Self-healing and optimizing of the HIP-based M2M overlay network

Amine Dhraief[1], Khalil Drira[2] and Abdelfettah Belghith[1]

[1]HANA Research Group
University of Manouba, Tunisia
{*first.last*}@*hanalab.org*
[2]LAAS-CNRS, France
Univ. de Toulouse, France
{*first.last*}@*lass.fr*

Abstract

Machine-to-Machine (M2M) paradigm is a novel communication technology under standardization at both the ETSI and the 3GPP. It involves a set of sensors and actuators (M2M devices) communicating with M2M applications via M2M gateways, with no human intervention. For M2M communications trust and privacy are key requirements. This drove us to propose a host identity protocol (HIP) based M2M overlay network, called HBMON, in order to ensure private communications between M2M devices, M2M gateway and M2M applications. In this paper, we first propose to add the self-healing capabilities to the M2M gateways. We enable at the M2M gateway level the REAP protocol, a failure detection and locator pair exploration protocol for IPv6 multihoming nodes. We also add mobility management capabilities to the M2M gateway in order to handle M2M devices mobility. Furthermore, in this paper we add the self-optimization capabilities to the M2M gateways. We also modify the REAP protocol to continuously monitor the overlay paths in order to always select the best available one in term of RTT. We implement our solution on the OMNeT++ network simulator. Results highlight the novel gateway capabilities: it recovers from failures, handle mobility and always select the best available path.

1 Introduction

Embedded systems such as sensors, smart meters and smart cards are experiencing a tremendous proliferation. Several market forecast predict that the number of these devices will soon outnumber the people on earth. According to the Wireless World Research Forum (WWRF), by 2017 we will have 7 trillion wireless devices serving 7 billion people [20]. Juniper Networks predicts that in 2015, the number of connections between embedded equipments will reach over 500 millions [11]. Machine-to-machine (M2M) communication is considered to be an adequate framework to handle the communication between these embedded systems and their corresponding applications. M2M communication is a novel communication technology under standardization at both the European Telecommunications Standardization Institute (ETSI) [10] and the 3rd Generation Partnership Project (3GPP) [19]. M2M communication is based on an autonomous communication between sensors/actuators and correspondent application over the Internet. The M2M architecture introduces a new level of indirection between the sensors/actuators and the application namely the M2M gateway. The M2M gateway aggregates data packets received form sensors and sends them to the M2M application. It generally communicates with M2M devices via short range communication technologies.

Internet is based on the well-known paradigm: ”keep-it simple in the middle, smart at the edge” [18], which survived for the last four decades. Nonetheless, the M2M gateway breaks this paradigm, instead of ”keep-it simple in the middle, smart at the edge”, it shifts the intelligence towards the middle, at the access level. Hence, M2M technologies leads us to imagine and conceive a novel inter-networking architecture. One of the key requirement of M2M communications is the privacy of the collected information. This requirement drove us to build an M2M overlay network over the Internet based on the Host Identity Protocol (HIP) [14, 15], named HBMON (HIP-based M2M Overlay Network) [6]. In this previous work, we have addressed the formation and the maintenance of the overlay.

In this paper, we propose to add the autonomic management of the overlay. We mainly focus on the self-healing and self-optimization autonomic properties. We enable at the M2M gateway level the REAP protocol, a failure detection and locator pair exploration protocol for IPv6 multihoming nodes [1]. Thus, in our overlay, M2M gateways are able to autonomically detect failures of the overlay links and recover from them. We also add to

the M2M gateway the mobility management capabilities. M2M devices mobility can be considered as a failure of the first hop and a failure detection and recovery protocol may handle M2M device mobility. Nonetheless, we need to use a specific mobility support to efficiently handle M2M device mobility as we demonstrated in [7]. In our design, M2M gateways are also able to monitor the available overlay paths and dynamically select the best path in term of Round Trip Time (RTT). We implement our solution on the OMNeT++ network simulator. Results show that our solution is able to detect overlay link failures and recover from them. It is also able to self-optimize the selection of the overlay paths.

