interface of the mobile node for each of the announced prefixes. In a conceptual sense there is no difference in the above two cases because the difference is only in the number of physical network interfaces. However BID number is used just to identify the binding. To simplify the situation BID numbers can be assigned to care-of addresses instead of physical network interfaces.

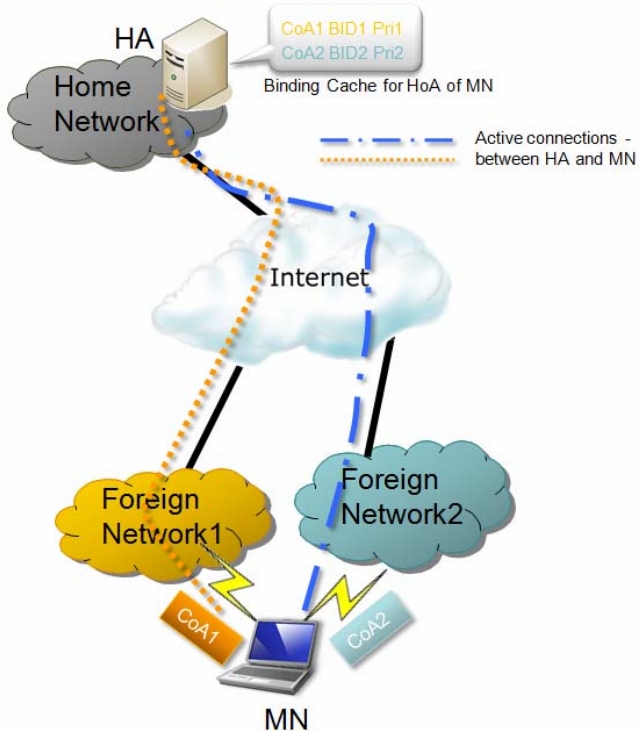


**Figure 1. A mobile node(MN) having active interfaces with two foreign networks, performs the multiple care-of addresses registration with its home agent(HA)**

# 3. FLOW MANAGEMENT

## 3.1 Definition

The term Flow Management (FM) is usually used in a wider sense where flow management can be considered within a network, between networks or even within a router. However in this research work flow management refers to how a stream of packets is directed through a certain path towards its destination. When a node connects to a network which is a part of Internet (e.g. a computer connected to its ISP) all of the network traffic that originates from that node will take its path towards its destination through the network it is connected to. Similarly all IP packets that are destined to the node will be forwarded to the node through that network and this is the only available way of routing that node's traffic. But for a node that is connected to Internet through two or more access technologies/networks (e.g. WLAN and Ethernet) and wants to make use of both access networks simultaneously, it will have to manage which type of traffic should take which network interface to be routed to the Internet. This is what is termed as flow management. In other words flow management is the way a node directs its certain traffic flows to certain available network interfaces. It is obvious that flow management is possible only if a node has multiple interfaces to the Internet and it can use some or all of them simultaneously.

Figure 2 shows how a node (which is connected to the Internet via WLAN and Ethernet) benefits from its resources. The file download is being conducted over high speed IEEE 802.3 network while user at the node is doing some web surfing over the second available connection via WLAN.

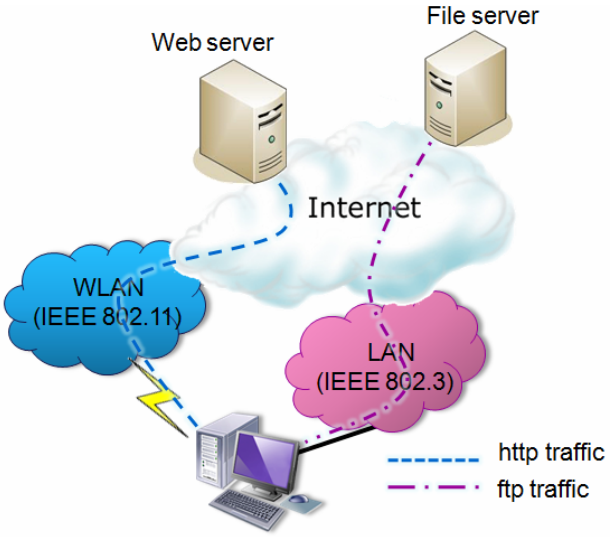


**Figure 2. Flow Management - Two traffic flows taking different paths to the destination node**

## 3.2 Flow Management in Mobile IPv6

In Mobile IPv6, flow management is based on the concept of utilizing multiple network attachments of a terminal (Mobile Node, MN) simultaneously. The most basic form of flow management is where the Home Agent (HA) or Mobility Anchor Point (MAP)[14], on the request of the MN decides to distribute flows to different network attachments of the MN. Figure 3 illustrates the distribution of two flows executed at the HA.

