Institut Eurécom [1]
Department of Mobile Communications
2229, route des Crêtes
B.P. 193
06904 Sophia-Antipolis
FRANCE

Research Report RR-06-175

An early application of SCTP/mSCTP for multi-homing in netLMM

November 23, 2006

Thesis Advisor: Prof. Christian BONNET
PhD Student: Huu Nghia NGUYEN

Tel: (+33) 04.93.00.82.38
Fax: (+33) 04.93.00.26.27
Email : {Huu-Nghia.Nguyen,Christian.Bonnet}@eurecom.fr

[1] Institut Eurécom's research is partially supported by its industrial members: BMW, Bouygues Télécom, Cisco Systems, France Télécom , Hitachi Europe, SFR, Sharp, STMicroelectronics, Swisscom, Thales

Abstract

This document briefly describes the localized mobility problem and the current edge-based solution of IETF, the NETLMM architecture. We implemented a simplified version of NETLMM under ns2, in which, we tried to integrate the multi-homing feature with the use of SCTP/mSCTP in both data plane and control plane of NETLMM. In the control plane, we try to use mSCTP signaling between the AR and the LMA instead of EMP or other standardizing signaling protocol as a first step to introduce multi-homing feature and to support both IPv4 and IPv6. In the data plane, we proposed a new SCTP encapsulation schema for NETLMM. This SCTP encapsulation mechanism provides a way to reduce the header overhead for small size packet networks by allowing many small packets to share the same header. By analysing the simulation results, we found the trade-off relation between the header overhead and the tunneling delay. This relationship is a kind of conservation that we can dynamically control the trade-off with regards to the network status. Besides, this new SCTP encapsulation schema is very promising for extending the capacity of the backhaul of NETLMM domain thanks to the bandwidth aggregation feature.

Keywords: *B3G, EMP, Encapsulation, Localized Mobility Management, Multi-homing, Mobility, NETLMM, mSCTP*

Table of contents

Table of Figures

Introduction

Localized mobility management (LMM) has been the topic of much work in the IETF for some time, and it may seem as if little remains to be said on the topic. The edge-based LMM (NetLMM) [1][2][3][4] is currently standardized by l'IETF and is principally based on an assumption of unmodified mobile nodes. It comprises of two parties. The first part defines the interface between the Mobile Node (MN) and the Access Router (AR) and the second part defines the interface between the access router (AR) and the localized mobility agent (LMA).

The NetLMM protocol only concentrates on the control plane between entities in the NetLMM domain and not the data plane. The data plane is supposed to use a tunneling mechanism (IP in IP, GRE, MPLS). There is something missing: The interaction between the Localized Mobility Management and Global Mobility Management in the control plane is still not defined. It supports a one-to-many relation between the mobile node identifier (MNID) and the locators. However, it doesn't mention about the use of simultaneous locators.

This document briefly describes the localized mobility problem and the current edge-based solution of IETF, the NETLMM architecture. Then, it proposes and analyses the application of SCTP/mSCTP [5] [6] [7] [8] for NETLMM. In the control plane, we try to use mSCTP signaling between the AR and the LMA instead of EMP or other standardizing signaling protocol as a first step to introduce multi-homing feature in NETLMM. In the data plane, we can benefit a message bundling mechanism with the use of SCTP encapsulation. This SCTP encapsulation mechanism provides a way to reduce the header overhead for small size packet networks by allowing many small packets to share the same header. We then show some simulation results and analysis to validate the idea, and to find the trade-off that we have to pay for the promising bandwidth aggregation feature.

1. Abbreviations

AR	Access Router
CGA	Cryptographically Generated Address.
CN	Correspondent Node
CoA	Care of Address

DNA	Detecting Network Attachment
EMP	Edge Mobility Protocol
GMM	Global Mobility Management
HoA	Home Address
LMA	Localized Mobility Agent (The old name is MAP)
LMM	Localized Mobility Management
LMMD	Localized Mobility Management Domain
LNMP	NetLMM Protocol used in the backhaul of the NetLMM domain (between ARs and LMA.).
MN	Mobile Node
MNID	Mobile node identifier
NA	Neighbor Advertisement
ND	Neighbor Discovery
NDP	Neighbor Discovery Protocol
NetLMM	Network-based LMM
NS	Neighbor Solicitation
RA	Router Advertisement
RHoA	Regional HoA
RS	Router Solicitation
SEND	SEcure Neighbor Discovery

2. Terminologies

Cryptographically Generated Addresses (CGA): are IPv6 addresses for which the interface identifier is generated by computing a cryptographic one-way hash function from a public key and auxiliary parameters. The binding between the public key and the address can be verified by re-computing the hash value and by comparing the hash with the interface identifier. Messages sent from an IPv6 address can be protected by attaching the public key and auxiliary parameters and by signing the message with the corresponding private key. The protection works *without a certification authority or* *any security infrastructure.*

CGA_LL: The link-local unicast CGA generated by the MN with its public key (It is assumed that the MN is using a single public key to configure all of its link-local unicast and global unicast CGAs.)

CGA_1: One of the Global Unicast CGA generated by the MN with its public key.

CGA_2: Another one of the Global Unicast CGA generated by the MN with its public key (e.g. with a different subnet prefix.)

CGA_*: Any Unicast CGA generated by the MN with its public key (i.e. link-local or global.)

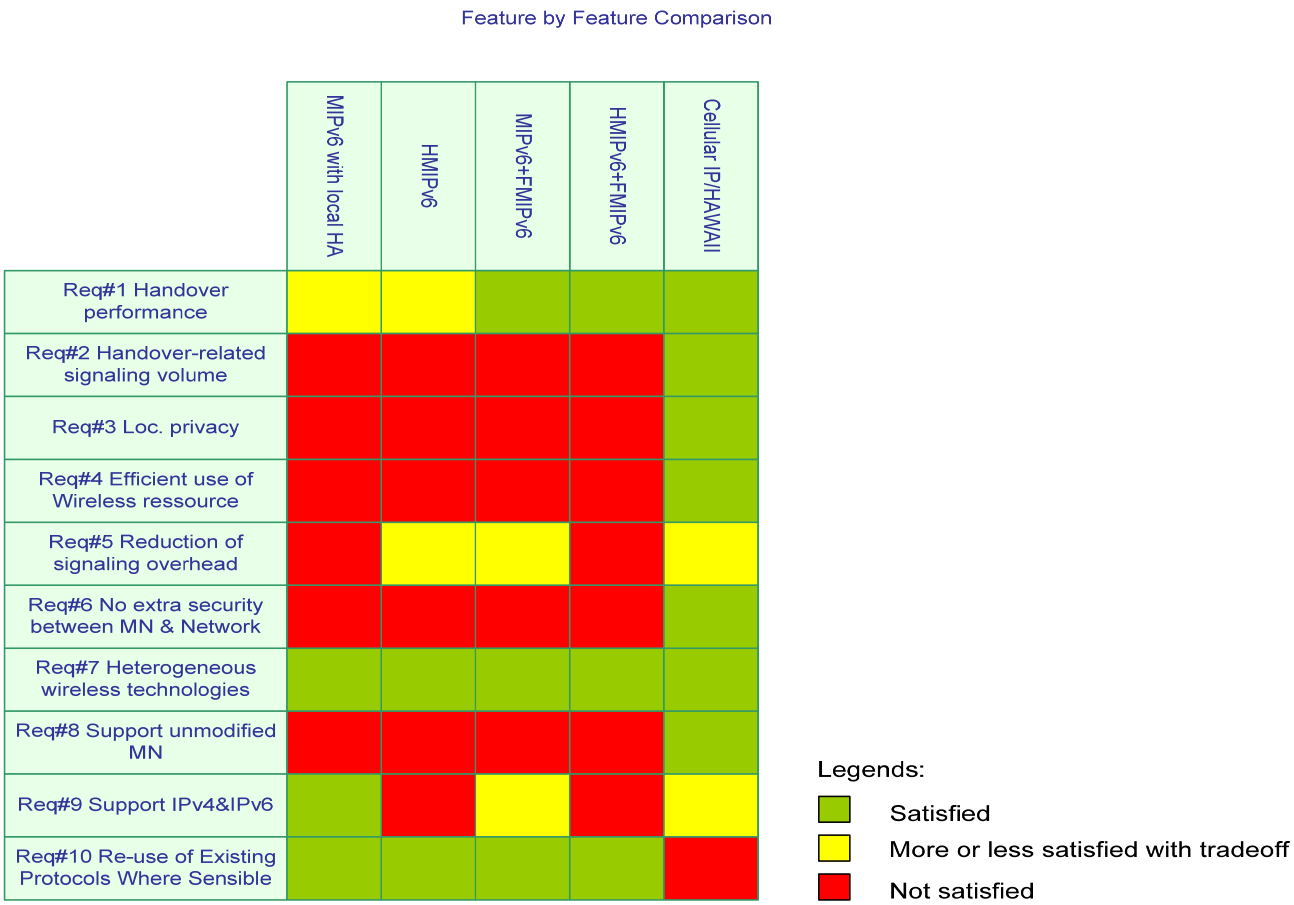
MNID: Mobile node identifier set to the public key used by the MN for generating its CGAs.