The remaining of this paper is organized as follows. Section 2 gives an overview of our previous work HBMON [6], then it details the REAP protocol, finally it focuses on the mobility support in the HIP protocol. Section 3 highlights our contribution; namely the self-healing and optimizing of the HIP-based M2M overlay network. Section 4 presents our simulation results. Section 5 concludes the paper.

2 Related works

In this section, we first give an overview of our previous work on M2M overlay network namely the HIP-based M2M overlay network. Then we detail REAP, a failure detection and locator pair exploration protocol for IPv6 multihoming nodes [1]. Finally, we focus on the mobility support in the HIP protocol.

2.1 The HIP-based M2M overlay network

An M2M communication involves an M2M device communicating with an M2M application via M2M gateways, with no human intervention. The first "*Machine*" in a Machine-to-Machine communication is a device embedding a sensor and an actuator. The second "*Machine*" is a device which processes the collected information from the sensor and according to these information may remotely control the actuator. The "*to*" refers to the M2M end-to-end communication network connecting the two machines. M2M devices upload their traffic to an M2M Gateway which aggregates data collected from several M2M devices and sends them to a corresponding M2M gateway or to an M2M application. The M2M application has a middleware layer where data collected from different M2M devices can be presented to the different applications and services to be further processed. M2M application portfolio covers a broad spectrum, ranging from industrial applications, to smart cities, and vehicular technologies. However, in all these applications, M2M communication trust and privacy are key requirements [2].

In order to build a secure M2M network, we proposed in a previous work a HIP-based M2M overlay network called HBMON [6]. Overlay networks are private logical networks built on the top of an existing network infrastructure (Internet for e.g.,). The overlay paradigm breaks the end-to-end principal. Instead of "*keep-it simple in the middle, intelligent at the edge*" [18], overlay networks move intelligent toward the middle. Overlay networks rely on middle-boxes (such as overlay router) connected through logical links referred as overlay links. Middle-boxes translates on-demand overlay links into Internet paths. Overlay networks are specialized networks such as peer-to-peer networks, Content-delivery networks (CDN), resilient routing networks and enhanced end-to-end security networks [3].

To define and manage our private M2M overlay network we use the Host Identity Protocol (HIP) [14, 15]. HIP introduces a new sub-layer between the transport and the IP layer. The HIP layer decouples end-host identification from its localization. End-hosts are identified with a cryptographic namespace named Host Identity Tag (HIT) while IP addresses are used as end-host locators. HIP introduces a proxy element in the network architecture, the rendezvous server which holds a secure binding between end-hosts IP addresses and their HITs. Finally, HIP is able to manage both mobility and multihoming transparently to upper layer protocols and thus provides session survivability upon end-host mobility or failures in the currently used path [16]. M2M devices within our overlay network may embed several network interfaces associated with distinct access technologies, each one associated with a distinct Internet Service Provider (ISP). Therefore, such M2M devices may be considered as multihomed M2M devices. Furthermore, M2M devices may be embedded in a vehicle to be traced or tracked and so they can be considered as mobile M2M devices, as they change their point of attachment to the network while they move.

In our previous work [6], we focused on the organization and the membership management of the M2M device within the overlay. We also proposed a novel IPv6 address assigning method in order to configure the overlay members with private IPv6 addresses. From an autonomic networking perspectives, we enabled the self-configuration and self-protection properties. The self-configuration properties allows the autonomic system to dynamically adapt itself to the deployment of new components or changes in its environment. In the HBMON, this functionality is provided by the registration functionality of the HIP protocol which allows M2M devices to autonomically register themselves with a rendezvous server and distribute overlay information between overlay members. The self-protection properties main goal is to give the system the possibility to protect itself from

intrusion and any hostile behavior. The cryptographic namespace HIT with the private addresses used within the overlay are the features used by the M2M devices in the HBMON to protect themselves from attacks.

