Boosting NEMO with Multi-Path TCP

Pratibha Mitharwal, Christophe Lohr, Annie Gravey

Boosting NEMO with Multi-Path TCP

Pratibha Mitharwal
Institute Mines-Telecom;
Telecom Bretagne;
UMR CNRS 6074 IRISA
Technopole Brest Iroise CS 83818;
29238 Brest Cedex 3 France
pratibha.mitharwal@telecom-bretagne.eu

Christophe Lohr
Institute Mines-Telecom;
Telecom Bretagne;
UMR CNRS 6285 Lab-STICC
Technopole Brest Iroise CS 83818;
29238 Brest Cedex 3 France
christophe.lohr@telecom-bretagne.eu

Annie Gravey
Institute Mines-Telecom;
Telecom Bretagne;
UMR CNRS 6074 IRISA
Technopole Brest Iroise CS 83818;
29238 Brest Cedex 3 France
annie.gravey@telecom-bretagne.eu

Abstract—**This paper is focused on boosting network mobility Basic support protocol (NEMO) with the help of existing Multi-Path TCP (MPTCP). In this paper, the network mobility is studied in context of vehicles. The current NEMO relies on tunneling in order to use multiple available interfaces on a vehicle but does not reach the goal to provide optimum multi-homing benefits because it is unable to balance the traffic over available network interfaces. Moreover in NEMO, the nodes behind the mobile router cannot participate in multi-homing as the mobile router installed on the vehicle makes the mobility transparent to the nodes. In this paper, we propose to combine NEMO with MPTCP which enables the host nodes to participate in mobility and multi-homing and thus improving throughput, handover performance and avoiding excess tunneling.**

I. Introduction

With the evolution of technology the demand for connectivity especially ”anywhere and everywhere” has increased. Users want to be connected with the rest of the world via Internet irrespective of their current location as well as through different access technologies. The movement of the user/network invokes a change in its IP address and it is no more reachable from its previous IP address. This breaks the ongoing communication as the conventional TCP sockets are bound to network identifiers (IP addresses). To maintain the ongoing communication, the traffic has to be redirected towards user's new acquired IP address from its current network point of attachment. This traffic redirection also happens during multi-homing. In multi-homing a user has more than one active network interfaces at the same time. The conventional TCP makes it impossible to use more than one network interfaces simultaneously for an ongoing communication.

Mobility and multi-homing share one common feature i.e., having to carry a given flow over different network interfaces. During mobility it is due to change in IP address while in multi-homing it can be due to path disruption or an attempt to split the connection over all available network interfaces. The simultaneous use of all the available network interfaces can improve throughput, provide load balancing and make the system more resilient. With an increased demand for connectivity there is an increased demand of bandwidth as well as throughput. Therefore, mobility combined with multi-homing can be beneficial in improving mobility, making handover smooth and increase throughput.

This need to solve host/network mobility and multi-homing has resulted in many proposals. All the solutions for mobility provide location management of the host/network and all present pros and cons with respect to deployment, infrastructure changes, handover delay, throughput, tunneling overhead etc.

Most of the mobility protocols are network layer based to easily provide location management. In order to perform ”make before break” mobility, the mobility solution requires the transport layer informations regarding RTTs, congestion or path disruption. In absence of this collaboration between network and transport layer, the mobility protocols achieve ”break before make”, which causes extra handover delay for the flow. However, transport layer solutions need a mechanism for location management. This leaves a room for improvement.

In this paper, we present a solution for multi-homed mobile networks combining network layer approach NEMO with transport layer approach MPTCP in order to achieve better mobility support with optimum multi-homing support in terms of reduced cost, improved through put and load balancing. NEMO is used for location management and network mobility while MPTCP is used for improving multi-homing and enable host to participate in multi-homing related decision making. Moreover, the proposed combination requires minimal infrastructure changes. In section 2, we present the state of art. Section 3 describes the proposed architecture, its signaling and its benefits. Section 4 presents the comparison between classical NEMO and proposed NEMO with MPTCP. Finally we present our conclusion and future work.

II. State of Art

There have been several attempts to solve mobility and multi-homing. NEMO [1], [2], Locator/Identifier Separation protocol (LISP) [3], Identifier Locator Networking protocol (ILNP) [4] are some of the solution to support network mobility. While Mobile IP [5], Multi-Path TCP (MPTCP) [6], Host Identity Protocol [7] are some of the solutions to provide host mobility and multi-homing.

NEMO is a network layer protocol and provides location management using an anchor point in its home network which takes care of the movement of mobile network. Another network layer protocol is LISP which is the most scalable solution for mobility and multi-homing [8]. LISP introduces a new naming convention as end-point identifiers (EID) which are valid only inside the network and routing locators (RLOC) which are valid for locating the networks. In attempt to provide location management it introduces a mapping system which provides a mapping between EID and RLOCs. The routing in between networks is done by encapsulating the packets with its RLOC. In both LISP and NEMO, hosts inside the mobile network does not participate in mobility or multi-homing. Differently from NEMO and LISP, MPTCP is a transport layer protocol which attempts to provide host mobility and multi-homing support. MPTCP provides load balancing by spreading the traffic over all the available network interfaces.