If a MN has multiple active network interfaces it registers them with its HA through the MCoA registration mechanism and also instructs its HA, by sending filter rules, how its traffic should be distributed over its registered CoAs. After a successful registration of CoAs and filter rules, HA now starts intercepting MN destined traffic in its home network and sorts out, using filter rules sent by MN, which traffic flow should be tunneled to which CoA of MN. Hence MN, in this way, can use all of its network attachments for receiving traffic as well as manage bandwidth resources by specifying which type of traffic it wants to receive over a certain network interface.

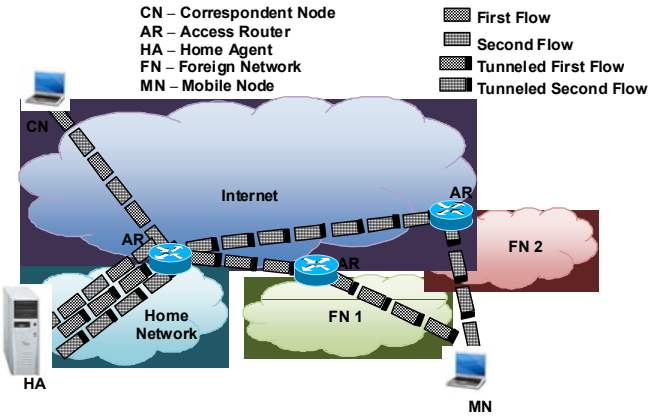


**Figure 3. Distribution of traffic flow over multiple network interfaces of mobile node through home agent in MIPv6**

Flow management can also be performed in route optimization mode of operation by sending filter rules directly to the correspondent node. In this case CN will be doing the filtering of its MN destined traffic.

## 3.3 Flow Management Options

A traffic flow is usually managed using one of the following options.

- Flow Distribution – MN can define different policies to distribute multiple flows over its active network interfaces. For example, first traffic flow is a Voice over IP (VoIP) application and has to be sent over Foreign Network #1 (FN1) because of the reason that it has better QoS than Foreign Network #2 (FN2). Second traffic flow is a FTP download, which does not have strict QoS demands and should be tunneled to MN through FN2.

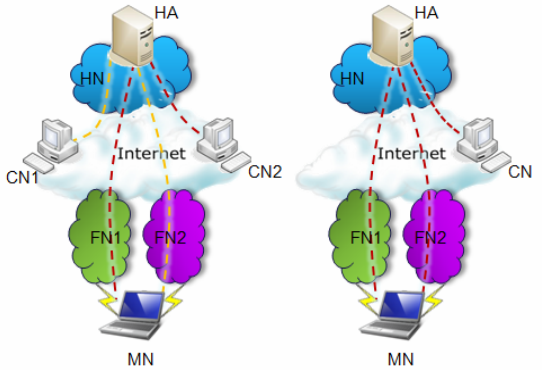


**Figure 4. Flow Management Options – Flow Distribution (left) and Flow Splitting (right)**

- **Flow Splitting** – MN can request HA/MAP/CN to split packets of one single traffic flow to different network attachments in order to speed up the transmission and aggregate the bandwidth. In this case, an application which has the capability to reorder packets at the receiver can perform well. For example, a FTP flow can be split over the attachments of FN1 and FN2.
- **N-Casting a Flow** – MN can request HA/MAP/CN to n-cast a certain traffic flow to its two or more global addresses (network attachments). So HA/MAP/CN would multicast the specified traffic flow to multiple global addresses of MN. This option would be helpful when MN has multiple connections of certain access technologies with high bit error rates. In this case, getting the same traffic flow over two or more connections will provide diversity to reduce overall transmission errors.
- **Dropping a Flow** – MN can request HA/MAP/CN to drop a particular traffic flow in some special situations. On receiving this flow drop request HA/MAP/CN will not forward any packet belonging to that traffic flow and will discard them. Flow dropping option can be helpful in situations where MN has very limited bandwidth resources and does not want to share it with low priority traffic flows.