2.2 The reachability protocol: REAP

Multihomed terminals have at least two IP addresses configured concurrently, each one associated with a distinct Internet Service Provider (ISP). These terminals are then reachable via different paths [4]. A multihomed terminal can spread its outgoing traffic among the available paths by applying a load sharing or balancing scheduling technique. However, such a scheduling technique has a negative impact on TCP. In fact, TCP segments sent on paths with lower delays may results in out-of-order TCP segments. Upon receiving an out-of-order segment, destination's TCP immediately sends a duplicated acknowledgment. Three duplicated acknowledgments results into the reduction of the TCP congestion window. Therefore, TCP erroneously concludes that duplicated acknowledgments are due to packet losses and enters in a congestion avoidance phase. Hence, multihomed terminal generally consider one path as primary and the alternate paths as backups. If a failure occurs in the primary path, multihomed terminals rehome their ongoing session to a backup path [8, 9]. The IETF has standardized a protocol for failure detection and locator pair exploration protocol for IPv6 multihoming terminals named the reachability protocol (REAP) [1]. The IETF has designed this protocol for the specific use of the Shim6 protocol. Shim6 is a host-centric multihoming management protocol [17].

REAP relies during its functioning on two timers (send timer, keepalive timer) and a state machine assuming that the communicating nodes have a prior knowledge about their locators. REAP starts the send timer whenever a node sends a packets. If this node has not received any packet until the send timer expires, it performs a full reachability exploration. Otherwise, it stops the send timer and starts the keepalive timer. If the node has not sent any packet until the keepalive timer expiry, then it sends a REAP keepalive message to its corresponding peers. If the corresponding peers receives a keepalive message, then it should stop the send timer and starts the keepalive one. The REAP specification recommends that the keepalive timer should be equal to the send timer divided by three. These two timers are mutually exclusive. In other word, the node is either expecting to receive a payload or preparing to send data. So the send timer is stopped when a payload or keepalive message is received and the keepalive timer is stopped when a payload is generated.

When REAP detects a failure, it starts a full reachability exploration in order to find a new bidirectional working address pair using Probe messages to perform the exploration and associates a state to each probe indicating the status of the communication. REAP defines three states. The first state is OPERATIONAL, it indicates that both of peers consider that their communication does not suffer from any failure. The second state is INBOUNDOK, it reflect the case where the peer considers that its communication has apparently no problem, but its correspondent peer has discovered a failure. The third state is EXPLORING, indicates that the peer has just discovered a problem and has not received any packet form its peers while it should has received.

REAP failure recovery procedure is as follows. First, REAP creates a list of all possible pair of addresses by combining the local locator list and the peer locator list and sorts this list according to some priority specified by the user. Then, it switches its state to Exploring and sends four probes successively, a probe every 0.5s. If it does not receive any probe, it retransmits a probe, but this time the retransmission is controlled by a back-off timer. A node in the OPERATIONAL state and receiving a probe having EXPLORING state means that its correspondent peer has not received its outgoing traffic. This peer then sends a probe having an INBOUNDOK state. A peer in the EXPLORING state and receiving an INBOUNDOK probe conclude that its correspondent peer has received its probe and also that the probed locator pair address is bidirectionally reachable. Thus, it sends a probe having an OPERATIONAL state and finally the communication can be resumed.

2.3 HIP mobility support

Before exchanging any data packets, HIP-enabled communicating nodes have to establish a context. The HIP context is established after a four-way handshake control messages I1,R1,I2,R2. This mechanism is based on a secure exchange of cryptographic keys to authenticate communicating hosts [15]. HIP also introduces a registrar element in its architecture: a rendezvous node (RVS) which binds nodes identification with their locations [13]. HIP nodes update their binding in the RVS node upon each change in their network connectivity. The HIP node may also interact with the the RVS element while establishing the HIP context. In fact, when a HIP node wants to establish a HIP association with a node known only by its HIT, it sends the I1 packet to the RVS indicating the Responder HIT. The RVS resolves the destination HIT into an IP address and relays the packet to the destination. After receiving an incoming I1 packet from a RVS, the Responder directly answers the Initiator and the HIP context establishment is then performed [13].