There are several proposal to combine network layer approach with transport layer in attempt to improve mobility support with multi-homing. In [9], there is novel combination of MIPv6 and MPTCP which improves host mobility and multi-homing. However, it does not support mobility for a whole network. In [10], there is a combination of LISP and improved MPTCP in order to improve multi-homing benefits to the host. However, LISP requires major infrastructure changes in conventional Internet and also it introduces a new naming convention as EIDs and RLOCs. On the other hand, NEMO requires the minimal infrastructure changes into conventional Internet architecture as compared to LISP and moreover it does not introduce any new naming convention [11]. However, it needs an improvement in multi-homing support. There is also a proposal for network mobility in space [12] combining modified NEMO where its location management is replaced by dynamic DNS with SCTP. However, in this proposal mobiles nodes does not participate in mobility and multi-homing. Our aim is to investigate an approach that requires minimal infrastructure changes in the Internet. To best of our knowledge there have not been any attempt to combine NEMO with MPTCP. The following subsection contains the basic functionalities of NEMO and MPTCP and their limitations.

A. NEMO

NEMO provides a mechanism for network mobility support in the Internet, proposed in RFC 3963 [1], [2]. NEMO provides mobility supports in case of both IPv4 [13] and IPv6. In this paper, we discuss the solution for IPv6 only. The mobile network has a router inside which is known as mobile router (MR) and different kind of nodes, two of them are mobile network nodes (MNNs) and local fixed nodes (LFN) [14] which we will consider in our paper. Local fixed nodes are unable to change their network point of attachment during an ongoing communication whereas the mobile network nodes have this ability. NEMO enables MR to change its network point of attachment in the Internet and continue to receive packets destined to MNN's home address, keeping the mobility transparent from MNNs. This is being done with the help of an anchor point inside mobile network's home network known as home agent (HA). Whenever MR changes its network attachment point in the Internet, it configures a new IP address called as current Care-of-Address (CoA) with the prefix of its current network attachment point using the IPv6 address auto-configuration [15]. After the address configuration, MR creates a binding between its home address and CoA with HA. When HA receives any packet destined for MNN from a communicating node (CN) in the Internet, it encapsulates the incoming packet with MR's CoA in the destination and HA's address in the source and routes it towards MR over the tunnel. Once MR receives the packet, it decapsulates the packet and routes it inside the mobile network. When MNNs receive the packet, they find their home address in the source address of the incoming packet. This way the mobility is kept transparent from MNNs. The above explained functionality of NEMO can be shown by Fig. 1.

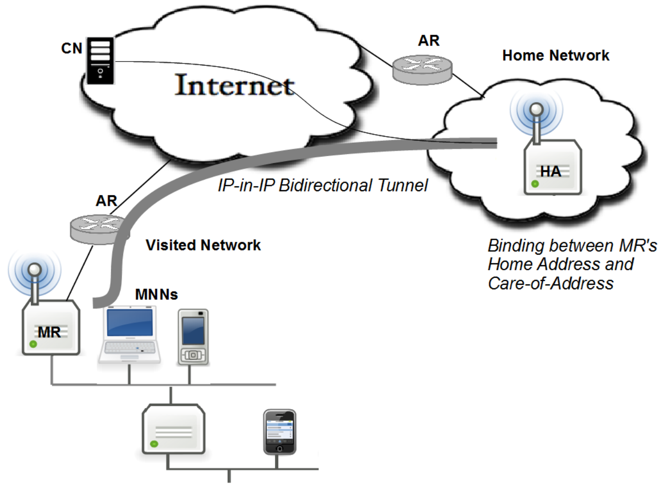

Fig. 1. NEMO Basic Support Protocol

The multi-homing support enables mobile network to have one or more MR, HA and MNNs thus different combinations [16]. Here in this paper, we are going to consider the mobile network with one MR, one HA and more than one MNNs. NEMO has been extended to provide multi-homing support to mobile networks [17] with Multiple Care-of-Address registration (RFC 5648 [18]) and flow binding extension (RFC 6089 [19]).

B. Multi-Path TCP

Multi-Path TCP protocol (MPTCP) is an extension of transmission control protocol (TCP) proposed in RFC 6824 [6], which is designed to enable simultaneous use of multiple available network interfaces on a host. MPTCP is backward compatible to TCP therefore, it uses the standard socket API used by most Internet applications which makes it compatible to the current applications and network [20]. The use of multiple paths between source and destination provides reliability, flexibility, fault tolerance and efficiency. In case of any failure, MPTCP can divert the traffic towards the other active paths. Compared to TCP, MPTCP does not add any overhead to data or tunnels. It just need an additional signaling for starting a communication. This is also needed for adding or removing the sub-flows due to IP address changes. MPTCP uses ”make before break” method which is beneficial in providing seamless mobility and smooth handover. The Fig. 2 shows a mobile host having multiple communication paths with a remote host/server using MPTCP.