Global Mobility Anchor Point: A node in the network where the mobile node maintains a permanent address and a mapping between the permanent address and the local temporary address where the mobile node happens to be currently located. The Global Mobility Anchor Point may be used for purposes of rendezvous and possibly traffic forwarding.

3. NetLMM protocol

Localized Mobility Management is a generic term for protocols dealing with IP mobility management confined within the access network. The Localized mobility management signaling is not routed outside the access network, although a handover may trigger Global Mobility Management signaling. Localized mobility management protocols exploit the locality of movement by confining movement related changes to the access network. The LMM addresses mainly at the 3 following problems: Update latency, Signaling overhead and Location privacy.

The document [3] develops more detailed requirements for a localized mobility management protocol (There are 10 requirements at the moment of writing this report). The analysis reveals that none of the existing protocol can satisfy all the requirement of Localized Mobility Management. IETF therefore recommended a network-based approach to localized mobility management called NetLMM.



Feature by Feature Comparison

	MIPv6 with local HA	HMIPv6	MIPv6+FMIPv6	HMIPv6+FMIPv6	Cellular IP/HAWAII
Req#1 Handover performance	🟨	🟨	🟩	🟩	🟩
Req#2 Handover-related signaling volume	🟥	🟥	🟥	🟥	🟩
Req#3 Loc. privacy	🟥	🟥	🟥	🟥	🟩
Req#4 Efficient use of Wireless ressource	🟥	🟥	🟥	🟥	🟩
Req#5 Reduction of signaling overhead	🟥	🟨	🟨	🟥	🟨
Req#6 No extra security between MN & Network	🟥	🟥	🟥	🟥	🟩
Req#7 Heterogeneous wireless technologies	🟩	🟩	🟩	🟩	🟩
Req#8 Support unmodified MN	🟥	🟥	🟥	🟥	🟩
Req#9 Support IPv4&IPv6	🟩	🟥	🟨	🟥	🟨
Req#10 Re-use of Existing Protocols Where Sensible	🟩	🟩	🟩	🟩	🟥

Legends:

🟩 Satisfied

🟨 More or less satisfied with tradeoff

🟥 Not satisfied

Figure 1. Feature by Feature comparison of different LMM solutions

The analysis is based on some personal estimation and the document [3]. Use it with your own risk! There are only 3 possible value for each feature {Not satisfied, More or less = Partial, Satisfied}.

3.1. Architecture

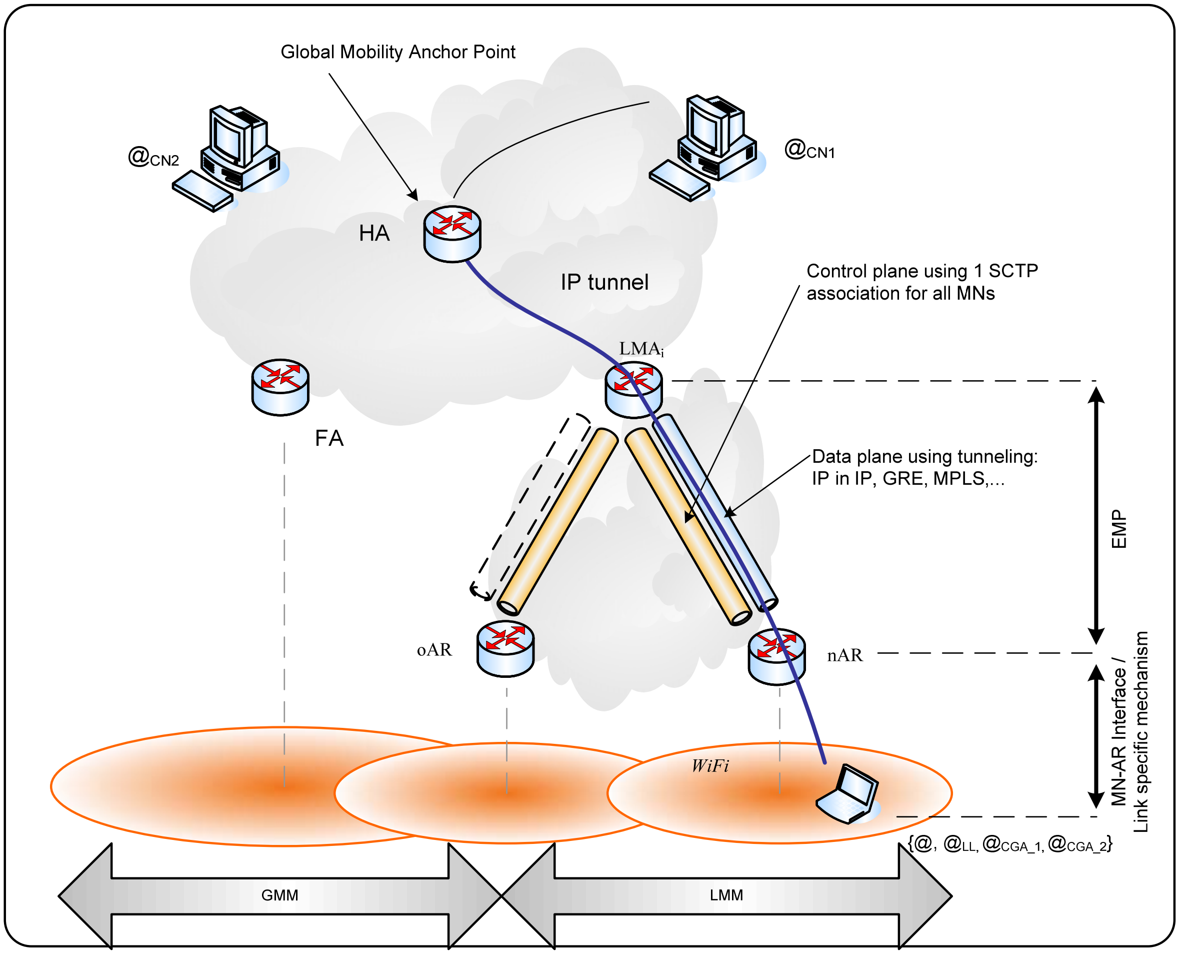

Global Mobility Anchor Point
@CN2
@CN1
HA
IP tunnel
Control plane using 1 SCTP association for all MNs
LMAi
FA
Data plane using tunneling: IP in IP, GRE, MPLS,...
EMP
oAR
nAR
MN-AR Interface / Link specific mechanism
WiFi
{@, @LL, @CGA_1, @CGA_2}
GMM
LMM

Figure 2. Architecture for netLMM solution using EMP

3.2. MN-AR Interface

The MN-AR NetLMM interface is used between a MN node and an AR of a NetLMM domain. In the absence of link-layer specific mechanism, it allows the AR to detect the network attachment of a MN and update routing at the LMA so that the MN stays reachable when it roams across the NetLMM domain. The draft draft-ietf-netlmm-mn-ar-if [4] specifies such an IP layer interface between mobile nodes (MN) and access routers (AR) of a network-based localized mobility. It is required that no NetLMM specific software support is present on MNs. The IP layer MN-AR interface described in this document fulfills these requirements by using the SEND public key as the MN identifier, while being solely based on standard track IPv6 protocols (DNA and SEND) implemented by non-NetLMM MNs.