The HIP communication between two hosts is based on a security association (SA) which is established upon the HIP Base Exchange mechanism [15]. A SA is a set of security parameters agreed by two hosts in order to encrypt and authenticate transferred data. However, several SAs may be established between two hosts such as each SA has its own identifier which is called Security Parameter Index (SPI). The main role of HIP layer is to demultiplex incoming packets to host identity tag (HIT) using the SPI value in the packet and to multiplex outgoing packets to the address source and interface according the SPI value in packet. Consequently, in a HIP network, the locator is not only a IP address but also a key indexing the correspondent security association [16]. Thus, when one of two HIP nodes having an ongoing communication changes its current location to another attachment point, it acquires a new IP address and changes the SPI into SA. So, the moving HIP node has to report to the correspondent node about its new locator in order to maintain the HIP SA. In the following, we illustrate how the Host Identity Protocol supports mobility. The basic HIP mobility scenario is illustrated as follows. For setting up the HIP mobility mechanism, there are two ways to be considered; either, mobility with a single SA pair (only one IP address bound to an interface) without re-keying or mobility with a single SA pair with re-keying. In the former case, which is the simplest one, when the mobile host moves and obtains a new IP address, it notifies the correspondent host sending an UPDATE message containing the new IP address in the LOCATOR parameter and the Old SPI and New SPI values in ESP-INFO parameter. When the correspondent host receives the UPDATE packet, it checks the new address and makes it UNVERIFIED in the interim, while the old address is DEPRECATED. Then it acknowledges the mobile host by the second UPDATE message which contains an ECHO REQUEST to validate the new peer address. As well, it includes ESP INFO with Old and New SPIs set to the current outgoing SPI. Lastly, once receiving the second UPDATE message, the mobile node sends the last UPDATE message including an ECHO RESPONSE in order to definitely validate the new address. Indeed, when the correspondent host receives this ECHO RESPONSE, it automatically marks the new address as ACTIVE and removes the old address. For the second case, a new ESP session key will be regenerated. The mobile host sends the UPDATE message containing a new SPI for the incoming SA. The correspondent host upon receiving the UPDATE message, executes the rekey and replies with the a second message containing its own new SPI, then the readdressing proces ends as without re-keying case.

3 Autonomic management of the HBMON

M2M devices, as defined by the ETSI, are sensors or meters that collect data from the environment and upload them to an M2M application [10]. M2M devices and/or M2M gateways are usually equipped with several access technologies associated with distinct ISPs. They are therefore multihomed entities and consequently several overlay paths exists between M2M devices and M2M applications. M2M devices are generally connected to the M2M gateway with short range technologies (ZigBee for e.g.,); whereas, M2M gateways are usually multihomed middle-boxes, equipped with several access technologies. One of the most fundamental constraint that should be satisfied by M2M technology is communication reliability, especially for fault-tolerant oriented applications such as e-Health monitoring. To ensure communication reliability, we add to our architecture failure detection and recovery capabilities along with path monitoring functionalities.

Moreover, according to the targeted application, M2M devices can either be static or mobile nodes. Mobile nodes usually execute a layer 2 (L2) handover which may be followed by a layer 3 (L3) handover. As a result of the L3 handover, current end-host IP addresses is changed to a new topologically correct one. IP addresses have a dual role, they are considered at the same time end-host locator and session identification. Hence, without an adequate support, running transport session are broken as a consequence of a L3 handover. To ensure transport session survivability upon movement, session identification should remain stable while end-hot locator is changed. HIP addresses this issue by introducing a new stable cryptographic Host Identity Tag (HIT) as node identifier [14]. Mobility can be considered as a failure in the first hop of the path between the M2M device and the M2M application. Thus, we can easily manage the M2M device mobility through the REAP protocol, a failure detection and recovery protocol. Nonetheless, we have shown in [7] that managing the mobility with a failure detection and recovery protocol leads to a huge L3 handover latency. Therefore, we rely on the HIP protocol to handle the mobility of our M2M devices within our overlay. From an autonomic networking perspective, the self-healing property includes failures detection and recovery capability as well as mobility management.