An MPTCP session start the same way as TCP with SYN flag set, with one addition in the SYN packet i.e., MP_CAPABLE option. If the receiving host supports MPTCP, it adds the option MP_CAPABLE in SYN-ACK reply. These two hosts create a cryptographic token/key for the secure communication. Finally the connection initiator host sends an ACK packet to the receiving host with MP_CAPABLE option. Considering multi-homing or mobility scenario a host

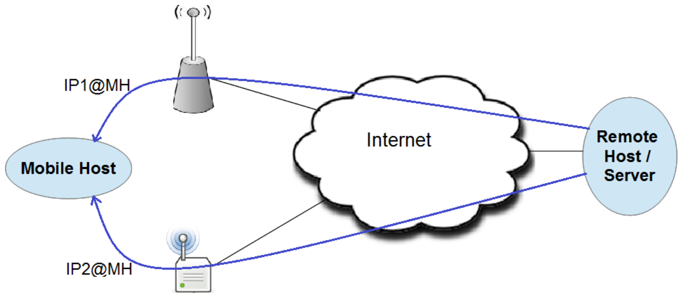

Fig. 2. Multi-Path TCP

either has more than one IP address or it configures new IP address(es) by connecting with new network attachment point(s). Once the communication is established in between the hosts, the other exiting or new acquired IP address can be added into the list of IP addresses of remote host using ADD_ADDR option. Afterwards, the new sub-flow can be created using MP_JOIN option through the other IP address. During mobility, the loss of a connection through one network attachment point makes one IP address unreachable. This lost IP address can be removed from the remote host's IP address list with REMOVE-ADDR option and then the sub-flows using this IP are removed too.

The MPTCP provides coupled congestion control [21], [22]. The sub-flows maintain their own congestion window and execute the slow start algorithm. Therefore, MPTCP can optimally use the network resources by redirecting the traffic towards the non-congested paths. This feature of MPTCP helps to provide resilience, higher throughput and manage the increased demand of bandwidth more efficiently.

For the users that do not support MPTCP, there are some proposals to use MPTCP proxy implemented on end host or use a global anchor point [23]. If the user is comfortable with implementing proxies on end host, the lightweight proxy is proposed. It is based on packet filter implemented in mobile hosts. Some filter based solutions such as IPsec and tunneling are already being used, which makes it easy to deploy MPTCP on the end host. After implementing the proxies, both the hosts can be benefited with multi-homing benefits.

C. *Limitations of NEMO and MPTCP*

NEMO introduces tunneling overhead in order to provide mobility as the packets have to be routed through tunnels between HA and MR. In NEMO, all the incoming and outgoing traffic has to pass through HA. If there is any disruption on HA, all the incoming and outgoing traffic will be disrupted.

NEMO tries to use all the existing IP addresses simultaneously with the help of its extensions. However, this requires additional tunnels between HA and MR. The flow binding extension of NEMO tries to manage the traffic distribution of flows over several existing paths. This is an effort to provide an equal distribution of flows over all available interfaces. However, even with flow binding extension it cannot provide a full load balancing and optimum benefits of multi-homing. Thus, there is a requirement of the improvement considering mobility and multi-homing with NEMO to reach the multi-homing goals optimally and reduce tunneling overhead.

The mobile network nodes behind MR cannot take part in multi-homing with NEMO. If mobile network nodes take part in the multi-homing the load from MR can be reduced and each user can manage their own connections. Thus, there is a requirement for an improvement considering mobility and multi-homing with NEMO to reach the multi-homing goals optimally.

MPTCP provides ”make before break” mobility to the hosts but it does not provide network mobility. MPTCP does not have its own location management to support mobility of the host. There are several proposal for location management with MPTCP using dynamic DNS [24] or a rendez-vous mechanism. However, these proposals work only for host mobility but not for mobile networks. Therefore, to solve network mobility and multi-homing with MPTCP, there is a requirement of location management mechanism.

III. Network Mobility with Host Multihoming

We propose to choose NEMO for location management and MPTCP for multi-homing. There can be other choices for location management like dynamic DNS or a rendez vous server. To provide location management for mobile networks, these proposals need some addtional functionalities to be introduced. As the use of dynamic DNS lacks in providing a seamless mobility for local fixed nodes due to their inability to change their network point of attachment for an ongoing connection. NEMO is the only solution that will be able to provide seamless mobility to local fixed nodes with the help of its HA route. Also in case of non-friendly visited networks (NAT, paranoiac, firewall, etc.), dynamic DNS cannot be used for incoming traffic [21]. This would require an rendez-vous mechanism to open the connection simultaneously from both the ends for e.g., VPN (with their tunnels). Whereas, NEMO already uses tunnels which can be used for incoming traffic in non-friendly visited networks. NEMO provides better location management than dynamic DNS in mobile networks.

There can be other choices for multi-homing as well for eg. CMT-SCTP [25]. However, unlike MPTCP, SCTP is not transparent to the applications. Moreover, MPTCP performs significantly better than SCTP [26]. Therefore, the combination of NEMO and MPTCP would be a better choice than any other rendez-vous mechanism or dynamic DNS with MPTCP in order to solve network mobility having local fixed nodes and mobile network nodes providing better multi-homing support.

