Towards an Architecture to Support Complex Multihomed Mobility Scenarios

Ibrahim S. Alsukayti
School of Computing and Communication
Lancaster University
Lancaster, United Kingdom
i.alsukayti@lancaster.ac.uk

Christopher Edwards
School of Computing and Communication
Lancaster University
Lancaster, United Kingdom
c.edwards@lancaster.ac.uk

ABSTRACT

In this paper, we present the Multihomed Mobile Network Architecture (MMNA), a comprehensive multihomed mobility solution for complex nested mobility scenarios. It provides a multihoming management mechanism for gateway discovery and selection, on top of an efficient multihomed mobility model integrating different mobility and multihoming protocols. We describe how the MMNA was experimentally implemented and evaluated in a testbed setup. We first validated the capabilities of the solution in terms of different multihoming features, namely load sharing, link failure recovery, and preference setting. We then examined the effectiveness and feasibility of the MMNA solution considering a use case example of a search and rescue scenario. The results highlight the practicality and advantages of deploying the MMNA solution into realistic scenarios.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Measurement, Design, Experimentation, Performance

Keywords

NEMO; MANEMO; Mobility; Multihoming

1. INTRODUCTION

IP mobility ensures network reachability and session continuity while IPv6 nodes are on the move. This is not only applicable to individual roaming IPv6 hosts but also to those interconnected into a mobile IPv6 network which can be formed in different environments. Personal Area Networks (PAN) and Internet access on public transport are examples of real-world use cases. However, the potential for theses mobile networks to interconnect allows the formation of more complex mobile network topologies. This model allows Internet access provided by a designated gateway to be extended in a remote area in scenarios such as public safety. With the span of wireless Internet access technologies such as Wi-Fi, 3G, and WiMax and the popularity of multi-interfaced wireless devices, more Internet access options can become available in these sorts of scenarios, but remain idle. In scenarios such as search & rescue, proliferating Internet connectivity with more Mobile Routers carried by in-field members and sharing direct Internet access would enhance the performance of their communication and have positive impact on their missions. Therefore, efficient multihoming support in such complex mobility scenarios becomes critical. A multihomed scenario would enable advanced mechanisms such as load sharing, traffic engineering, and failover recovery. Eventually, this would allow better network performance and optimal utilisation of available resources. In this paper, we address the need for efficient multihoming support in the context of nested mobility scenarios. We present the Multihomed Mobile Network Architecture (MMNA), a comprehensive multihomed mobility solution.

2. BACKGROUND

2.1 Network Mobility

NEMO Basic Support (NEMO BS) [1] provides a roaming Mobile Network of a group of nodes, referred to as Mobile Network Nodes (MNNs), with mobility support managed by its Mobile Router (MR). Once the MR connects and configures a Care-of-Address (CoA), it performs the Binding Update process with its Home Agent (HA). The HA then installs a binding between the CoA, home address, and the Mobile Network Prefix (MNP) of the MR. Upon a successful binding update, a bi-directional tunnel is established between them. The reachability of the Mobile Network is then maintained over the tunnel, transparently to the communication of its MMNs and a Correspondent Node (CN).

In the NEMO model, a remote MR can connect to the mobile subnet of another MR and gains indirect Internet access. Once that happened, the remote MR registers and establishes a tunnel with its HA over the existing tunnel of the MR to which it is connecting. The chain can also extend resulting in topological structure known as Nested NEMO. With this model, the communication of the MR needs to traverse a multi-tunnels path. This routing sub-optimality of Nested NEMO is known as the Pinball Routing problem [2].

2.2 MANET for NEMO (MANEMO)

The concept of MANET for NEMO (MANEMO) is based on combining the properties of the Mobile Adhoc Network (MANET) and NEMO technologies [3]. It defines two different models, the NEMO-Centric MANEMO (NCM) model, addressing the Nested mobility issues, and the MANET-Centric MANEMO (MCM) model, addressing mobility support for MANET. An example of a comprehensive MANEMO-based solution is the Unified MANEMO Architecture (UMA) [4].