HIP already ensures the self-configuring and the self-protecting properties of the autonomic management of our M2M overlay network. In order to provide the remaining properties (self-healing and self-optimization), we propose to add the REAP support in the M2M gateways of our HBMON.

3.1 Self-healing of the HBMON

3.1.1 Failure detection and recovery

In our M2M overlay network, several overlay paths might exist between the gateway and the corresponding M2M applications. This path diversity is highly recommended for specific fault-tolerant system such as security-oriented applications. In order to design a resilient M2M overlay network, we use the REAP protocol to: (i) monitor the existing paths, (ii) detect failures and recover to a new working path. We enable REAP at the gateway level for several reasons. First of all, in our design [6], the overlay architecture is maintained at the gateway, which is viewed form a HIP perspective as Rendezvous node. Second, the overlay link diversity starts at the gateway level as the sensors are usually single-homed entities. Thus, we couple HIP with REAP at the gateway level. We define new parameters in the HIP messages to support the REAP protocol namely "PROBE" and "KEEP ALIVE". They are of type "NOTIFY". The former is exchanged between peers when a failure is detected and the latter is used to monitor unidirectional communications. We add to the HIP two REAP timers, namely send and keepalive timers. If a peer's send timer expires without receiving any incoming packets, the peer assumes that a failure has affected its currently used overlay path and starts exploring the remaining available overlay paths. In unidirectional communications, the peer has to periodically inform its corresponding node that the currently used overlay link is working. When the keepalive timer expires, the peers sends a keepalive message. At the beginning of a communication, the M2M gateway exchanges with the M2M application data packets and eventually keepalive messages. REAP only monitors the currently used overlay link. If REAP detects a failure through the expiry of the send timer, REAP starts the overlay paths explorations. During this exploration, REAP sends probe messages on each available overlay link having the status exploring. The corresponding peer receiving the probe message replies with a probe message indicating the status of the probed overlay link. Upon receiving a probe message with the status inbound OK, REAP replies with a probe with an operational state and switch the ongoing communication to this newly operational overlay link.

3.1.2 Mobility management

To efficiently manage M2M devices mobility, we propose to enhance the HIP rendez-vous server functionalities, embed at the M2M gateway level, to ensure session survivability between HBMON members.

First of all, an M2M device, member of the HBMON, performs a layer 2 (L2) handover. Once the layer 2 connectivity is established, the M2M device receives an IPv6 router advertisement from the new access router and configures a new global IPv6 address. At this stage, both M2M devices corresponding peers and the HBMON rendez-vous servers are not aware about the M2M device new location. To correctly handle the HBMON mobile nodes (HMN) mobility, we introduce in the HIP protocol the following signaling messages. (i) RVS_Discovery: This signaling message allows to discover the nearest HBMON Rendez-vous server (NHRVS). This message is sent in anycast. (ii) HMN_Loc_Up: Contains two main fields; (1) NEW_IP: to report the new HMN's IP address to the correspondent node, (2) CONTEXT_Req: to request the HBMON Context. (iii) Context Update: Once a HMN obtains a new IP address upon moving, it should inform all the rendez-vous server (HRVS) members of the HBMON multicast group about this new IP address. This message is sent in multicast via the old HRVS.

Fig. 1 illustrates this mechanism through an example where a HBMON overlay is established between a HBMON mobile node (HMN) and a HBMON correspondent node (HCN), HMN and HCN have an ongoing communication and the HMN moves to another autonomous systems (AS). This strategy is an enhanced version of the HIP mobility management presented in [16]. Fig. 2 presents the sequence diagram of the exchanged signaling messages for this strategy.