A. *Description of the proposed Architecture*

In this section, we describe the procedure to combine NEMO with MPTCP protocol. Their integration can be done without any major modifications to NEMO and MPTCP.

In our proposal, NEMO is being used for the location management of mobile network for incoming traffic during connection initiation and afterwards the MPTCP handles the mobility and multi-homing. This proposal enables MNNs to participate in mobility and multi-homing. For implementing this, the MNNs need to be aware of mobility which requires two minor changes in MR. Firstly, the MR needs to advertise its current network prefixes or Care-of-prefixes to MNNs. After receiving the current network prefixes MNNs can configure and add the new acquired IP address to their interfaces.

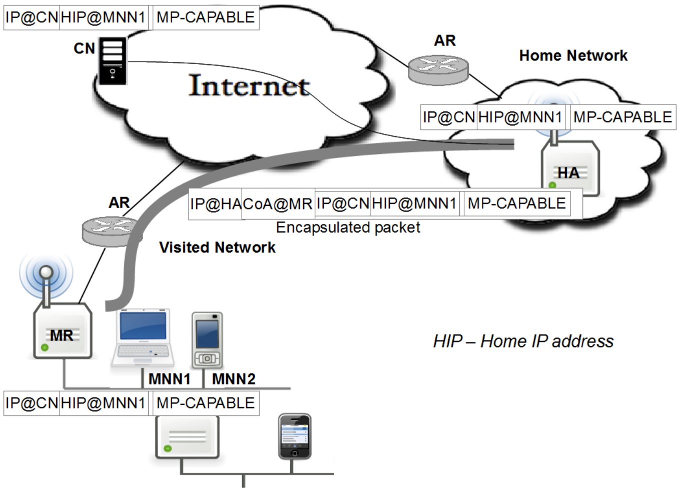

Fig. 3. Connection initiation in proposed architecture

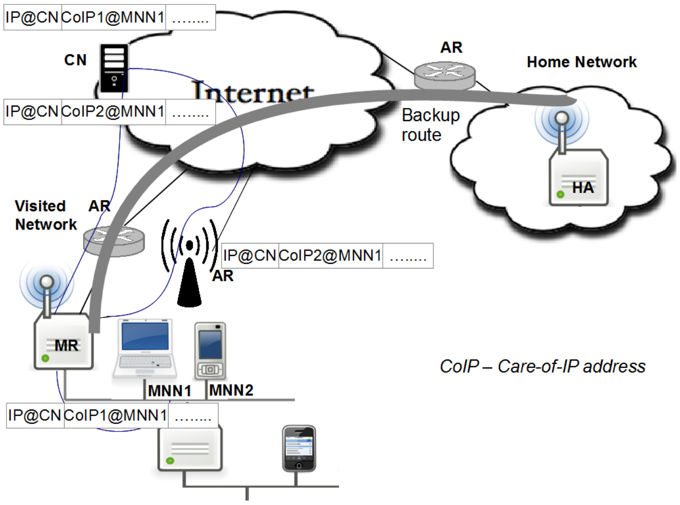

Fig. 4. Communication after connection establishment in proposed architecture

Secondly, MR should be able to differentiate between the packet flows. The packet flow with MNN's home IP address need to be sent through HA-MR tunnel whereas the packet flow having MNN's new acquired care-of-address need to be routed towards Internet.

In order to illustrate the key points we consider the following scenario. When a mobile network is outside its home network and a communicating node (CN) initiates a communication with the node which is behind MR, the communication has to pass through the HA because CN is not aware of MNNs current location in the Internet. Home agent then encapsulates the packets and forwards them towards MR's current care-of-address which is exactly the same as NEMO shown in Fig. 3.

During the connection establishment, the mobile network node or communicating node adds MP_CAPABLE option inside the TCP packet. If both nodes natively support MPTCP then they establish an MPTCP connection and exchange all the existing IP addresses. If any node does not support MPTCP, it can install a lightweight proxy as explained in the state of art of MPTCP. Once the communication is established the mobile network node can set the HA path as BACKUP path with help of MP_PRIO option and create sub flows using the rest of the IP addresses. Mobile router should be able to route the packets directly using its current care-of-address rather than through HA.

Whenever during mobility, the mobile network attaches to a new point in the Internet, it receives a new prefix. Using this prefix, it creates a care-of-address and updates the binding with HA. It then advertises the new prefix to the MNNs. Once the nodes receive new prefix they can configure the new IP address and communicate it with the communicating node using MPTCP and create another sub-flow as shown in Fig. 4. In presence of multiple CoAs on MR, HA have to make a choice among available interfaces for forwarding the connection initiation packets. This can be solved by implementing policies in between HA and MR with help of flow binding extension [27] of NEMO.

As MPTCP uses "make before break", the proposed approach makes handover smooth compared to classical NEMO. In the proposed approach, NEMO is used only for starting the communication from the nodes in the rest of the Internet to a MNN. If MNN wants to initiate a connection it can use its current care-of-address and then MPTCP handles the further communication.

B. Signaling

For explaining the signaling in the proposed approach, we consider two different scenarios.

- When MNN initiates the communication with CN i.e., for outbound traffic
- When CN initiates the communication with MNN, i.e., for inbound traffic.