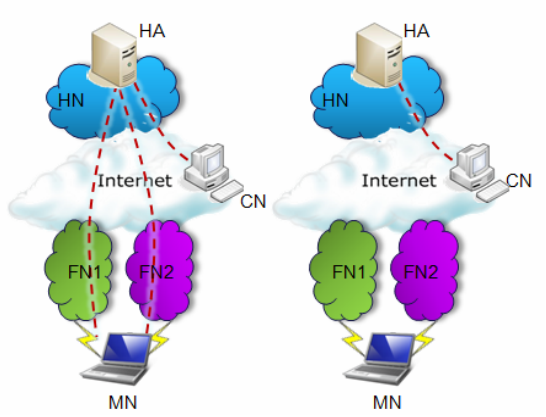


**Figure 5. Flow Management Options – n-Casting (left) and Flow Dropping (right)**

## 3.4 Types of Traffic Filtering

Flow management can be performed for downlink traffic flows (called as forward filtering) as well as for uplink traffic flows (called as reverse filtering) of MN. The need to distinguish uplink and downlink traffic flows comes from the fact that application's QoS requirements might be different for uplink and downlink traffic flows. Moreover MN might also have asymmetric connections to networks with different uplink and downlink bandwidths and hence wants to manage uplink and downlink traffic quite differently.

- **Forward Filtering** – Forward filtering means to do flow management for downlink traffic of MN (i.e. MN destined traffic). In forward filtering, different traffic flows are usually identified by transport protocol type, source address and source/destination port numbers. Forward filtering of the traffic is achieved by setting filters at HA, MAP or CN. Forward filtering plays a very important role for a MN having limited downlink bandwidth and running applications that are competing for downlink data rate and therefore is in need of better bandwidth resource management.
- **Reverse Filtering** – Although forward filtering significantly improves the efficiency of use of available bandwidth resources and provides a tool to mange incoming traffic flows MN should also be able to manage its outgoing traffic flows in order to take full advantage of flow management. This traffic flow management is called reverse filtering. In order to perform reverse filtering the filters are applied at MN.

The work presented in this paper deals only with forward filtering.

## 3.5 Flow Management Protocol

### *3.5.1 IETF Proposals*

Network traffic flow management is not a novel idea. Because of its need and importance it has been in discussion since long. There exist many IETF proposals as well as implementations of flow management for various protocols. Still, with the passage of time as different types of networks become available with high QoS demanding Internet applications, flow management is becoming a more and more hot topic. Especially, there is a need to integrate the FM functionality in newly introduced Mobile IPv6 protocol because Mobile IPv6 is becoming a widely accepted protocol for mobility. For example within 3GPP (3rd Generation Partnership Project) to provide mobility between 3GPP and non-3GPP access networks Mobile IPv6 is one of the available choices.

MONAMI[12] is an IETF working group where different proposals related to multi-homing and flow management are being discussed. Specifically related to flow management there are two active proposals under discussion.

i. draft-soliman-monami6-flow-binding-04[4]
ii. draft-larsson-monami6-filter-rules-02[5]

The work in this paper considers the first draft [4] for implementation and evaluation due to following reasons.

- This draft has an integrated filter rules exchange mechanism instead of having a separate protocol for this purpose.
- Filter rules exchange mechanism benefits from inherent standard Mobile IPv6 security measures.
- It is a simple protocol and hence is easy to implement and integrate with existing MCoA registration capable NEMO software.
- It has been under discussion in MONAMI6 WG since long and therefore it is more mature than its counterpart drafts.

### *3.5.2 Operation*

This section briefly describes the operation of the flow management protocol proposed by [4]. This protocol introduces a new mobility option for Mobile IPv6 mobility header. This mobility option is named as "flow identification" option and is used to establish flow bindings between MN and HA/CN. Just like a regular binding which is used to inform receiver about the current location of MN, flow binding is used to send filter rules to the other end. These flow bindings can be refreshed, removed and also get expired. A flow binding is identified by a unique integer number (referred to as FID) and is always associated with a

certain CoA. Therefore a flow binding message is usually piggy-backed on its associated CoA's binding message.

When a MN wants to send a filter rule to its HA, it construct a "flow identification" option by describing the traffic flow using valid flow identifiers (e.g. source address, destination address, source port number, destination port number, protocol name), specifying the filtering action (e.g. forward, drop, n-cast or split this traffic flow) and identification number (FID) for this flow binding. This "flow identification" option is then attached to the binding message of the associated CoA and is sent to HA. A future refresh request for this flow binding will always be included in its associated CoA's binding update message. A flow binding can also be removed or replaced by other flow bindings at any time by sending a request for that operation. All other traffic flows that do not come under any filter rule description will take their path to MN through the default binding's CoA.