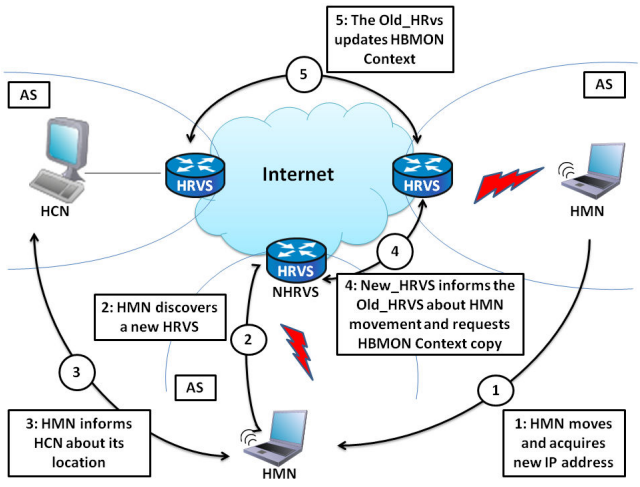

Figure 1: Mobility management scenario

When the HMN moves and acquires a new topologically correct IP address (step 1 Fig. 1), it sends an RVS_Discovery message containing the old HBMON Rendez-vous server's (HRVS's) IP address and its HIT (step 2 Fig. 1). The RVS_Discovery message is sent to a specific anycast address in order to discover the nearest HBMON rendez-vous server (NHRVS). After that, the HMN reports its new IP address to its HCNs us-

ing the HIP mobility mechanism; as explained in section 2.3 (step 4 Fig. 1). The new RVS notifies the old HRVS about the new HMN location and triggers the HBMON context update by sending the Context_Req message (step 3 Fig. 1). Once the old HRVS receives a Context_Req message, it updates the mapping between the HMN's HIT and its new IP address. Afterwards, it updates the HBMON context forwarding to all HBMON RVS the Context Update message (step 5 Fig. 1). This message is sent on specific multicast address including all HBMON rendez-vous servers.

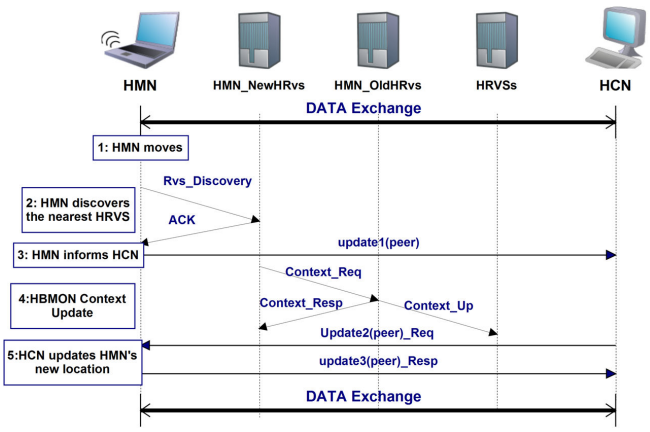

Figure 2: Mobility management sequence diagram

Consequently, we build a self-healing HIP-based M2M overlay network by adding both the failure detection and recovery capability to the M2M gateway, and the M2M device mobility management.

3.2 Self-optimization of the HBMON

The available overlay paths have different network characteristics (RTT, jitter, errors,...) as they cross different Internet Service Providers. An overlay link can experience for a certain period of time a degradation of its network characteristics. Such overlay link can be used by an M2M communication requiring a certain level of quality of service. We propose to use the REAP exploring mechanism to offer to the M2M running communication always the best available link. Instead of triggering the REAP exploring process at the expiry of the send Timer, we continuously monitor the available paths and infer their respective RTT. If the currently used overlay link experience a degradation of its RTT, REAP proposes to HIP a new destination/source address pair of an overlay link having lower RTT. If we frequently perform the inferring of the RTT and overlay paths switching, we can cause overlay paths oscillation, known as route flapping. To avoid route flapping, we add a new timer, namely probe timer which defines the time between two consecutive path exploration. Thus, our HIP-based M2M overlay network is self-optimized as it always selects the best available overlay path in term of RTT.