The NCM model provides a Route Optimisation solution for the Nested NEMO scenario. It is based on enabling a MANET-like routing model within the nested infrastructure to allow only a single tunnelling layer via its gateway-MR. Using the Tree Discovery (TD) protocol, interconnected MRs form a tree-based structure and establish default routes towards the gateway MR. The Network In Node Advertisement (NINA) protocol is also used to exchanged routing information over the tree, by each MR advertising its MNP up the tree. The Binding Update process is then performed over the tree infrastructure. Upon a successful home registration, the existing tunnel of the gateway is utilised for the MRs communication. If the gateway and a MR within the tree belong to the same HA, the scenario is called the Aggregated Roaming Scenario. Otherwise, the binding process is performed as a Non-Aggregated Roaming Scenario in which the gateway's HA becomes a Proxy-HA and carries out the MR binding process to establish a tunnel with the Target-HA.

2.3 Multihomed Mobility

NEMO BS and Mobile IPv6 do not have any multihoming support. However, they were extended with the Multiple CoA Registration (MCoA) protocol [5] enabling a multi-interfaced MR to register multiple CoAs and establish multiple tunnels with its HA. Each CoA is assigned a unique Binding Identifier (BID), which is then used to identify the different bindings of the MR. The MCoA protocol enables the maintenance of multiple communication paths over the multiple tunnels, without defining how the traffic is forwarded among them.

3. MMNA DESIGN

The Multihomed Mobile Network Architecture (MMNA) is a comprehensive multihoming solution for nested mobility scenarios. It enables the establishment of a multihomed mobile tree of heterogeneous Internet access, and provides an efficient solution for multihoming management. Figure 1 presents an architectural overview of the MMNA design. It shows the two main MMNA processes, namely Multihomed Tree Establishment and Gateway Discovery and Selection. This section describes these main components.

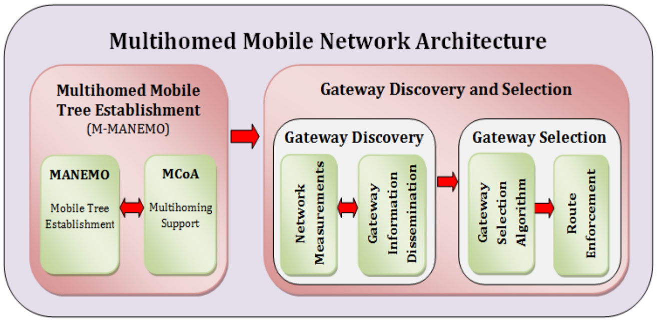
Multihomed Mobile Network Architecture
Multihomed Mobile Tree Establishment
(M-MANEMO)
MANEMO
Mobile Tree Establishment
MCoA
Multihoming Support
Gateway Discovery and Selection
Gateway Discovery
Network Measurements
Gateway Information Dissemination
Gateway Selection
Gateway Selection Algorithm
Route Enforcement

Figure 1. MMNA Design Overview

3.1 Multihomed Mobile Tree Establishment

This process enables the establishment of a multihomed mobile tree with multiple gateways spanning across the tree. To achieve this, we extended the MANEMO architecture by integrating the MCoA protocol to enable efficient support of nested mobility and multihoming. We called this collection the Multihomed-MANEMO (M-MANEMO) protocol. Adopting MANEMO enabled the establishment of an optimised tree-based routing model using the TD and NINA protocols in addition to performing an enhanced home binding process. The MCoA protocol provides the multihoming functionality supporting the emergence of additional Gateways within the tree. Figure 2 shows a simple M-MANEMO tree. M-MANEMO also enables trees convergence whereby one tree can join another tree over an additional egress interface of its gateway. For efficient tunnel management, each tunnel in a M-MANEMO tree is assigned a unique identifier called a Tunnel ID (TID).

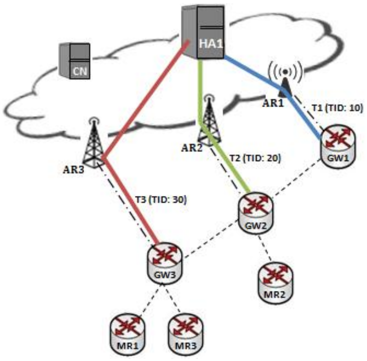

Figure 2. M-MANEMO Tree

3.2 Gateway Discovery

The process of Gateway Discovery enables the MRs in a M-MANEMO tree to discover and learn the capabilities and performance of the available Gateways within the tree. This would enable the nodes to make informed decisions when selecting the optimal Gateway to access the Internet. We developed a Gateway Discovery Protocol (GDP) defining how gateway information is conveyed, propagated, and collected within the tree. We also backed this up with the Network Measurement process to enable collection of IP performance and capabilities metrics describing each Internet access option available within a M-MANEMO tree.