1) Signaling for Outbound Traffic: In outbound traffic, MNN initiates a communication with CN. To initiate the communication, MNN creates a SYN packet with MP_CAPABLE option and routes it toward MR using its CoIP@MR-AR1. MR then routes this packet towards the Internet. CN receives the packet and routes the reply SYN-ACK packet towards MNN's CoIP@MR-AR1. On reception of SYN-ACK, MNN sends an ACK packet and the MPTCP connection is established. After the connection establishment, MNN can share its existing CoIPs. MNN has an option to share its Home IP address with the CN to keep this path as a back up path, generally it is not required. If mobility occurs in between, then MNN receives a new prefix from MR and configures a new CoIP. It then shares this new CoIP with CN and creates another sub-flow. If the communication is lost with the previous access network, MNN sends a request to remove the previous IP address and the communication continues through other sub-flows. Fig. 5 illustrates the above explained signaling.

The proposed approach shows an expected gain in case of outbound traffic from mobile network nodes compared to classical NEMO as the traffic does not flow anymore through the tunnels between HA and MR. For outbound traffic from local fixed nodes, NEMO will be used and will work in the

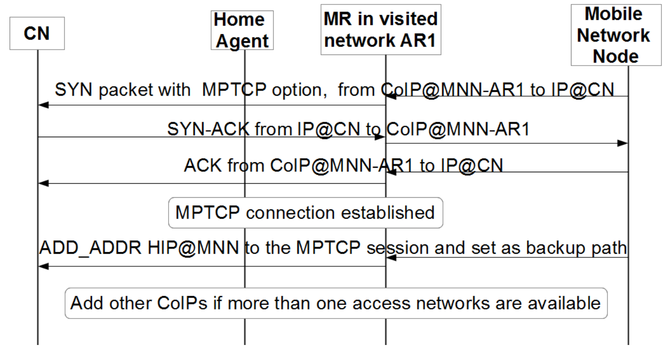
CN
Home Agent
MR in visited network AR1
Mobile Network Node
SYN packet with MPTCP option, from CoIP@MNN-AR1 to IP@CN
SYN-ACK from IP@CN to CoIP@MNN-AR1
ACK from CoIP@MNN-AR1 to IP@CN
MPTCP connection established
ADD_ADDR HIP@MNN to the MPTCP session and set as backup path
Add other CoIPs if more than one access networks are available

Fig. 5. MPTCP connection establishment for outbound flow in proposed architecture

classical way. Therefore, the proposed approach reduces cost of tunnels and improves load balancing thus throughput for mobile network nodes.

2) Signaling for Inbound Traffic: To illustrate the signaling for inbound traffic, we consider the scenario where mobile network is on move and currently connected to access router 1 (AR1). Once it is connected with AR1, it creates a binding with its HA for its current care-of-address. It then advertises the new prefix to MNN and it then configures and adds the new IP (CoIP@MNN-AR1) to one of its interfaces. When any communicating node from the outside Internet initiates a communication with MNN, the SYN segment packet is generated with MP-Capable option for MNN's home IP address and routed towards its home network. In the home network, HA intercepts the packets and finds a binding entry for MNN's MR. Afterwards, HA encapsulates the packets using care-of-address of MR@AR1 (CoIP@MR-AR1) and forwards the packet towards mobile network's visiting network. Upon the reception of the packet, MR decapsulates it and forwards it inside the mobile network. MNN receives the packets and create a SYN-ACK packet segment for IP@CN and route it towards MR. MR encapsulates this SYN-ACK packet and send it to HA. On reception of the packet from MR, HA decapsulate it and route it towards CN's network. CN receives the packet and generates reply as an ACK packet and the packet takes the same route as before. On reception of ACK packet, the MPTCP connection is established. After the connection establishment, MNN sends a request to ADD-ADDR providing CoIP@MNN-AR1 and create a sub-flow using this IP. Once the new sub-flow is added to the ongoing communication MNN sets the HA-MR route as backup path and continues to communicate through its care-of-address. After this step a tunneling overhead is gained with exchange of each packet for the ongoing communication. Fig. 6 illustrates the above explained signaling.

The above explained scenario is valid when MNNs and MR are connected to single access network. There can be multiple available access networks. In that scenario, MNN can create new sub-flows by adding other available CoIP address to the ongoing communication with CN and communicate using all the sub flows.

The only difference between outbound and inbound traffic signaling is during connection initiation. As in case of inbound traffic the proposed approach use tunnels for connection initiation. After the connection initiation both the scenarios show similar expected gain in terms of reduces tunneling overhead and improved load balancing thus improved throughput.