4 Evaluation

To evaluate our proposal, we use the OMNeT++ simulator coupled with the HIPSim++ framework. We implement the autonomic management of the HBMON in the HIPSim++ framework.

4.1 Failure detection and recovery time

The targeted testbed consists of an M2M device connected to a mutlihomed M2M gateway. The M2M gateway has four available overlay paths having the following RTTs: 50ms, 100ms, 150ms and 200ms. The correspondent node is an M2M application. We set all the wireless accesses to 802.11b at 11Mbit/s. Between the M2M application and the M2M device we use two types of traffic: the first one is an UDP flow having the following characteristics: 20 Bytes the packet length and 40 ms the inter-packet interval, the second traffic is TCP flows, namely an FTP application with hight data traffic. We focus on two metrics: the application recovery time and the instant throughput. The application recovery time (ART) is defined as latency between the last packet received/sent before the outage and the first packet received/sent after the outage.

We evaluate in this section the failure detection and recovery capabilities of our solution. A failure occurs after 20s from the beginning of the communication and lasts twice as the send timer. We measure the ART of UDP traffic and the TCP/FTP traffic. Results are presented by Fig. 3, the x-axis is the send timer value while the y-axis is the measured ART. La Oliva et al [12]. have already measured the ART of both TCP and UDP traffic. By this figures, we aim to validate our REAP implementation in the HIPSim++ framework. We obtain the same results as the one obtained by La Oliva et al. in [12]. Results show that for an UDP application, the ART time increases linearly while we increase the send timer value; whereas, for TCP application the ART experiences several plateaus. After failure recovery, UDP application immediately sends data packets to the newly selected path. Even if a new overlay path is selected, TCP does not send immediately its data segments. TCP has to wait until the TCP Retransmission Timeout (RTO) timer expiry. TCP does not distinguish between a failure recovery process and the congestion in the currently used path [5]. It adjusts the RTO timer as if it has experience of a congestion phase which explains the plateaus in Fig. 3