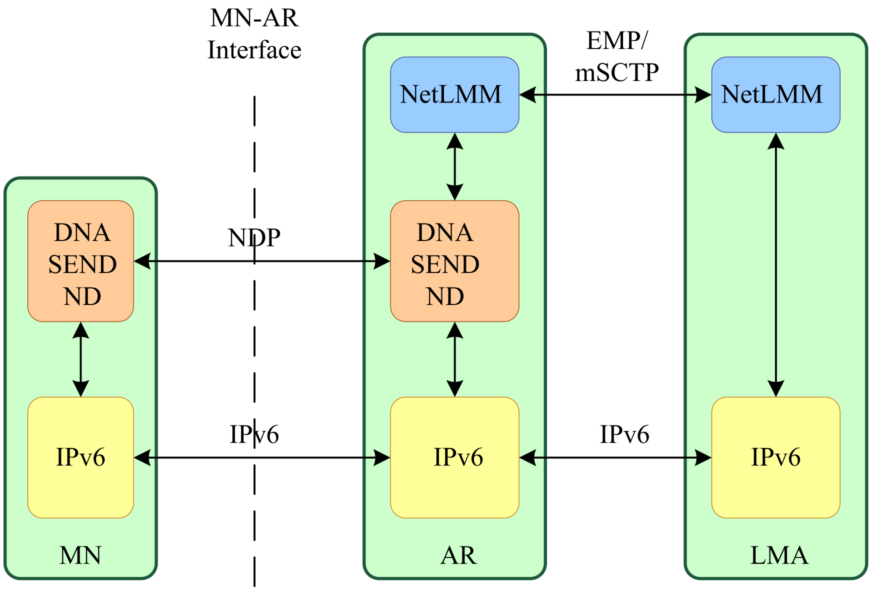


Figure 3. Protocol stack for netLMM solution

The interface MN-AR supports the following scenarios

- MN powers on in a NetLMM domain
- First attachment of MN moving into a NetLMM domain
- MN handovers in a NetLMM-domain
- MN configuring additional CGAs
- MN configuring CGA that is in use by another MN in the NETLMM domain
- MN un-configures CGAs, powers off, crash or leave the domain

3.3. AR-LMA Interface (EMP)

The interface between LMA and the AR can be EMP. EMP only defines the control plane. The data plane is supposed to use any available tunneling method specified in the HELLO message.

EMP uses a MN identifier, referred to as a MNID in this document, to manage tunnel information or forwarding entries at the LMA or AR. The MNID must be unique and unchanging in the LMM domain, and is used to associate the MN with its related information. Some examples of MNIDs are a Network Access Identifier, a Mobile IP Home Address, and a link dependent identifier. In the case of the 802.11 binding, the ID will be simply the 802.11 MAC address. The AR must be able to set the MNID in all EMP messages it sends. If the link-layer technology is unable to provide such functionality, the

AR must keep some state on the MNID.

The EMP signaling is sent using SCTP association between the LMA and the AR. The association is established when the AR powers up and is used for all MNs. The message structure follows the TLV format like other SCTP messages. EMP defines 4 messages:

Name	**Meanings**
Hello	HELLO messages are exchanged between an AR and the LMA during AR startup.
Query	When an AR detects that a MN has joined its link, it sends a QUERY containing the MNs ID to the LMA. The LMA responds with an UPDATE REPLY containing the MN's ID and all global addresses belonging to the MN, if any are known.
Update	Either an AR or the LMA can send an UPDATE. When sent from an AR to the LMA with the code set to 0, the message contains the MN ID and a new IP Address for verification, and the AR expects a reply.
Reply	REPLY messages are sent from the LMA to the AR in response to an UPDATE or a QUERY. Each REPLY message always contains a MNID. If the REPLY is sent in response to an UPDATE, the address is the same address that was in the UPDATE, and conveys status information to the AR. If the REPLY is sent in response to a QUERY, the reply contains all known IP addresses belonging to the MN.

EMP must handle three basic scenarios:

1. A MN powers-on in the LMMD.
2. A MN moves to a new AR in the same LMMD
3. A MN crashes, powers-off, leaves the coverage area, or moves to a different LMMD

3.4. *Control plane scenarios*

3.4.1. First attachment of MN to the LMM domain (Control plane)

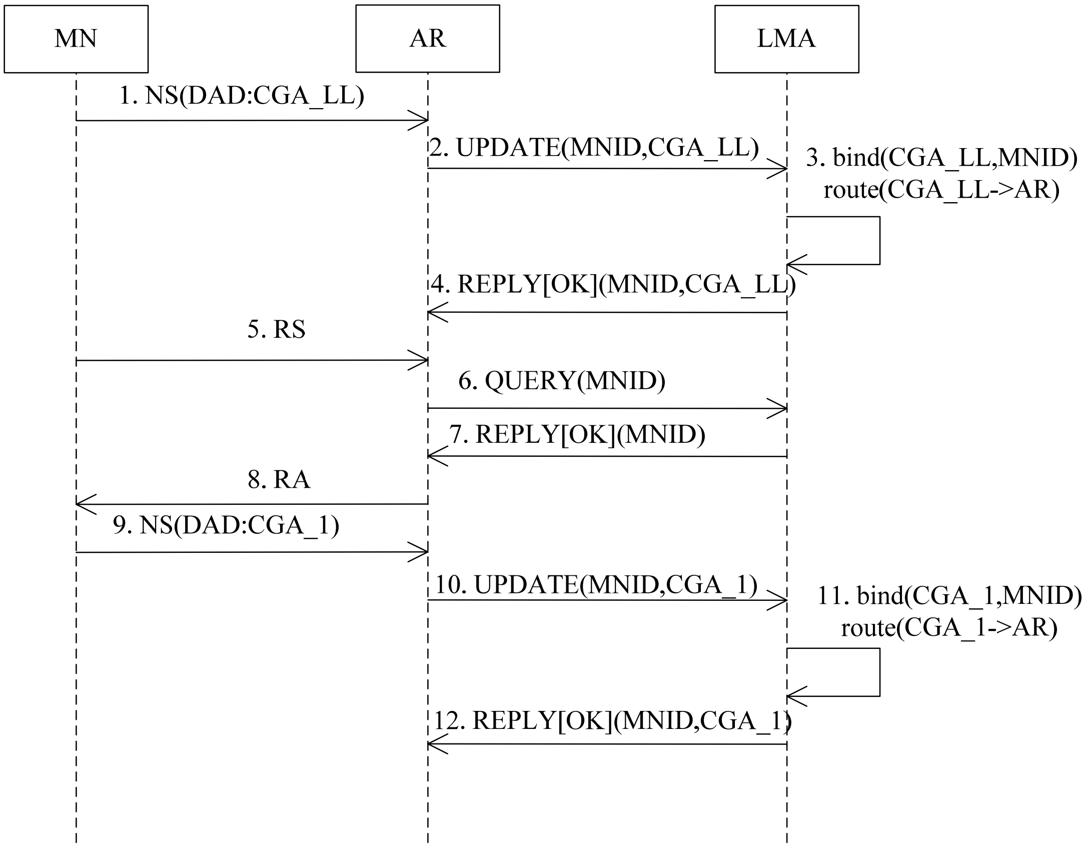


Figure 4: MN powers on and configures a Link-Local and 1 Global Unicast CGAs

The Idea is similar to the idea of HIP in the sense of using the SEND public key as an identifier MNID. When a MN powers on for the first time, it will generate a link local address based on its public key (CGA_LL) as per RFC3972 [9]:

1. The MN performs DAD on the address as per RFC2462 [10]. The DAD-NS message generated will contain the public key in the CGA option as defined by SEND [11].
2. Upon reception of this NS message, the access router AR SHOULD generate a UPDATE to the LMA with the public key as the MNID along with CGA_LL.
3. The LMA SHOULD bind the CGA_LL to the MNID and establish a route binding for the CGA_LL to the access router AR1.

4. The LMA acknowledges the receipt of the UPDATE message.
5. While waiting for the completion of DAD, the MN may generate RS message as per RFC2461 [12] with the unspecified address as the source address. Such an RS message will not contain a CGA option.
6. When the AR detects that a MN has connected to its link (i.e. by receipt of a RS), in order to recognize if the MN is powered or is moving, the AR queries the LMA for information about the MN.
7. Because this is the first attachment, the LMA has no information for the MN, so it replies with a message empty except for the MNID.
8. The access router will respond with a **multicast RA** as per RFC2461 [12]. With the prefix information received in the RA message,
9. The MN will cryptographically generate one or more global addresses (CGA_*). For each of these addresses, the MN will perform DAD as the IID (???) is likely to be different for each of these cryptographically generated addresses. In this example, we assume that there is a global address CGA_1
10. For every DAD-NS received from the MN, the access router AR1 will generate a UPDATE message to the LMA establishing binding in the LMA.
11. The LMA SHOULD bind the CGA_1 to the MNID and establish a route binding for the CGA_1 to the access router AR1.
12. The LMA acknowledges the receipt of the UPDATE message.

3.4.2. Moving to a new link in the LMM domain (Control plane)

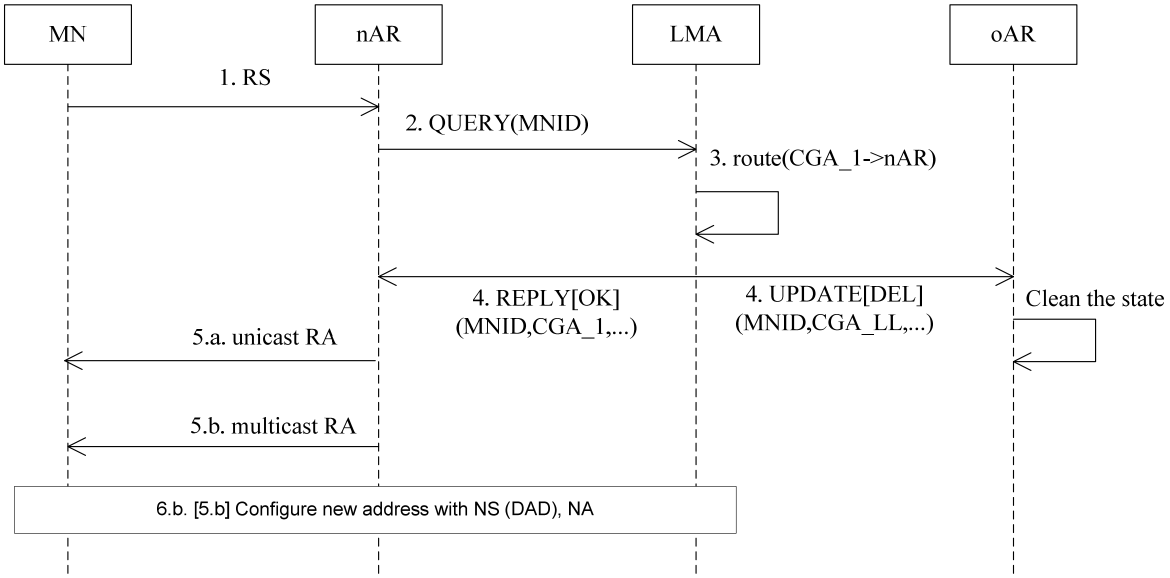


Figure 5: MN getting handover hint

A MN can configure a new address at any time; however it is most likely to do so when it enters a new LMMD. When the MN moves within the NETLMM domain:

1. It will send a RS message with the source address as its link-local address as specified by [13].
2. The new Access Router again can use the public key in CGA option to infer the MNID and sends a QUERY to the LMA. Because the MN has registered to the LMA before and is moving to a new AR, the LMA has an entry for the MN,
3. It also deduces that the MN has moved to a new AR in its LMMD, so it switches the MN's traffic to the tunnel to the new AR,
4. The LMA sends the new AR the MN's IP addresses so the new AR can update its forwarding state (Figure 2) and informs the old AR so that it can clean up state.
5. The new AR responds a message RA to the MN
    a. If the new access router chooses to respond with a unicast RA, all required steps are done.
    b. The new access router can choose to respond with a multicast RA

6. If 5.b happens, the MN will send a NS to learn about the new access router and confirm the reachability.

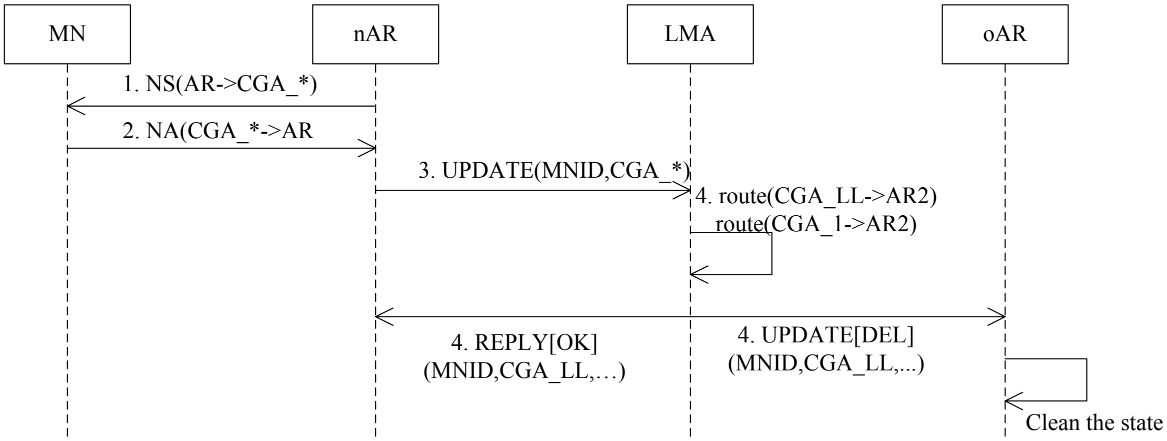


Figure 6: AR getting handover hint of MN whose IP address is known

Instead of the MN receiving the hint, in scenarios were the new access router receives the hint with the IP address of the handing over MN,

1. The AR can send a NS to that IP address.
2. The NA message received in response will contain the public key of the MN
3. With the received MNID, the AR can send update message to the LMA.
4. The LMA sends REPLY to nAR and send UPDATE[DEL] to oAR to clean up the state as in the previous scenario.