Considering the different requirements that could be imposed by different scenarios, we designed a customisable measurement collection process. It contains a measurement profile enabling the definition of three main parameters to meet the requirements of a particular scenario. The first is the network path over which the measurements are collected. The second is the measurement metrics that need to be collected in order to support the Gateway Selection process. The third enables selecting the measurement mode, *Active* or *Passive*, applicable to a deployment scenario. The next stage after creating a measurement profile is to process it to configure the applicable measurement tools. There are a number of both passive and active tools that can be utilised and integrated into the process (Netperf and UDPMon for example). At the final stage, the measurement collection is carried out and repeated according to a configured time interval suited to a given deployment.

Each gateway advertises its capabilities and measurements to other MRs within the M-MANEMO tree. The TD protocol advertisements are extended to carry gateway information over the tree. The base TIO option is amended with a new sub-option, called the Gateway Information Sub-Option (GISO). Gateway attributes such as the Home-of-Address, HA address, and the current depth within the tree are included into the gateway advertisement in addition to the ID of the tunnel being advertised. The advertisement also contains network measurements collected during the network measurement process. The gateway advertisement is then propagated down the tree enabling each gateway to disseminate its information to the sub-tree of MRs branching off its ingress interface. Each MR receiving gateway advertisements collects the disseminated gateway information into a list, called the Gateway Discovery List. Each entry in the list corresponds to an available gateway and is frequently updated with the most up to date information.

3.3 Gateway Selection

The process of gateway selection enables a selecting node to make the selection decision according to the selection policies defined for a given MMNA deployment, as well as real-time information being disseminated by the candidate gateways. In order to insure that outbound and inbound traffic of a Mobile Network within a M-MANEMO tree is always tunnelled via the currently selected gateway instead of traversing the default path, this process also contains a route enforcement mechanism.

The selection process is run by all the gateways of a tree and each MR having more than one gateway available. The design of this process is based on the gateway selection decision-making module which takes two inputs. The first are the weights calculated for the selection criteria. The calculation is based on the importance rate given to each of the criteria according to the applied policy. The second input is the gateway information after being collected from the Gateway Discovery List and then normalised. Once the decision has been made, the selected gateway is provided as an input to the route enforcement process. The gateway selection process could be implemented either as a flow- or network-based selection. Flow-based selection allows finer granular selection where every flow type or set of flows is mapped to a selected gateway. In the case of network-based selection, the granularity level is coarser in that the decision is made for the traffic of a given mobile network. The process can be configured to run at a given time interval or based on specific events (link failure at a gateway, for example).

Additionally, we developed a mechanism whereby route enforcement is realised upon collaborative operations performed at the different M-MANEMO entities. Each gateway installs the necessary filtering and tunnelling rules to intercept and tunnel relevant outbound traffic via its tunnel. The respective HA also installs a tunnelling rule enabling inbound traffic to be tunnelled via the relevant tunnel. Once a MR has made a selection, it notifies its HA of the newly selected gateway. It sends an immediate BU message to its HA after attaching a new mobility option, called the Selected Gateway Information Option (SGIO), to contain the information of the newly selected gateway. Once received, the HA collects the information into its Traffic Forwarding List and installs the necessary filtering rules to intercept and route corresponding inbound traffic via the selected gateway. Upon receiving a successful acknowledgement from the HA, the MR continues the process of enforcing the selection, and applies a marking to the outgoing traffic of its Mobile Network. Given that the gateways and HA associate the tunnelling and filtering rules for each tunnel with its ID, the tunnel ID of the selected gateway is also utilised for packet marking. Each packet generated by the MNNs connecting to the MR is marked with the relevant tunnel ID. In a scenario where a MR selects a gateway belonging to a different HA, the Proxy-HA in this case also performs the role of Proxy-Gateway. Once it receives a BU+SIGO message, the Proxy-HA sends a Proxy BU+SIGO message to the Target-HA containing the ID of their HA-HA tunnel. This indicates to the Target-HA to route the corresponding inbound traffic via the HA-HA tunnel.