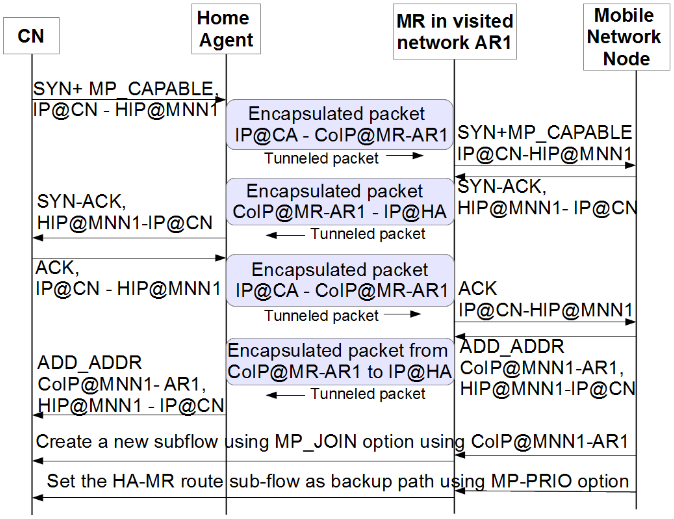
CN
Home Agent
MR in visited network AR1
Mobile Network Node
SYN+ MP_CAPABLE, IP@CN - HIP@MNN1
Encapsulated packet IP@CA - CoIP@MR-AR1
Tunneled packet
SYN+MP_CAPABLE IP@CN-HIP@MNN1
SYN-ACK, HIP@MNN1-IP@CN
Encapsulated packet CoIP@MR-AR1 - IP@HA
Tunneled packet
SYN-ACK, HIP@MNN1- IP@CN
ACK, IP@CN - HIP@MNN1
Encapsulated packet IP@CA - CoIP@MR-AR1
Tunneled packet
ACK IP@CN-HIP@MNN1
ADD_ADDR CoIP@MNN1- AR1, HIP@MNN1 - IP@CN
Encapsulated packet from CoIP@MR-AR1 to IP@HA
Tunneled packet
ADD_ADDR CoIP@MNN1-AR1, HIP@MNN1-IP@CN
Create a new subflow using MP_JOIN option using CoIP@MNN1-AR1
Set the HA-MR route sub-flow as backup path using MP-PRIO option

Fig. 6. MPTCP connection establishment for inbound flow in proposed architecture

C. Benefits and Use cases

For describing the benefits of proposed architecture, we have to reconsider the limitations of NEMO and show that the combination of NEMO and MPTCP can improve these limitations. To perform an ideal mobility or "make before break" mobility it requires some path related information from transport layer as talked earlier. The proposed approach is a collaboration between network and transport layer protocols. Therefore, MPTCP helps NEMO to provides "make before break" mobility with smooth handover.

All the traffic having to go through HA makes NEMO less fault tolerant. While in the proposed approach, only the connection initiation takes place over the HA-MR bidirectional tunnel. Afterwards, all the traffic flows through MR as it flows through any normal router towards the Internet. This removes the tunneling overhead of the data packet exchange after the connection establishment between HA and MR and system becomes more fault tolerant than HA-MR route.

In order to support multi-homing with the help of its multiple care-of-address registration (MCoA) extension, NEMO adds more tunnels between HA and MR. While in the proposed approach, MR does not have to perform multi-homing. MPTCP takes over after the connection is established and the rest of the communication happens in a similar way to that of the Internet. The use of MPTCP reduces these additional tunnels and provides multi-homing benefits to the host.

The flow binding extension of NEMO attempts to provide an equal distribution of flows over all available interfaces but optimum load balancing cannot be achieved. While in the proposed scenario, MPTCP helps to achieve an optimum load balancing over all the available network interfaces.

The network nodes behind Mobile router cannot take part into multi-homing in NEMO. While the proposed approach provides a way for MNNs to participate in multi-homing with help of MPTCP and helps to improve the throughput of the whole system. This participation also gives an opportunity to the user to decide the communication mode in terms of cost efficiency, energy efficiency, or best availability of an interface etc.

There can be a scenario where the communicating node is very near to MR. An example of such a scenario in the real world situation can be a boat/ship which resides in France (its home network) and currently sailing near Australian coast. A random user from Australia would want to communicate/transfer a file or video with/to a node on this boat/ship. This will cause the traffic to travel through its home network i.e., France. It can be improved by the proposed architecture as the data transfer takes normal routing path instead of its HA path.

Considering the scenario where two mobile network nodes inside boats of different companies (i.e. different network operators) want to communicate or exchange data with NEMO, the traffic has to pass through both of their HAs and then tunneled towards MRs. The proposed approach shows a gain in this scenario as after the connection establishment these two nodes can communicate independent of HAs-MRs route thus no more tunnels. Moreover, the boats can have several connection to Internet via e.g., satellite, LTE, 3G, WiFi. In this scenario, switching between different terminals becomes difficult. In some circumstance there can be a technician in board to manually perform the commutation between link terminals, set up IP and routing parameters (e.g., on scientific boats). But this is hardly the case (e.g., on merchant vessels). The proposed approach takes care of the switching between terminals.

Some other use case scenarios can be considered as in case of boats, mail-servers can receive any amount of data in real time without requiring to pass through tunnels or by polling a cache server. In present time, some commercial ship vessels have some limitations of the file size in the on-board mail-servers. This file-size constraint can be removed with the help of the proposed architecture and on-board email servers can be reached at any time instead of polling.