Let's take the previous example where a MN has two active connections to its HA through two foreign networks. This MN has registered its both CoAs (CoA1 and CoA2) to its HA and now wants to receive FTP traffic at CoA1 and HTTP traffic at its CoA2. For this purpose, MN will have to construct a "flow identification" option in which it will describe the FTP traffic flow and will attach it to the binding update message for CoA1 to send it to HA. Another "flow identification" option will also be constructed by MN where it will describe the HTTP traffic and will attach it to binding update message for CoA2 to send it to HA. After the successful registration of these two flow bindings, FTP traffic will be tunnelled to MN at its CoA1 and HTTP traffic will be tunnelled to MN at its CoA2 by the HA.

### *3.5.3 Implementation*

As discussed earlier, flow management is only possible if a user has multiple active network interfaces which mean that implementation of multiple care-of addresses (MCoA) registration mechanism is a prerequisite for flow management implementation. MCoA registration implementation for NEMO is already available from Nautilus6 [6] under WIDE [7] project with particulars listed in Table 1.

**Table 1. Nautilus6's MCoA registration capable NEMO software details**

| Item | Details |
|---|---|
| Implemented draft proposal | [2],draft-ietf-monami6-multiplecoa-01[3] |
| Operating System Name | Linux |
| Operating System Version | Kernel 2.6.16 |
| License | Open Source (GNU) |
| Software Components | i. Kernel patch<br>ii. User land software |
| Implementation Status | Main features have been implemented while others are in progress. |
| Available Software Version | nepl-0.2-mcoa-beta3-20070118 |

In order to expedite the implementation process as well as to take the advantage of already done work, Nautilus6's NEMO implementation has been used as a basis for our flow management implementation. Therefore available source code of MCoA registration capable NEMO implementation has been modified and extended to incorporate flow management functionality as proposed by [4]. This process resulted in a NEMO implementation which is MCoA registration capable as well as has flow management functionality incorporated in it.

## 4. Performance Tests

This section discusses some performance tests and their outcomes to show how flow management can be helpful in managing different traffic flows. These tests actually emulate real world scenarios and hence give a better understanding of the working and usage of flow management. Figure 6 shows network topology of the test-bed that has been used in carrying out performance tests. There exist a Home Network (HN) and a Home Agent (HA) together with 2 Foreign Networks (FN). First foreign network (FN1) provides connectivity through WLAN while the second foreign network (FN2) is accessible through an Ethernet cable. Both of the foreign networks are connected to each other and the home network through a router. Correspondent Node (CN) is also connected to the router and has access to home network nodes.

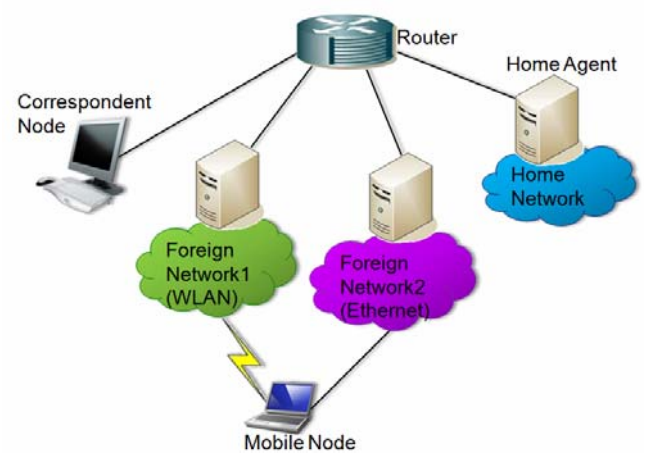


**Figure 6. Testbed setup for performance tests execution**

Table 2 gives the details of hardware and software configurations of machines that have been used as MN, HA, CN and foreign routers (FR) in foreign networks. NEMO software at HA and MN has been patched to support flow management. This patch has been developed as a result of flow management implementation as discussed in previous section.

**Table 2. Hardware configuration of testbed machines**

| Hardware | Specification | |
|---|---|---|
| | CN,FR and HA | MN |
| Processor | AMD® Athlon® 64 Processor 3700+ | Intel® Centrino® M 1.8 GHz |
| RAM | 1 GB | 512 MB |
| Ethernet | 100 Mbps Fast Ethernet | |
| WLAN | 802.11b 11Mbps | |

## 4.1 Performance Test Scenarios

In order to evaluate the performance of the FM implementation, the following two test scenarios will be presented.