4. MMNA IMPLEMENTATION

We experimentally implemented the MMNA solution (using Linux kernel version 3.8.2) and this section provides an overview of the proof-of-concept implementation.

M-MANEMO was developed based on merging two main protocols, MANEMO and MCoA. These protocols have openly available Linux-based implementations on top of the original NEMO implementation. The MANEMO implementation (known as UMA+) was developed at Lancaster University whereas the MCoA implementation is available as Linux kernel and userland patches. We extended UMA+ to integrate the different MCoA functionality including BID mobility option processing. It also incorporates additional functionality such as TID processing.

In the current implementation, we defined two measurement profiles. The first is set as the default profile indicating the collection of measurements for metrics including bandwidth, delay, and packet loss, using an active mode. Accordingly, the Iperf and UDPMon active tools were incorporated to collect such measurements at a given interval. The profile also includes the collection of gateway uptime that is computed periodically, and access cost that is supplied manually. The second profile enables lightweight measurement to collect network bandwidth and load metrics using a simple passive monitoring tool that was developed based on tcpdump. For both profiles, the measurement is performed at the gateway entity for the tunnelling path between the gateway and HA entities.

For information dissemination, the Gateway Information Sub-Option (GISO) message was implemented containing attributes such as the HoA of the gateway and the its HA's IPv6 address. It also includes the TID of the advertised tunnel. The GISO also indicates the number of hops the recipient is from the advertising gateway (Depth), and hops number to which the advertisement is limited during propagation (Time-To-Live). It also contains metrics such as the Uptime to provide the elapsed time since establishing the advertised tunnel. The message also provides measurement information such as throughput, delay, and packet loss over the advertised tunnel. It also indicates the type of access link over which the tunnel is established.

To experimentally enable gateway selection functionality, we developed a simple selection algorithm. For criteria rating, a numerical scale of [1-5] is adapted to apply the relevant importance to each of the criteria of interest according to a static policy. Calculating the criteria weights based on the rating data was implemented using the Pairwise Comparison method. The criteria are compared against each other to build a comparison matrix in order to calculate the criteria weights based on the calculation of the geometric mean for each one. The normalization of the collected gateway data was implemented based on the min-max normalization method to map the data to values ranging from 0 to 1. Once the gateway data has been normalized and the criteria weights are in place, the decision is made using the Simple Additive Weighting method to select the gateway with the maximum sum.

The implementation of the route enforcement process was based on a number of functional components including packet marking, HA signaling, and tunneling and filtering rules installation. To enable in-line route enforcement for outbound traffic, the main IPv6 header is utilised to mark outgoing packets at the MR entity with the ID of the preferred tunnel. The Traffic Class (TC) field is set for packet marking. Since TC was mainly developed for QoS support, it has only local effect across the MMNA tree in this implementation and is reset for each packet leaving the tree. For HA signaling, a selecting MR communicates selection information, such as the preferred tunnel ID and the IPv6 address of the selected gateway, into a SGIO that is attached to an immediate BU message. Furthermore, the Linux XFRM framework in conjunction with the Linux IP filtering framework "Netfilter" were adopted for enabling each gateway and HA to install traffic tunneling and filtering rules allowing the new selection to be enforced. A Netfilter rule enables a gateway to intercept IPv6 packets with the TC set to the ID of its tunnel and mark them locally within the kernel to be matched and tunneled by the XFRM framework according to installed XFRM policies.

5. EVALUATION

To evaluate MMNA performance and capabilities, we built an experimental testbed, which is described in section 5.1. We then examine different multihoming properties in section 5.2. We highlight and focus on the capability of the approach to support preference setting. In addition, we briefly consider load sharing and failure recovery (for more details see [6]). Finally, we validate the MMNA in a real-world scenario by developing a Mountain Rescue use case. This is described in section 5.3 where we explain our assumptions and methodology and discuss the results.