The proposed approach does not need any major modification in already implemented NEMO and MPTCP. This makes its deployment manageable. For UDP connections or in absence of MPTCP proxies, tunnels are still available as a fall-back solution. For non-TCP connections in real-time traffic such as VoIP, IETF is working on standardizing multi-path extension for RTP [21]. This can also be used with NEMO.

The tunnels of NEMO may have increased overhead in normal scenarios but they are beneficial for the incoming traffic in non-friendly visited networks for e.g., NAT, firewall etc. and for providing seamless mobility to the local fixed nodes inside a mobile network. NEMO's tunnels are also convenient natural fall-back solution in the scenario when there is no MPTCP support. Moreover, the proposed solutions helps to avoid tunnels when it is not necessary. This makes the system more flexible and more robust. Therefore, MPTCP with NEMO helps to provide better network mobility with optimum multihoming benefits.

TABLE I. CLASSICAL NEMO VS. NEMO WITH MPTCP

Parameters	Classical NEMO	NEMO with MPTCP
Routing	Inefficient	Efficient
Throughput	Low	High
Round Trip Time	High	Low
Signaling Cost	Low	High
Tunneling overhead	Always	Only for incoming Traffic, local fixed nodes and in non-friendly visited network

IV. COMPARISON BETWEEN CLASSICAL NEMO AND NEMO WITH MPTCP

In this section, we compare the features of NEMO and our proposal NEMO with MPTCP in Table I based on following parameters:

- **Routing**: In classical NEMO, the traffic has to pass through HA and then it is tunneled towards MR which is inefficient. Whereas, in the proposed approach, the traffic follows a classical Internet path which makes the routing more efficient. The HA-MR route is used only for signaling for incoming traffics and for providing mobility to LFNs.
- **Throughput**: The throughput of the system can be almost doubled by using MPTCP compared to TCP in mobile scenario [21]. Therefore, we can expect an improved throughput from the proposed architecture compared to classical NEMO.
- **Round Trip Time**: Compared to classical NEMO, in proposed approach the traffic takes classical Internet routing path rather than traffic through MR's home network. Thus the proposed architecture improves RTT also.
- **Signaling Cost**: The classical NEMO will have signaling cost required for NEMO (SC_{NEMO}). Whereas, the proposed architecture will have NEMO signaling cost with additional MPTCP signaling cost for adding or removing sub-flows during mobility $SC = SC_{NEMO} + SC_{MPTCP}$.
- **Tunneling overhead**: The proposed architecture significantly reduces tunneling cost compared to classical NEMO. As in classical NEMO, the whole traffic takes HA-MR route which adds bi-directional encapsulation overhead. Whereas in the proposed architecture, the traffic is routed directly towards Internet from MR rather than towards HA.

V. FUTURE WORK

In future, we would be evaluating the gain obtained using test-bed implementation of the proposed approach shown in Fig. 7. MNN and CN are MPTCP enabled here. We will try to calculate throughput, handover latency, round trip time (RTT) and retransmission during hand-off by downloading some files from CN to MNN and vice-versa, with the help of proposed approach and compare this scenario with classical NEMO. We will also calculate RTT for the outgoing signaling and compare this with the classical NEMO signaling. We can capture the packets using wire-shark and plot the graphs between packet captured versus time to show the difference between the

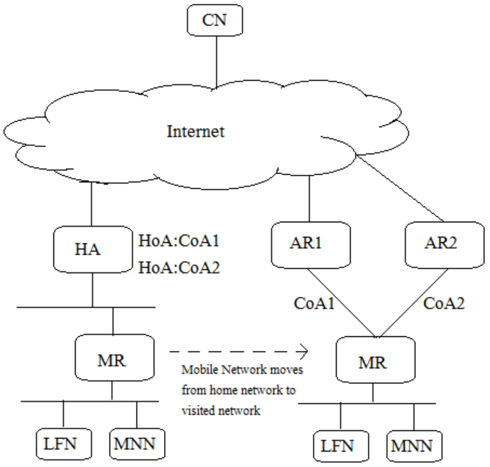

Fig. 7. Test-bed implementation architecture

proposed and classical throughput, handover latency, RTT and retransmission during hand off. Through these graphs we will be able to show that NEMO works better with MPTCP.

VI. Conclusion

In this paper, we proposed an architecture for providing better multi-homing support in mobile networks by combining NEMO with MPTCP. The proposed architecture requires some minor changes in functionalities of mobile router in NEMO and none in MPTCP. There are two changes that need to be implemented in the MR. First, it should advertise its care-of-prefixes to mobile network nodes for them to configure their own care-of-address. Second, it should be able to route the packets directly using MNNs care-of-addresses instead of routing them towards HA through the tunnel. This solution limits tunneling overhead. The established connection will survive as long as there is atleast one available care-of-address. Moreover, the performance of the communication is surely increased by using MPTCP when compared to TCP [28]. Therefore, this novel combination of NEMO and MPTCP provides a better network mobility with improved multi-homing support compared to classical NEMO in terms of reduced tunneling overhead, increased throughput and improved load balancing.

Acknowledgment

The research leading to these results has received funding from the European Community Seventh Framework Program FP7/2013- 2015 under grant agreement no. 317762 COMBO project.