3.5. Data plane scenarios

Draft-wood-netlmm-emp-base [14] assumes to use SCTP transport layer only for signaling messages. For the data delivery, it assumes to allow the LMA and ARs to choose the right tunneling methods: IP in IP, GRE, PLMS, Null method.

3.5.1. IPv6 / IPv6

IPv6 packets destined to the MN are encapsulated at the LMA in an IPv6 tunnel terminating at the MN's current AR. This has the advantage of utilizing the IPv6 routing topology that is likely to be in place. However, due to the size of IPv6 headers, this method may impose a larger overhead, relative to other tunnel methods.

3.5.2. GRE

IPv6 packets destined to the MN are encapsulated at the LMA in a GRE tunnel.The GRE tunnel terminates at the MN's current AR.

3.5.3. MPLS

IPv6 packets destined to the MN are assigned to a forwarding equivalence class (FEC) by the LMA. The packets then traverse a label switched path (LSP) mapped to the MN's FEC. The LSP terminates at the AR (i.e. the AR is the LSP egress).

The path begins at a Label Edge Router (LER), which makes a decision on which label to prepend to a packet based on the appropriate FEC. It then forwards the packet along to the next router in the path, which swaps the packet's outer label for another label, and forwards it to the next router. The last router in the path removes the label from the packet. and forwards the packet based on the header of its next layer, for example IPv4. Due to the forwarding of packets through an LSP being opaque to higher network layers, an LSP is also sometimes referred to as an MPLS tunnel. The router which first prepends the MPLS header to a packet is called an ingress router. The last router in an LSP, which pops the label from the packet, is called an egress router. Routers in between, which need only swap labels, are called transit routers or Label Switching Routers.

For some networks, MPLS may have a number of benefits compared to other tunnel methods. Its forwarding overhead can be lower and it can utilize simpler routers, and the encapsulating header can be smaller than that required by other tunnel methods. It also lends itself to the application of traffic engineering within an LMMD, permitting traffic optimization techniques such as load balancing, routing around failures, and enhanced QoS. It may also be possible to enhance a LDP to perform route optimization for traffic between MNs in the same LMMD. However, MPLS tunnels may also entail more complexity than other tunnel methods, since it may require significantly more effort to set up and manage the protocols and infrastructure necessary.

3.5.4. Null Method

This is a pseudo tunnel method. When using it, the LMA and AR do not set up any sort of tunnel. It can be used when tunneling is not necessary (i.e. when the LMA is co-located with an AR) or some other mechanism is in place to deliver the packets to the AR.

3.5.5. GTP

To be analyzed by IETF.

4. Application of mSCTP in NetLMM control plane

The current EMP is just a straw man AR-LMA interface without any experimentation and is used as a base protocol for the AR-LMA interface design process. We can find many similarities between EMP and mSCTP. mSCTP is well defined in draft-ietf-tsvwg-addip-sctp [15] and has been implemented in Linux Kernel SCTP (lkSCTP). Some functionalities of EMP (e.g. Query) can be done with mSCTP by cleverly using mSCTP messages.

	EMP	**mSCTP**
Requirement	SCTP association	SCTP association
Message Format	TLV, Sent as a message in the DATA chunk	TLV, Sent as a ASCONF parameter in a ASCONF chunk
Message Names	Hello, Query, Update, Reply	Add IP Address, Set Primary IP Address, Delete IP Address in ASCONF and ASCONF-ACK chunk.
Functionalities	Choose the tunneling method Find a MN Update a MN route Delete a MN route	Add a route (IP Address) Choose a primary route Delete a route Use concurrent route
Experimented	No	Yes (lkSCTP)

Some issues may raise: How to recreate the mSCTP state at another AR while the MN is moving.

5. SCTP encapsulation for NetLMM data plane

This section mentions briefly the idea of using SCTP encapsulation in the Data plane. This kind of tunneling is useful for telecommunication/multimedia networks in certain conditions. If we can cleverly use the SCTP tunnel between ARs and LMAs, we can reduce the overhead per packet in the backhaul of the NetLMM domain (between ARs and LMAs). However, the real-time constrain and resource constrains require more consideration.

The SCTP encapsulation is proposed for the 3 following reasons:

- It allows load sharing (with LS SCTP extension): With the SCTP encapsulation, if LMA and MN support the LS SCTP version, we will be able to allow the LMA to distribute load to the MN by different paths (over different access technologies) therefore we have a larger aggregated bandwidth.
- It is suitable for small-size packet networks. It can bundle small packets (e.g. VoIP packet) in one SCTP packet/datagram, therefore reduces the overhead.
- It requires minimum implementation (we can reuse mSCTP code for both Control plane and Data plane therefore reduce the number of tasks)

Of course, the idea needs further consideration for the feasibility and optimal performance. For example, define a new chunk type for the encapsulation.

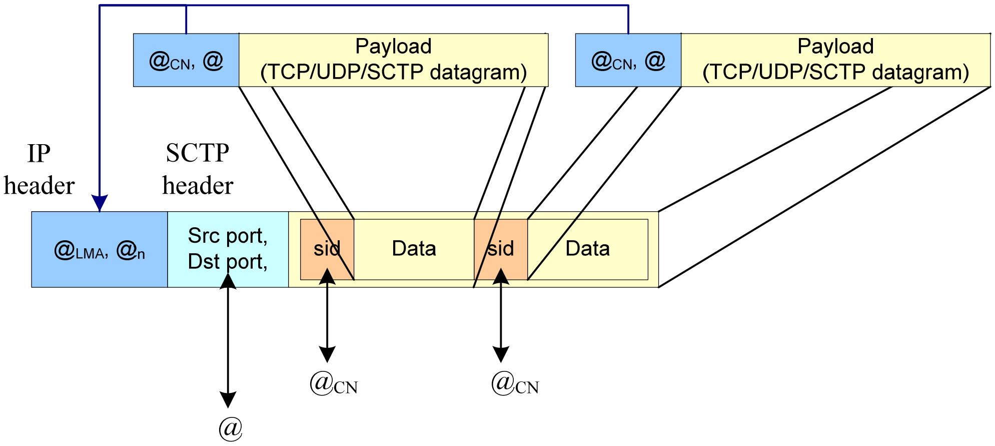


Figure 7. Bundling in case of no encryption

The above figure shows a schema in which many small TCP/UDP packets share the same IP routing information. Perhaps this is the best case of SCTP encapsulation. The idea is to map ($@_{CN}$, @) <--> (Stream identifier *sid*, (*Src port*, *Dest port*)) so that the LMA can deduce the original routing information without looking inside the encapsulated packet.

Issue: In this first schema, we have to maintain the sid <--> $@_{CN}$ mapping at both the LMA and the MN. It means we may need at least a message for the synchronization between the LMA and the MN.

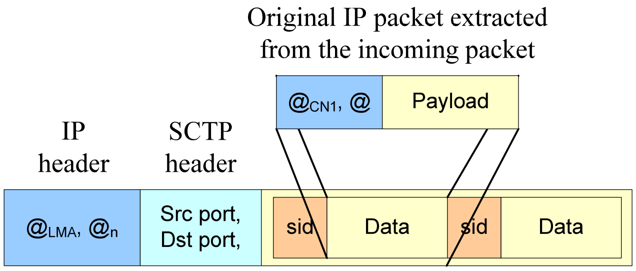


Figure 8. Bundling in case of encryption or different IP header information.

The figure shows a schema in which we are unable to reconstruct all original packets from the common IP header. In this case, we may have to put the whole original IP packet in a data chunk. This is the simplest case of SCTP encapsulation.

6. Results and Analysis

The use of mSCTP and SCTP source code for the AR-LMA interface requires to modify the SCTP finite states machine so that a ASCONF chunk can be sent without the acknowledgement for the previous outstanding ASCONF (In the original version, if there is an outstanding ASCONF chunk, the next ASCONF chunk will be dropped).