5.1 Testbed Description

Different testbed setups (similar to Figure 2) were configured to conduct each of the experiments. All the testbed setups were built using a collection of Linux desktop PCs (2.9GHz CPU and 6GB RAM), fitted with Atheros Chipset 802.11a/b/g wireless interfaces in addition to two Ethernet interfaces. Linux kernel version 3.8.2 was installed on these machines. Three of them were configured to run as Access Routers (AR1, AR2, and AR3) while two machines were configured to operate as a Correspondent Node and a MMNA-enabled HA. These entities were interconnected via an Ethernet backbone network using a Netgear switch. The other PCs were configured to run the MMNA implementation as Gateways (GW1, GW2, and GW3) and Mobile Routers (MR1, MR2, and MR3). They were configured with Software-based Access Points (RADVD) to provide Mobile Networks. The gateways were connected to the ARs over Ethernet interfaces configured to emulate different connectivity. This is required to evaluate the behaviour of our solution on a more controlled environment and eliminate as much as possible the side effects of wireless properties. The GW1-AR1 link was configured to emulate a WiFi link (at 4.5Mbps) whereas GW2-AR2 and GW3-AR3 links were configured to emulate HSPA connections (at 1.8Mbps). Additionally, the wired infrastructure was configured with a dynamically varying delay (≈ 80ms). Each experiment was run ten times and the average result taken for each experiment.

5.2 Multihoming Study

In the preference setting experiment, each of MR1, MR2, and MR3 was configured to download a different file. Additionally, MR1 was configured to run the video streaming application and make three simultaneous VoIP calls. MR2 was also configured to run the video streaming application while MR3 was configured to make two simultaneous VoIP calls. All these communications were carried out with the CN. The experiment was carried out for a duration of 180 seconds and all the applications were being running during that period. The test started with a non-multihomed setup with only GW1 being advertised and consequently no preference was considered for the different applications. After 60 seconds of the experiment time, AL2 was brought up at GW2 which then started advertising low delay and packet loss link. This resulted in VoIP communications were redirected to AL2. The decision was also made by MR1 to redirect the file download traffic to AL2 as it had been less loaded than AL1. At time 100 seconds, GW3 established a connection over AL3 and advertising it as a secure access link. After running the selection process, MR2 and MR3 then redirected their respective file download traffic via AL3.

The results of the preference setting experiment in Figures 3 and 4 show that the situation improved once the tree had become multihomed after 60 seconds. The redirected communications of MR1 file downloading achieved an average increase of about 80% in throughput, as shown in Figure 3. There was also an increase of about 15% in the throughput achieved by MR2 and MR3 file downloads. Figure 4 shows that better TCP throughputs were being achieved by the video streaming, with the videos being streamed at average rate of 900-1100 Kbps. When GW3 became available at 120 seconds, a noticeable increase of more than 85% was achieved on the TCP throughput of MR1 and MR2 video streaming. Additionally, we examined VoIP performance focusing on jitter and the results showed a decrease of about 18% in the measured jitter after having multihomed access.