References

[1] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, “Rfc 3963 network mobility (nemo) basic support protocol,” 2005.

[2] C. J. Bernardos, A. De La Oliva, M. Calderón, D. von Hugo, and H. Kahle, “Nemo: Network mobility. bringing ubiquity to the internet access,” *demonstration at IEEE INFOCOM*, 2006.

[3] D. Farinacci, D. Lewis, D. Meyer, and V. Fuller, “The locator/id separation protocol (lisp),” 2013.

[4] S. Bhatti and R. Atkinson, “Identifier-locator network protocol (ilnp) architectural description,” 2012.

[5] C. E. Perkins, “Mobile networking through mobile ip,” *Internet Computing, IEEE*, vol. 2, no. 1, pp. 58–69, 1998.

[6] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, “Rfc 6824: Tcp extensions for multipath operation with multiple addresses,” 2013.

[7] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, “Host identity protocol,” *RFC5201, April*, 2008.

[8] D. Farinacci, D. Lewis, D. Meyer, and V. Fuller, “The locator/id separation protocol (lisp),” 2013.

[9] M. Bagnulo, P. Eardley, A. Ford, A. García-Martínez, A. Kostopoulos, C. Raiciu, and F. Valera, “Boosting mobility performance with multipath tcp,” in *Future Network and Mobile Summit, 2010*. IEEE, 2010, pp. 1–8.

[10] M. Coudron, S. Secci, G. Pujolle, P. Raad, and P. Gallard, “Cross-layer cooperation to boost multipath tcp performance in cloud networks,” in *Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on*. IEEE, 2013, pp. 58–66.

[11] P. Mitharwal, C. Lohr, and A. Gravey, “Survey on network interface selection in multihomed mobile networks,” in *Advances in Communication Networking*. Springer, 2014, pp. 134–146.

[12] P. K. Chowdhury, M. Atiquzzaman, and W. Ivancic, “Sinemo: An ip-diversity based approach for network mobility in space,” in *Space Mission Challenges for Information Technology, 2006. SMC-IT 2006. Second IEEE International Conference on*. IEEE, 2006, pp. 7–pp.

[13] K. Leung, G. Dommety, V. Narayanan, and A. Petrescu, “Network mobility (nemo) extensions for mobile ipv4,” Tech. Rep., 2008.

[14] T. Ernst, “Network mobility support terminology,” *Network*, 2007.

[15] T. Narten, S. Thomson, and T. Jinmei, “Ipv6 stateless address autoconfiguration,” 2007.

[16] C. Ng, T. Ernst, E. Paik, and M. Bagnulo, “Analysis of multihoming in network mobility support,” Tech. Rep., 2007.

[17] M. S. Hossain, M. Atiquzzaman, and W. Ivancic, “Performance evaluation of multihomed nemo,” in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 5429–5433.

[18] R. Wakikawa, V. Devarapalli, G. Tsirtsis, T. Ernst, and K. Nagami, “Multiple care-of addresses registration,” *Work in Progress*, 2006.

[19] G. Tsirtsis, H. Soliman, N. Montavont, G. Giaretta, and K. Kuladinithi, “Flow bindings in mobile ipv6 and network mobility (nemo) basic support,” *IETF request for comments*, vol. 6089, 2011.

[20] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, “Exploring mobile/wifi handover with multipath tcp,” in *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*. ACM, 2012, pp. 31–36.

[21] C. Raiciu, D. Niculescu, M. Bagnulo, and M. J. Handley, “Opportunistic mobility with multipath tcp,” in *Proceedings of the sixth international workshop on MobiArch*. ACM, 2011, pp. 7–12.

[22] C. Raiciu, M. Handley, and D. Wischik, “Coupled congestion control for multipath transport protocols,” *RFC6356, October*, 2011.

[23] G. Hampel, A. Rana, and T. Klein, “Seamless tcp mobility using lightweight mptcp proxy,” in *Proceedings of the 11th ACM international symposium on Mobility management and wireless access*. ACM, 2013, pp. 139–146.

[24] Y. Sun, Y. Cui, W. Wang, T. Ma, Y. Ismailov, and X. Zheng, “Mobility support in multi-path tcp,” in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*. IEEE, 2011, pp. 195–199.

[25] J. R. Iyengar, P. D. Amer, and R. Stewart, “Concurrent multipath transfer using sctp multihoming over independent end-to-end paths,” *Networking, IEEE/ACM Transactions on*, vol. 14, no. 5, pp. 951–964, 2006.

[26] M. Becke, H. Adhari, E. P. Rathgeb, F. Fa, X. Yang, and X. Zhou, “Comparison of multipath tcp and cmt-sctp based on intercontinental measurements,” in *Global Communications Conference (GLOBECOM), 2013 IEEE*. IEEE, 2013, pp. 1360–1366.

[27] T. Ropitault and N. Montavont, “Implementation of flow binding mechanism,” in *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*. IEEE, 2008, pp. 342–347.

[28] M. Scharf and A. Ford, “Multipath tcp (mptcp) application interface considerations,” RFC 6897, March, Tech. Rep., 2013.