1. Test of shifting traffic flows

i. UDP traffic
ii. TCP traffic

2. Test of dropping a traffic flow

In all test scenarios MN will have two care-of addresses from two foreign networks and will register both of these care-of addresses with its HA (refer to Figure 6). Care-of address from FN1 (i.e. WLAN interface) will be registered as the default care-of address. CN will always be originating the traffic flows destined to MN's home address (HoA). These traffic flows will be intercepted by HA and will be forwarded to at its default care-of address unless there are some filter rules set by MN for those traffic flows.

CN uses Iperf[11] to generate the required traffic flow. All packets generated by Iperf carry sequence numbers so that statistics (e.g. throughput, packet loss rate, etc.) can be generated for each traffic flow. On MN side, packets of traffic flows are captured by ethereal[10] and are analyzed to generate different graphs to show throughput and other parameters.

### *4.1.1 Test of Shifting Traffic Flows*

This performance test scenario will be executed on UDP and TCP traffic flows separately. In this test two traffic flows will be issued by CN with destination address as MN's home address. In the beginning there is no filter set by MN at HA so both traffic flows are forwarded to MN at its default care-of address (i.e. through WLAN). After some time MN sends flow binding to HA in order to set a filter which shifts one of the flows to the second care-of address (i.e. through Ethernet). These traffic flows are monitored for a sufficient period of time and graphs are generated.

#### *4.1.1.1 Shifting a UDP Traffic Flow*

This particular scenario emulates a real world scenario where MN receives TV channel traffic from some multimedia server consisting of audio and video UDP streams. MN receives both of the traffic flows at its default care-of address i.e. through WLAN. But due to high bandwidth of these traffic flows and background traffic in WLAN a high packet loss rate is observed which results in quality deterioration of audio as well as of video. The user of MN now decides to shift either video or audio traffic flow to another high speed network interface e.g. 3GPP radio access link. This shifting of flow gives a relief to both of the flows and the user at MN can enjoy his favorite video program. 3GPP radio access has been emulated here with fast Ethernet. Audio and video streams have been emulated with 4Mbps UDP streams.

Without having a look over the throughput graph of this test, it is expected that in the beginning when both flows follow WLAN to reach MN there will be a decrease in throughput of both traffic flows. It is because WLAN cannot support 8Mbps throughput at transport layer and will drop many packets from both of the flows. When flow binding will be sent to HA to set a filter in order to shift one of the traffic flows to Ethernet, both of the traffic flows will have the required bandwidth and hence the throughput will increase to 4Mbps for each flow.

Figure 7 shows the throughput graph of the two UDP traffic flows that are forwarded to MN through WLAN interface and afterwards one of the flows is shifted to Ethernet interface. It can be seen that, as expected, two 4Mbps UDP traffic flows compete for the available bandwidth in WLAN which results in many packet drops of each traffic flow and hence reduction in the throughput of each traffic flow. At about 40 sec on time axis, a Binding Update (BU) is seen which carries the flow binding to set a filter at home agent so that one of the traffic flows can be shifted to other available network interface (i.e. Ethernet). When this filter is set at home agent, one of the traffic flows is shifted immediately to Ethernet interface while the other traffic flow continues using WLAN to reach MN. Afterwards both traffic flows enjoy sufficient available bandwidth in their respective foreign networks and packet loss reduces to almost zero.

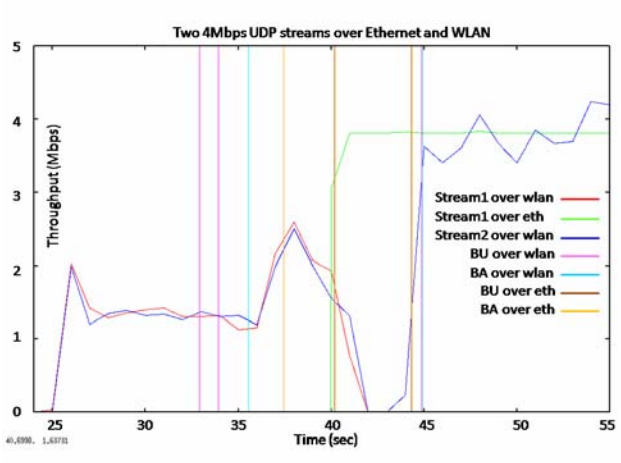


**Figure 7. Shifting of UDP flow**

A careful look at above graph shows that there are two unexpected things in the graph. Firstly, it shows a shifting of the traffic flow even before the BU is actually sent. Secondly throughput of the un-shifted traffic flow reduces to zero between 40 sec and 45 sec marks on time axis for about 1.5 sec. Both of these anomalies have actually their root in one single problem. This problem is somehow an implementation related issue where one of the "IPv6 over IPv6" tunnels which is used to tunnel MN destined traffic from HA to MN gets dropped for some time and then gets established automatically after a few seconds. This dropping of tunnel problem occurs quite randomly and seems to have no relationship with Mobile IPv6 signaling.