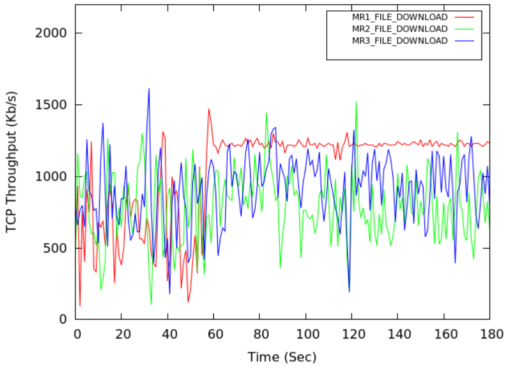

Figure 3. Preference Setting - File Download Results

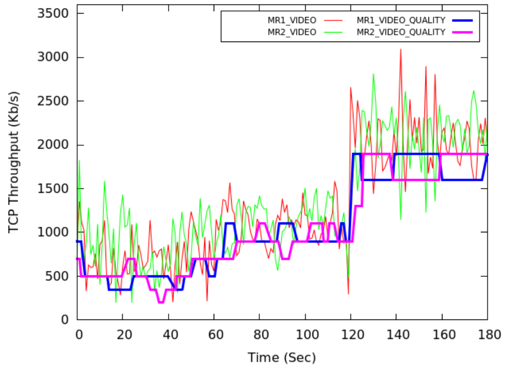

Figure 4. Preference Setting - Video Streaming Results

In the load sharing experiment, MR1, MR2, and MR3 were configured to receive 1.75, 3.75, and 1.5 Mbps UDP flows respectively. The test started with only the access of GW1 being available. Then, MR1 selected GW2 when it advertised its access option at 70 seconds. Figure 5 shows how the overall load was shared, at that point, among the available gateways (GW1 and GW2) allowing the stream over MR1 to reach its targeted throughput of 1.75Mbps. Meanwhile, the throughput at MR2 and MR3 also showed an increase of about 20-30%. Once GW3 disseminated the advertisement of its access link, it was selected by MR3 after 100 seconds and the overall load of the tree was shared among the three gateways. As a result, each of the MRs was able to receive the corresponding UDP flow at the targeted throughput. Additionally, we calculated the handoff delay when redirecting the traffic from one gateway to another. An average handoff delay of about 110ms was experienced by MR1 traffic and a shorter delay in the case of MR3 traffic since it traversed less hops via GW3 within the tree. No additional delay was experienced by MR2 traffic, which remained tunnelled via GW1.

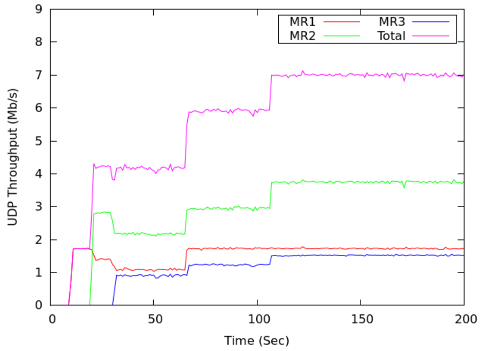

Figure 5. Load Sharing

The link failure recovery experiment was conducted through four scenarios. In all of the scenarios, a 1.5Mbps UDP flow was sent between one of the MRs and the CN over GW2. In scenario 1, 2, and 3 the stream was run by GW3 (has no access link), MR1, and MR3 (MR3 was connected behind MR1), respectively. The experiments ran for 200 seconds and after 110 seconds had elapsed the GW2-AR2 link went down and the traffic was then redirect to GW1. Table I shows that An average failure recovery delay of 3.3 seconds was required to redirect GW3 traffic to the default Gateway. The results show that the delay increases at about 1.1 seconds when the hops between the communicating MR and the failing Gateway increase in the other scenarios. In scenario 4, GW2 registered with HA2 and the stream was run between GW3 and the CN. The imposed HA-HA communication resulted in 10% additional delay on average to the experienced delay in scenario 1.

Table 1. Link Failure Recovery Delay (sec)

Scenario	Min	Max	Avg	Stdev
Scenario_1 (GW3)	2.989	4.579	**3.316**	0.493
Scenario_2 (MR1)	3.452	5.740	**4.581**	0.830
Scenario_3 (MR3)	4.483	7.196	**5.659**	0.974
Scenario_4 (GW3)	3.194	4.972	**3.611**	0.639

5.3 Mountain Rescue Use Case

In order to evaluate and examine the MMNA solution in a real-world scenario, we developed a use case example around mountain rescue. The focus was on a multi-team mountain rescue operation involving three search groups - SG1, SG2, and SG3, with SG3 belonging to a different team. The operation controller is based in the Headquarters and maintains permanent communication with the in-field members of the different search groups, over the Internet. Each of the rescue vehicles and group members is equipped with a MMNA-enabled multi-interfaced Mobile Router. The main focus was on the activities of SG1 and their interaction with other search groups when they converged on one search location at some points during the rescue mission. Management and control of the mountain rescue operation is based on different services (location tracking, telemedicine, voice communication, video steaming, and image sharing) integrated into a command and control system.