The transmission of traffic in the backhaul network between AR and LMA requires some extra header information which causes header overhead. The optimization of the data plane aims at increasing the effective bandwidth (the utilization) of the backhaul network resources. The idea is simplified and implemented under ns2 version 2.29. In this implementation, we use the above second schema of encapsulation (the simplest schema).

We constructed an LMM infrastructure as described in section 3 and section 5. While varying the number of mobile nodes and/or the incoming IP packet size, by measuring the total size of CBR packets going out from the AR to MNs ($B_{encapsulated}$) and the total size of encapsulating SCTP packets on the link between the AR and the LMA ($B_{encapsulating}$), we calculate the bandwidth utilization and the header overhead by the following formulas:

$$Overhead = \frac{B_{encapsulating} - B_{encapsulated}}{B_{encapsulating}}$$

$$Utilization = \frac{B_{encapsulated}}{B_{encapsulating}} = 1 - Overhead$$

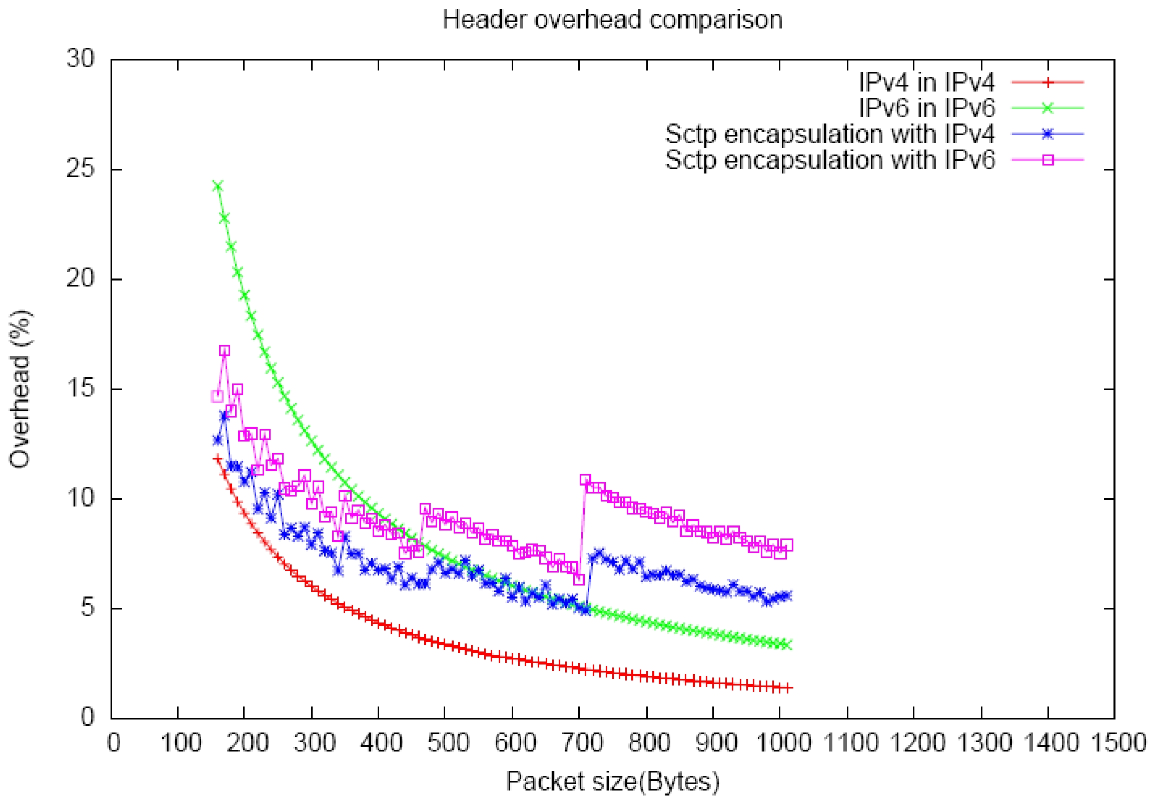


Figure 9. The header overhead comparison (in percent of total bandwidth)

The above figure shows the header overhead comparison between SCTP encapsulation with IP-in-IP encapsulation (under both IPv4 and IPv6). Under IPv4, the IPv4-in-IPv4 encapsulation is best in any case. However under IPv6, the overhead of SCTP encapsulation is smaller (therefore better) for packet size smaller than 450 Bytes. We can still optimize the SCTP encapsulation and archive better bandwidth utilization by defining a new chunk type with a minimal chunk header size in which we keep only the chunk type and chunk length fields.

Returning to the comparison between SCTP encapsulation under IPv6 and IPv6-in-IPv6, while the packet size increases, the number of encapsulating data chunk decreases because the packet size is limited by the frame size (For Etherenet, the frame size is 1500). There will be a peak whenever the number of encapsulating data chunk decreases. And step by step the header overhead of SCTP encapsulation with IPv6 will reach the value of IPv6-in-IPv6 encapsulation. Only after that, the SCTP encapsulation gives a bigger overhead.

We are going to illustrate here a scenario of good use of SCTP encapsulation. For example, if G.711 is used as Pulse Code Modulation (PCM) for VoIP, a VoIP packet is composed as follows:

UDP header = 8bytes
RTP header = 12bytes
Payload = 160bytes. per 20ms Because G.711 is sampling at 8KHz, then every sample can be expressed by 8bits: 20 ms * (8000 samples / 1000) * 8(bits) = 1280bits (= 160bytes).

The SCTP encapsulation with IPv6 give a smaller header overhead of about 10% of the total bandwidth in compare to IPv6-in-IPv6.