When "IPv6 over IPv6" tunnel of WLAN interface gets dropped, HA starts using Ethernet's "IPv6 over IPv6" tunnel to send traffic to MN. And after some time when "IPv6 over IPv6" tunnel gets established, HA starts using this default tunnel unless there have been some filter rules set by MN. The throughput graph lines in Figure 7 show the activity of a certain traffic flow over the given network interface. That is, if a traffic flow is monitored over the Ethernet interface then all packets except those which belong to the target flow are ignored. That's the reason for the time period when "IPv6 over IPv6" tunnel of WLAN interface drops and both traffic flows are sent over the Ethernet interface, only the target traffic flow is monitored (that can be seen at the time just before the BU) and the other traffic flow is ignored (that results in the zero throughput of that traffic flow shown in the graph). Figure 8 that shows number of lost packets during this test confirms the above explanation.

This implementation bug has been reported to NEMO software developers at Nautilus6.

Another interesting behavior to be noticed in the Figure 7 graph is that when one traffic flow is shifted from WLAN to Ethernet throughput of this traffic flow reduces to zero gradually rather than abruptly. The reason for this behavior is that queue buffer at

the WLAN access point is very much full with the packets from both of the traffic flows and even though there are no more new packets of the shifted traffic flow, MN continue receiving the buffered packets through the WLAN interface.

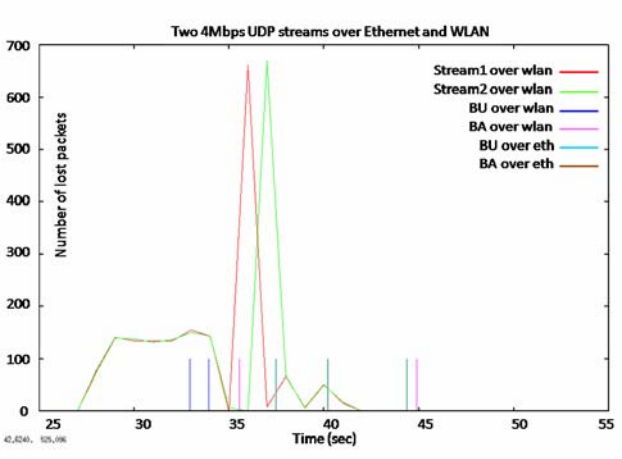


**Figure 8. Graph of number of lost packets in shifting of UDP flow scenario**

### *4.1.1.2 Shifting a TCP Traffic Flow*

This particular scenario where one of the two TCP traffic flows is shifted to another available network interface, emulates the real world scenario in which user is receiving two TCP traffic flows; one for email downloads and second for an FTP download. In the beginning, user receives both of these TCP traffic flows over WLAN interface which is the default network interface but afterwards due to limited bandwidth available in WLAN user decides to shift his FTP download flow to high bandwidth 3GPP radio access. This action will let both of the TCP traffic flows to use available bandwidth in their respective foreign networks. Since 3GGP radio access network can support high throughput, the user will notice that his FTP download is now much faster than it was before as well as his email download is also a bit faster than it was before.

Email download has been emulated here with TCP flow1 and FTP download has been emulated with TCP flow2. 3GGP radio access network has been emulated with fast Ethernet.

If the result of this test scenario is anticipated, it is expected that in the beginning when both of the traffic flows will be following WLAN interface to reach MN there will be a competition between the two traffic flows to get more and more available bandwidth. And after setting the filter at HA to shift one of the traffic flows to Ethernet interface both of the TCP flows will be using maximum available bandwidth in their respective foreign networks.

Figure 9 shows the throughput graph of this test scenario. It can be seen that both of TCP traffic flows take a slow start and then reach the maximum available WLAN bandwidth which is approximately 3Mbps for each traffic flow. After some time when BU is sent to HA, with flow binding piggy-backed on it, to set a filter which shifts one of the TCP traffic flows to Ethernet interface, throughput of both of the traffic flows increases sharply.