The evaluation of the MMNA in the mountain rescue scenario was carried out on an experimental testbed. The setup is similar to Figure 2 and as described earlier in section 5.1, with an additional HA and a number of additional access link. GW1-AR1 were configured to emulate a Wi-Fi access link (at 1800Kbps). Another link were set up for GW1 to emulate a satellite connection (at 670Kbps). The HSPA cellular connections were emulated over the GW2-AR3 and GW3-AR4 links (at 970Kbps). The communication among the different search parties in our use case was represented using a suite of different applications over the testbed. Location and biomedical data, and Image sharing were implemented as text and image files transfer over client-server TCP-sockets. To make G.711 VoIP calls, we used Linphone. A basic implementation of the Dynamic Adaptive Streaming over HTTP (DASH) was developed to run adaptive video streaming (according to measured throughput). Each of the testbed entities in Figure 2 corresponds to an entity of the mountain rescue use case example. The CN represents the management and control server at the Headquarters. HA1 and HA2 were the Home Agents of the teams. The evaluation was accomplished in nine stages carried out with different configurations and considerations. Here we present three sets of the more interesting results that focused on the interaction between the different search groups.

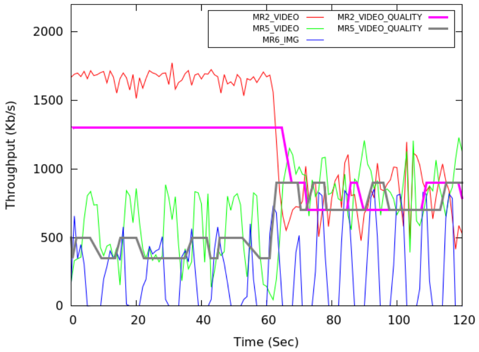

Figure 6. Tree Convergence (Aggregated) - Video Streaming and Image Transfer

In set 1, the test started with two distinct MMNA trees during the initial 60 seconds. The first tree (GW1, GW2, and MR3) had multiple access options whereas the second tree (GW3, MR5, and MR6) was single-homed. As shown in Figure 6, MR5's video stream achieved low TCP throughput with frequent reductions resulting in a fluctuating streaming rate (350-500 kbps). After 60 seconds, the two trees merged and GW3 operated as a MMNA Alternative Gateway for the combined tree. Consequently, MR5's video stream achieved a better TCP throughput resulting in an increase of about 28% on its streaming rate as shown in Figure 6. The Image-Transfer at MR6 was able to transmit 8 images compared to only 6 images before that. However, the TCP throughput and streaming rate of MR2 video streaming dropped via GW1, which can be adhered to the applied selection policies. Additionally, we examined VoIP jitter and found that a decrease of about 13% was achieved by GW3 VoIP communication on the experienced jitter after the tree convergence.

In set 2, the two trees were still converged for the initial 60 seconds. When the two trees split at 60 seconds, MR6 continued receiving the video stream with a more stable TCP throughput and streaming rate of 500 kbps as shown in figure 7. The TCP throughput of MR2 image traffic also improved, enabling more images to be transferred. However, the jitter experienced by MR4 VoIP traffic showed a noticeable increase. MR3 and MR5 were internally communicating over the tree during the initial 60 seconds. This enabled high TCP throughput to be achieved as shown in Figure 8. After the two trees split, MR3-MR5 communication was maintained as inter-tree communication over GW1 and GW3 (via HA1), with the frequent reductions due to overlapping with MR2 image transfer.