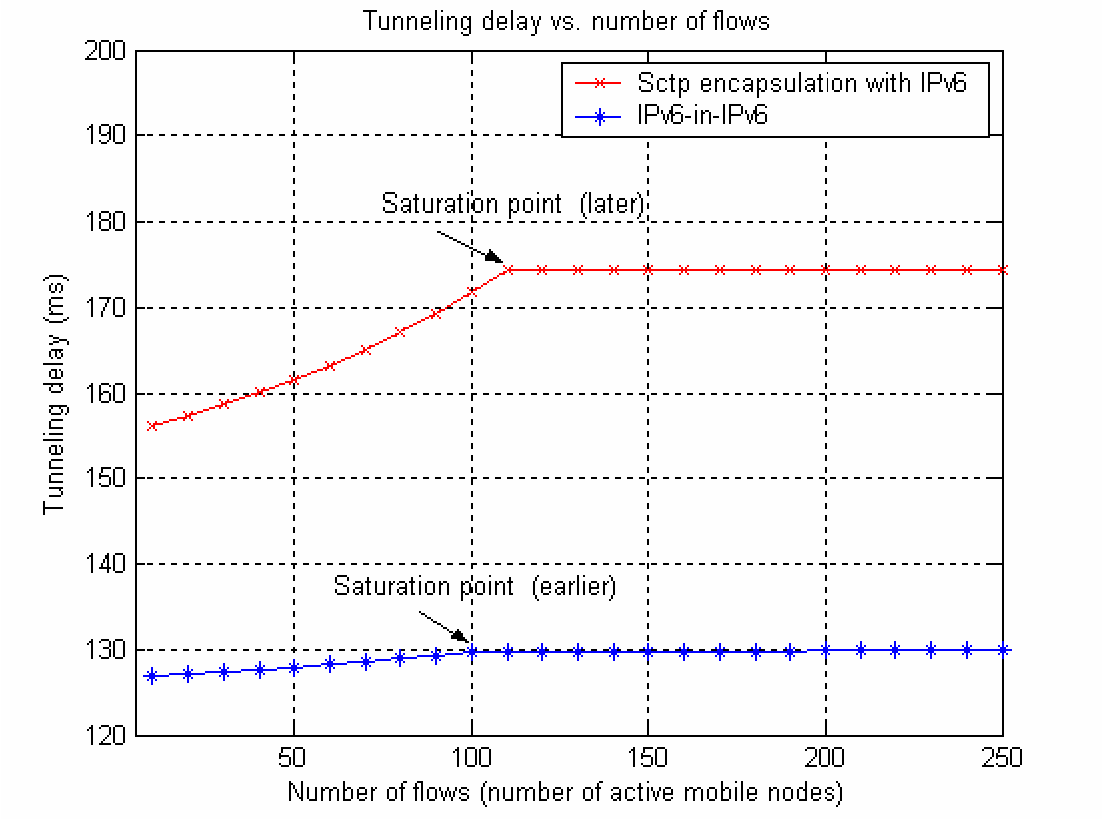
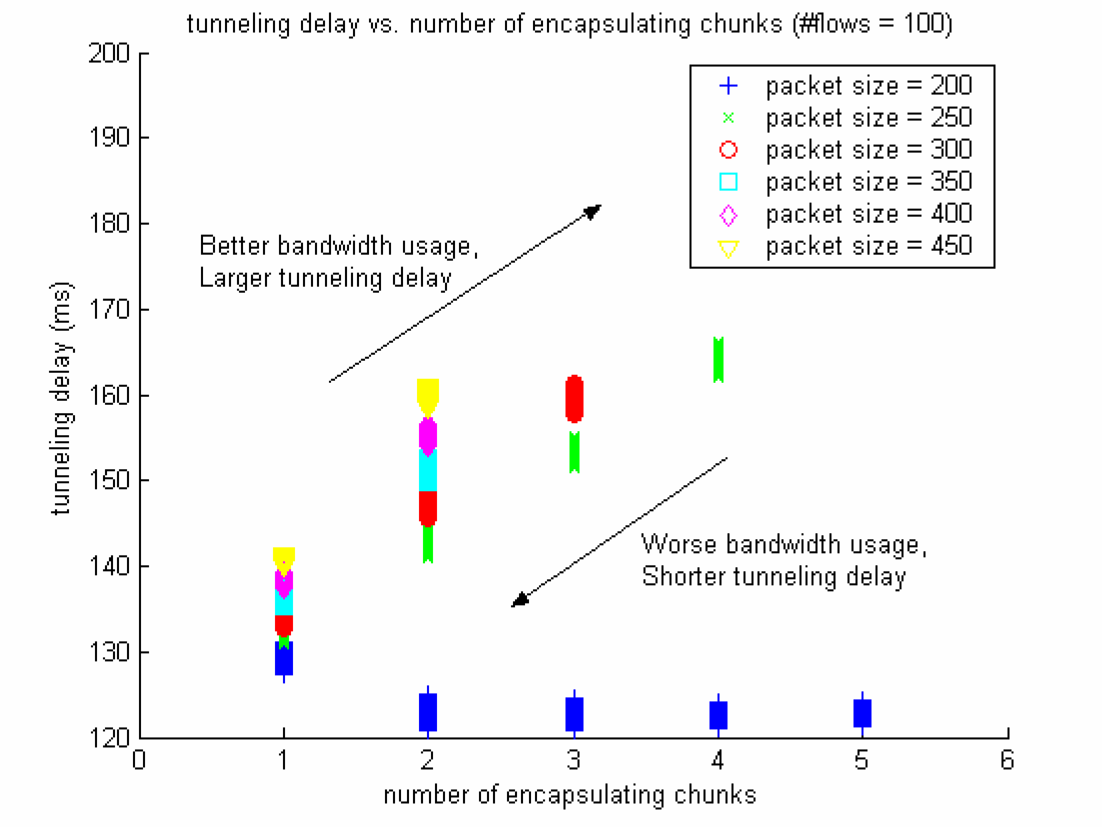


Figure 10. SCTP encapsulation and IPvt-in-IPv6 delay comparison

Of course, there is a trade-off between the bandwidth utilization and the tunneling delay. The simulation shows that, the saturation point (the point at which the network is saturated and overloading packets are dropped) of SCTP encapsulation arrives later than that of IPv6-in-IPv6 encapsulation. In contrast, the tunneling delay of SCTP encapsulation is bigger than IPv6-in-IPv6 delay because it is influenced by 2 factors: the encapsulating packet transmission delay and the buffering delay (the waiting time for many incoming packets). However the difference is not very big (about 30ms to 45 ms in our simulation for a link of 10Mbps), depends on the link speed and can be ignored.

The first factor (transmission delay) is reverse proportional to the number of encapsulated IP packets in an encapsulating SCTP packet. When the number of encapsulated packet decreases, this factor approaches the IPv6-in-IPv6 delay. Because the transmission delay for SCTP encapsulation is around the value of $\frac{52+(16+s)k}{D}+c$ and the transmission delay for IPv6-in-IPv6 is around $\frac{40+s}{D}+c$ where s is the incoming packet size, D is the link speed, c is the propagation delay, and k is the number of encapsulated packet or the number of encapsulating data chunk whereas $k_{m\tilde{a}x}=\left\lfloor\frac{MTU-frameheader-40-12}{16+s}\right\rfloor$

The second factor (buffering delay) depends on the traffic pattern. For CBR (Constant Bit Rate) flows with interval of 20ms, when the number of flows increases, the buffering delay caused by this factor converges to 0 and can be eliminated.



Because there is a trade-off relationship between the SCTP tunneling delay and the header overhead, we can dynamically adjust the parameter "number of encapsulating data chunk" to adapt to the network situation. When the link is not saturated, the delay is more important and the number of encapsulating chunk is set to the smallest value possible. When the link is overloaded, it may be reasonable to pay larger delay to reduce the header overhead, increase the bandwidth utilization and therefore reduce the packet drop rates.

Another feature that the SCTP encapsulation will offers to the NETLMM architecture is the *bandwidth aggregation*. Thanks to the multi-homing ability of SCTP, ARs and LMAs can be connected by multiple paths, armed with multiple IP addresses to increase the capacity and the reach-ability. In order to have this feature to offer service to a great number of mobile nodes, we can choose to pay either some acceptable delay which can be cured by some buffering mechanism at the mobile node or some header overhead. In return, we can have a mechanism to extend the capacity of the link LMA-AR as illustrated in the following figures:

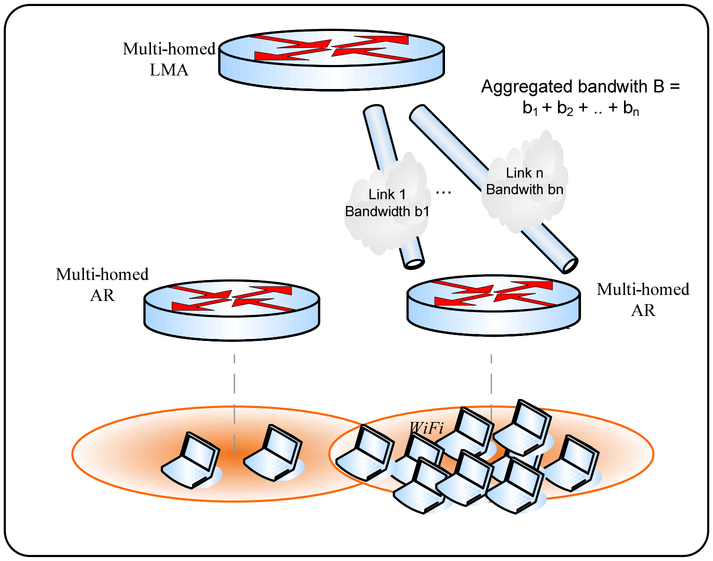


Figure 11. Bandwidth aggregation with multi-homed LMAs & ARs

When combining this feature with cluster technologies, Inter Access Point Protocol... we will have a complete robust and flexible solution to extend the capacity of the whole NETLMM architecture to increase the capacity of the nodes (LMAs, ARs), of wired links or wireless links.