### *4.1.2 Test of Dropping a Traffic Flow*

This is the second performance test where dropping of a traffic flow is tested. This test can be performed over both UDP and TCP traffic flows but in this section only UDP will be used for testing purposes. This test scenario emulates the real world scenario in which a user has only one WLAN interface available and he is receiving audio and video streams of a NEWS TV channel over WLAN interface. As there is not enough bandwidth available in WLAN to support both audio and video streams, a lot of packets are dropped due to congestion in the network and user is left with bad quality video as well as that of the audio. The user now decides to drop the video streams so that audio stream can take its required bandwidth which results in good quality audio. Audio and video streams have been emulated here with 4Mbps UDP streams. In this test scenario MN will use only WLAN interface and Ethernet interface will be kept disabled throughout this test.

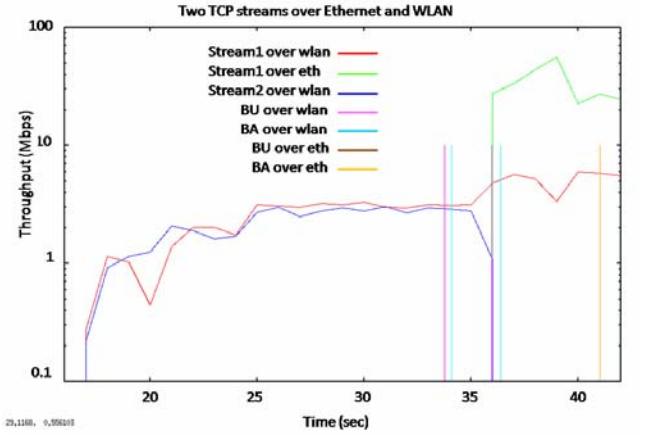


**Figure 9. Shifting of a TCP traffic flow**

The expected result of this test is that the two UDP streams will be competing, in the beginning, for the available bandwidth in WLAN and due to not enough bandwidth available the throughput of both of the traffic flows will be reduced due to packet losses. And as soon as one of the traffic flows is dropped, the other traffic flow will start enjoying the whole available bandwidth and hence throughput of the traffic flow will reach its expected value.

Figure 10 shows the throughput graph obtained when this test is performed. The actual result is quite similar to the expected one. The two UDP traffic flows when sharing the WLAN bandwidth have lower throughput than their actual 4Mbps throughput but as soon as a BU is sent to HA carrying flow binding to drop one of the traffic flows, the other traffic flow achieves its 4Mbps throughput.

It can be seen in the throughput graph shown in Figure 10 that after dropping one traffic flow, the throughput of the other traffic flow sometimes shoots above 4Mbps and sometimes below 4Mbps. The reason for this behavior is related to queue management of the operating system and that of the WLAN access point. The queue buffer that overflows due to 8Mbps UDP traffic flows, after dropping one traffic flows, starts releasing the buffered packets which results in throughput overshoots. But once queue buffer is settled with single 4Mbps traffic flow it can be seen that throughput of the traffic flow becomes stable at about 4Mbps.

The results of these performance tests are quite encouraging and imply that FM can be incorporated into NEMO Basic Support protocol as well as Mobile IPv6 without any problems. The real world scenarios emulated in these performance tests confirm that FM can be very useful for MN in several ways. Moreover FM can perform equally well for TCP as well as for UDP traffic flows.