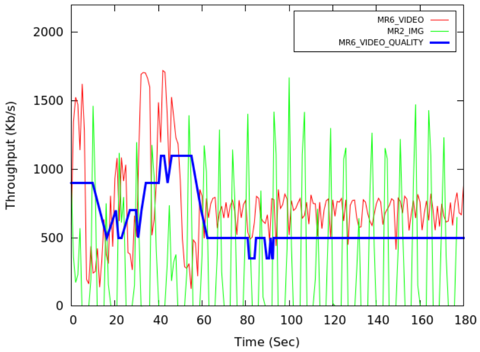

Figure 7. Tree Split - Video Streaming and Image Transfer

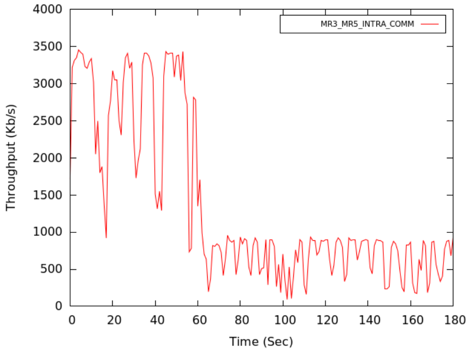

Figure 8. Tree Split - MR3-MR5 Intra-communication

In set 3, the test started with two distinct MMNA trees during the initial 60 seconds. The first tree (GW1, GW2, MR3) registered with HA1, whereas the other tree (GW3 and MR5) registered with HA2. Figure 9 shows that the throughputs achieved by MR5 video streams experienced frequent drops and achieved a streaming rate of 500 kbps. After 60 seconds, when the two trees converged, MR5's video stream achieved a better throughput resulting in an increase of about 20% on its streaming rate. Figure 10 also shows a reduction of about 39% and 17% on the jitter experienced by the VoIP traffic of MR4 and MR5, respectively. However, the throughput of MR2 video streaming also declined after 60 seconds. This can be adhered to the applied selection policies which caused an increase in traffic contention at GW1.

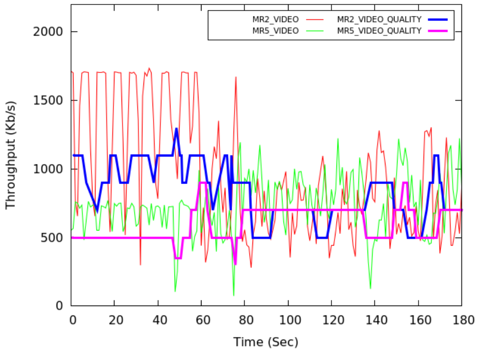

Figure 9. Tree Convergence (Non-Aggregated) - Video Traffic

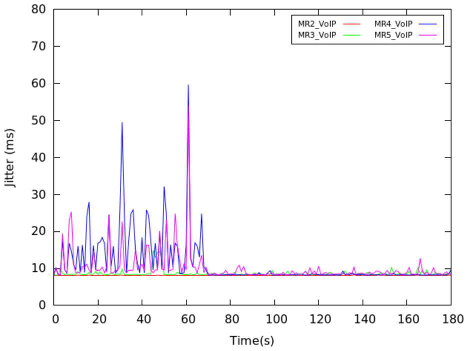

Figure 10. Tree Convergence (Non-Aggregated) - VoIP Jitter

6. CONCLUSION

In this paper, we have presented an overview of the Multihomed Mobile Network Architecture (MMNA) providing comprehensive multihoming support for complex Nested mobility scenarios. The MMNA solution was experimentally implemented and evaluated over different testbed setups. As the results explained, MMNA provided adequate support for the capabilities of load sharing, link failure recovery, and preference setting, allowing improved overall performance of ongoing communications. We believe that that the MMNA has the potential for advanced support in this context. The results also highlighted the effectiveness and feasibility of the MMNA approach considering a Mountain Rescue use case example. The support of the convergence of multiple trees and sharing Internet connectivity even between those originating from different Home Networks demonstrated a practical dimension to the approach. While the focus has been mainly on Internet accessibility across an MMNA topology, local communication is of great importance in such scenarios. The ability to sustain communication between different nodes within a tree, even after a tree split, demonstrates the applicability of MMNA to such failure-prone environments.

7. REFERENCES

[1] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, *Network mobility basic support protocol*, IETF RFC 3963, January 2005.

[2] C. Ng, P. Thubert, M. Watari, and F. Zhao, *Network mobility route optimization problem statement*, IETF RFC 4888, July 2007.

[3] B. McCarthy, C. Edwards, and M. Dunmore. "Advances in MANEMO: Denition of the Problem Domain and the Design of a NEMO-Centric Approach". *2007 International Symposium on Applications and the Internet Workshops (SAINTW'07)*, September 2007.

[4] B. McCarthy, M. Jakeman, and C. Edwards, "Supporting Nested NEMO networks with the Unified MANEMO Architecture", In *Proc. Of the 34th IEEE Conference on Local Computer Networks (LCN 2009)*, pp. 609-616, 2009.

[5] R. Wakikawa, V. Devarapalli, G. Tsirtsis, T. Ernst, and K. Nagami, *Multiple Care-of Addresses Registration*, IETF RFC 5648, 2009.

[6] I. Alsukayti and C. Edwards, "Multihomed Mobile Network Architecture," *Networking Conference, 2015 IFIP* , pp.195-203, 2015.