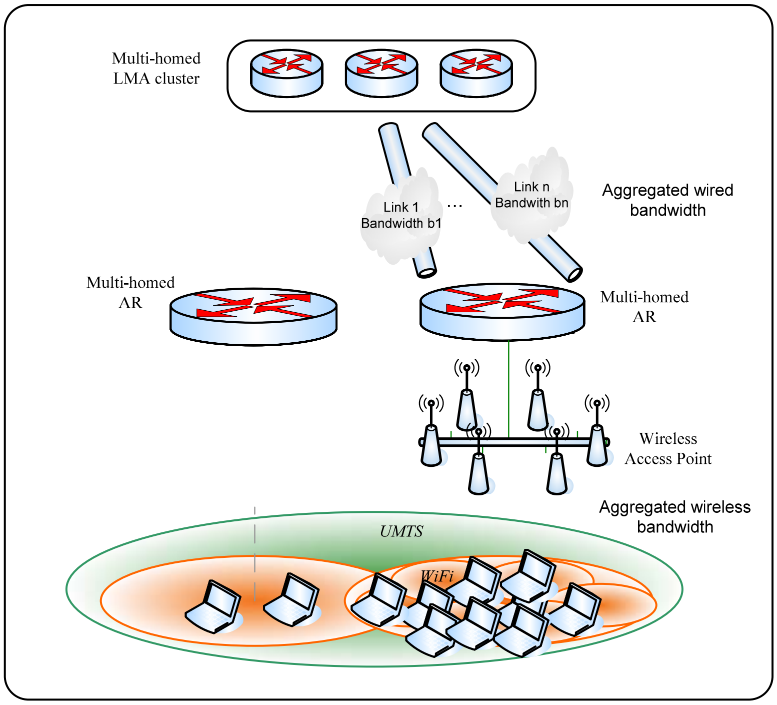


Figure 12. A flexible capacity extension mechanism

The use of SCTP/mSCTP in NetLMM may also introduce load-balancing between LMAs, between ARs of different access technologies to avoid the single point of failure problem. In this case there is an issue of how to distribute/synchronize the state between LMAs and ARs.

7. Conclusions and perspectives

The edge-based LMM (NetLMM) is currently standardized by l'IETF and is principally based on an assumption of unmodified MN. It comprises two parties. The first part defines the interface between the MN and the AR which can be realized with DNA, NDP and SEND for Stateless address auconfiguration or with the help of DHCP for Stateful address configuration. The second part defines the interface between the AR and the LMA which is still evolving. The simplest version of the AR-LMA interface is EMP (Edge Mobility Protocol). Recently a new version for this interface is proposed by Giaretta in the draft-giaretta-netlmm-dt-protocol-00.txt. The NetLMM addressing mechanism is CGA (Cryptographically Generated Address) which provides a mean to secure the mobility. The NetLMM protocol only concentrates on the control plane between entities in the NetLMM domain and not the data plane. There is something missing: The interaction between the LMM and GMM in the control plane is still not defined. It supports a one-to-many relation between the MNID and the locators. However, it doesn't mention about the use of simultaneous locators.

In fact, we can find a similarity between signaling messages of mSCTP and EMP. This allows us to have a quick and simple implementation of NetLMM by exploiting the code source of the well known SCTP protocol and its extensions. We can have a modified version of AR-LMA interface of netLMM with all required NetLMM functionalities – this approach is general and similar to any IETF drafts and it can later be replaced by a stable IETF protocol. By using mSCTP for the control plane, we can save the time to concentrate on the optimization of data plane with SCTP encapsulation or SHIM6 address translation mechanism.

The transmission of traffic in the backhaul network between AR and LMA requires some extra header information which causes header overhead. The optimization of the data plane aims at increasing the effective bandwidth (the utilization) of the backhaul network resources. As the total bandwidth of the backhaul network is bounded, header overhead reduction can improve the utilization. We can benefit a message bundling mechanism with the use of SCTP encapsulation in the data plane. This SCTP encapsulation mechanism promises a way to reduce the header overhead for small size packet networks by allowing many small packets to share the same header.

Though the delay increases but it is not very large and we can dynamically control the trade-off between the delay and the header overhead to have best performance in different situations. Moreover, we will have the *bandwidth aggregation* feature of SCTP to archive the Always Best Connected provision.

The localized mobility management is evolving with NetLMM solution, but strict requirements of NetLMM have been causing a lot of debates and changes. We assume that the SCTP protocol is the future transport protocol and try to benefit all its advantages while solving the LMM. In our future work, we will validate our idea of using mSCTP/SCTP for the multi-homing feature (bandwidth aggregation feature). We will also continue to optimize the SCTP encapsulation and archive better bandwidth utilization by defining a new chunk type with a minimal chunk header size

References

[1] G. Giaretta, K. Leung, M. Liebsch, P. Roberts, K. Nishida, H. Yokota, M. Parthasarathy, and H. Levkowetz, “Netlmm protocol,” Internet draft, draft-giaretta-netlmm-dt-protocol-00.txt (work in progress), June 2006.

[2] J. Kempf, “Problem statement for network-based localized mobility management,” draft-ietf-netlmm-nohost-ps-04.txt, June 2006.

[3] J. Kempf, “Goals for network-based localized mobility management (netlmm),” Internet draft, draft-ietf-netlmm-nohost-req-04.txt (work in progress), August 2006.

[4] J. Laganier, S. Narayanan, and F. Templin, “Network-based localized mobility management interface between mobile node and access router,” Internet draft, draft-ietf-netlmm-mn-ar-if-01.txt (work in progress), June 2006.

[5] A. A. E. Al, T. N. Saadawi, and M. J. Lee, “A transport layer load sharing mechanism for mobile wireless hosts.” in PerCom Workshops, 2004, pp. 87–91.

[6] M. R. Helsinki, “Which layer for mobility? - comparing mobile ipv6, hip and sctp.” [Online]. Available: citeseer.ist.psu.edu/711917.html

[7] S. J. Koh, M. J. Chang, and M. Lee;, “msctp for soft handover in transport layer,” Communications Letters, IEEE, vol. 8, pp. 189 – 191, March 2004.

[8] S.-J. Koh and S.-W. Kim, “msctp for vertical handover between heterogeneous networks,” Web and Communication Technologies and Internet Related Social Issues HSI 2005, 2005.

[9] T. Aura, “Cryptographically generated addresses (cga),” RFC3972, March 2005.

[10] B. S. Thomson and T. Narten, “Ipv6 stateless address autoconfiguration,” RFC2462, December 1998.

[11] J. Arkko, J. Kempf, B. Zill, and P. Nikander, “Secure neighbor discovery (send),” RFC3971, March 2005.

[12] T. Narten, E. Nordmark, and W. Simpson, “Neighbor discovery for ip version 6 (ipv6),” RFC2461, December 1998.

[13] S. Narayanan, “Detecting network attachment in ipv6 networks (dnav6),” Internet draft, draft-pentland-dna-protocol-01 (work in progress), July 2005.

[14] J. Wood and K. Nishida, “Edge mobility protocol (emp),” Internet draft, draft-wood-netlmm-emp-base-00.txt (work in progress), October 2005.

[15] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad, “Stream control transmission protocol (sctp) dynamic address reconfiguration,” draftietf- tsvwg-addip-sctp, May 2006.