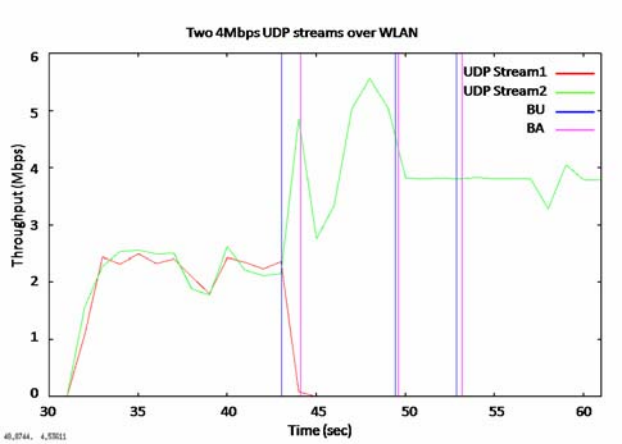


**Figure 10. Throughput graph when a traffic flow is dropped**

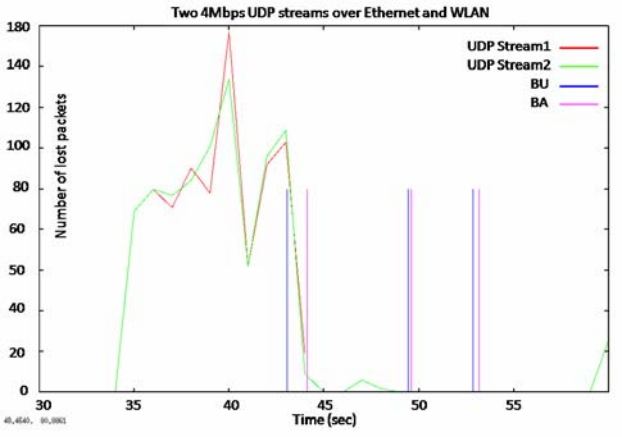


**Figure 11. Number of lost packets in traffic flow drop scenario**

## 5. Conclusions

This research work has been devoted to the study of traffic flow management, its implementation in NEMO[2] and performance analysis.

The paper presented a brief introduction about multiple access interfaces realization and flow management and how these functionalities can be incorporated in MobileIPv6. In order to take advantage of the available research work in FM area, a state-of-the-art was presented where proposed drafts from IETF MONAMI6 WG[12] were considered. One of these proposals, draft-soliman-monami6-flow-binding-04.txt[4], was implemented by extending an available implementation of NEMO from Nautilus6.

In order to test the implementation and its integration with existing NEMO software several performance tests were carried out. These performance tests were emulating real world scenarios and their results confirmed that with the help of flow management mobile node can manage its traffic flows quite efficiently.

Flow management helps users to make efficient use of available bandwidth resources, take advantage of diversity, prioritize his flows and do load balancing, control surfing costs as well as choose secure paths for sensitive traffic flows. For a network operator, flow management can not only potentially enhance value-add of offered services but also provide an additional network management tool. Flow management in Mobile IPv6 research and the implementation developed as part of the ScaleNet project has been used in the ScaleNet demonstrator with NETCAPE which was shown at the CeBIT exhibition, held in March 2007 in Hannover.

## 6. REFERENCES


[1] Johnson, D., Perkins, C., and J. Arkko, *Mobility Support in IPv6*, RFC 3775, June 2004

[2] Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert, *Network Mobility (NEMO) Basic Support Protocol*, RFC 3963, January 2005.

[3] Wakikawa, R., *Multiple Care-of Addresses Registration*, Technical Report Internet Draft, draft-ietf-monami6-multiplecoa-03, Work in Progress, IETF, July 2007.

[4] H. Soliman, N. Montavont, N. Fikouras, K. Kuladinithi, *Flow Bindings in Mobile IPv6 and Nemo Basic Support*, Technical Report Internet Draft, draft-soliman-monami6-flow-binding-04, Work in Progress, February 2007

[5] C. Larsson, H. Levkowetz, H. Mahkonen, T. Kauppinen, *A Filter Rule Mechanism for Multi-access Mobile IPv6*, Technical Report Internet Draft, draft-larsson-monami6-filter-rules-02, Work in Progress, March 2007

[6] NEPL (NEMO Platform for Linux) http://www.nautilus6.org

[7] WIDE (Widely Integrated Distributed Environment) http://www.wide.ad.jp

[8] 3GPP System Architecture Evolution: Report on Technical Options and Conclusions (Release 7), 3GPP TR 23.882 V1.9.0 (2007-03)

[9] Fikouras, Nikolaos Albertos, *Performance Evaluation and Improvement of the Mobile Internet Protocol: A Study of Hand-offs, Transport Layer Performance and Flow Mobility*, Ph.D. Thesis, University of Bremen, Germany, 2007.

[10] Gerald Combs. *Ethereal Network Protocol Analyzer*. Version 0.10.14. http://www.ethernet.com

[11] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, K. Gibbs, *Iperf – The TCP/UDP Bandwidth Measurement Tool*. Version 1.7.0. http://dast.nlanr.net/Projects/Iperf

[12] MONAMI6 (Mobile Nodes and Multiple Interfaces in IPv6) IETF WG,

[13] Soliman Hesham. *Mobile IPv6: Mobility in a Wireless* Internet. Addison-Wesley Professional, 2004

[14] H. Soliman, C. Castelluccia, K. El Malki, L. Bellier, *Hierarchical Mobile IPv6 Mobility Management (HMIPv6)*, RFC 4140, August 2005