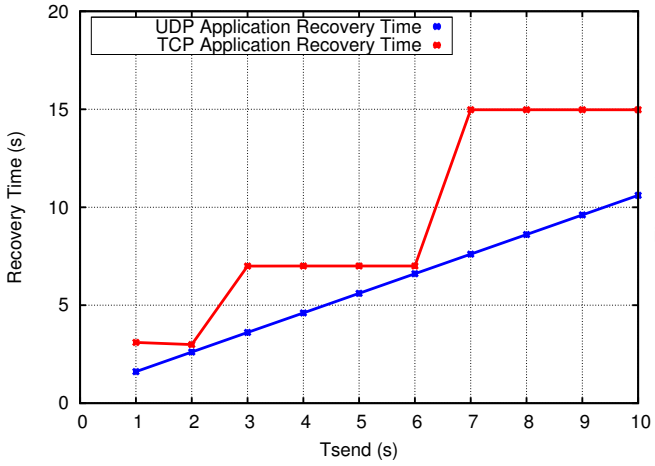

Figure 3: Application recovery time after an outage

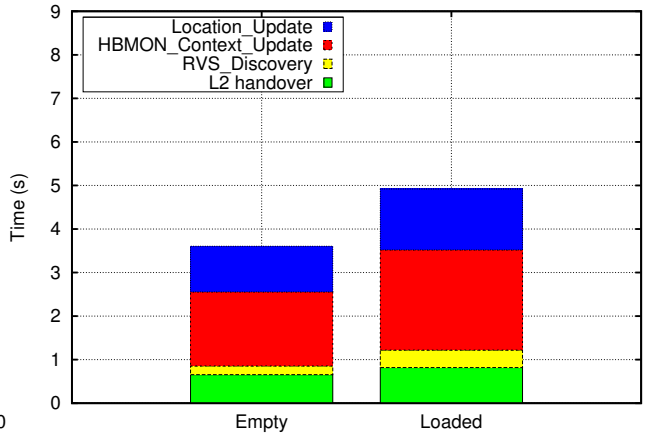

Figure 4: Application recovery time after after M2M device mobility

4.2 Mobility management

We evaluate in this section the mobility management capabilities of our solution. In this scenario, we configure two M2M devices: HMN1 and HMN2, registered respectively with HRVS1 and HRVS2. HMN1 has a 802.11b interface associated with access point AP1 and HMN2 has a 802.11b interface registered with access point AP2. HMN1 is a static node; whereas, HMN2 is a moving according to the random waypoint model. HMN1 and HMN2 exchange a 1 Mbit/s UDP traffic. We load the visited network with three nodes, each of them generating a UDP traffic at 1 Mbit/s In our simulation, we measured the ART which is the latency elapsed between the last packet sent with the old IP address and the first packet sent with the new IP address. The histogram presented in Fig. 4 illustrates the measured ART for an empty and loaded visited network.

The ART latency is decomposed into 4 phases: (i) L2 handover, (ii) RVS_Discovery, (iii) HBMON_Context_Update and (iv) Location_Update. In an empty visited network, the L2 handover latency is 0.65s, the RVS_Discovery latency is 0.2s. The HBMON_Context_Update latency is 1.7s and the Location_Update latency is 1.05s. In a loaded visited network, the L2 handover latency is 0.819s, the RVS_Discovery latency is 0.4s, the HBMON_Context_Update latency is 2.3s and the Location_Update latency is 1.4s. We observe that with our solution, running session effectively resume after the mobility. The mobility singling lasts more than 3.6s for the case of an empty visited network (the best measured case) which is inadequate for real time applications. Nonetheless, M2M applications are usually low data-rate application, and providing session survivability - even after 3.6s of interruption- is preferable than completely losing the currently ongoing session.

4.3 Path exploration

We evaluate in this section the self-optimization capability of our solution. We modify REAP to actively monitor the available paths in order to offer the ongoing M2M communication the best available overlay path in term of RTT.

We focus on the following scenario: the currently used overlay path has an RTT of 50ms and a transient failure affects this path after 20s of the beginning of the M2M communication, the failure lasts the double of the probe timer. Fig. 5 shows the obtained results for a TCP session and a probe timer set to 3s. The x-axis is the time in second and the y-axis is the instant throughput. The obtained results show that during the first 20s, the throughput reaches its maximum because the used path has the minimum RTT (50ms). After the failure recovery, REAP detects a new working overlay path having the second best RTT (100ms). As soon as the best overlay path (50ms) recovers forms its failure, M2M communication switches to this new path and the throughput reaches again its maximum value. Fig. 6 shows the obtained results for a running UDP session and a probe timer set to 3s. The obtained results show the same behavior as for the TCP case in Fig. 5. After the outage, the UDP session is rehomed to a new working ovelray path (100ms). As soon as the new overlay path (50ms) becom ready, the UDP session is rehomed to this newly available path, and the throughput reaches again its maximum value.

In a second scenario, we explore the self-optimization capability of our solution by modifying the load of the currently used overlay path. The M2M communication starts in the overlay path having the lowest RTT. A congestion appears in this path, so the TCP ongoing con-

nection experiences packet losses, TCP reduces its congestion window which impact the instant throughput of the M2M communication. Our solution detects the quality degradation of the path and switches the communication to the second best path in term of RTT. Results presented by Fig. 7 and Fig. 8 shows this dynamic selection of the most stable path. During the first 20s, the M2M communication flows via the path having the lowest RTT (50ms). We inject in this path aggressive UDP traffic, creating therefore a congestion path. Our solution detects the degradation of the RTT of this path and its fluctuations. It switches the ongoing communication to the second path. We repeat the same scenario on this second path. Our solution switches one more time the communication to a third path and finally to the last one until it finds a stable path in term of RTT and packet loss.

From Fig. 5, Fig. 6, Fig. 7 and Fig. 8 we clearly see that we build a self-optimized solution. It is able to detect failure in the currently used overlay path, select a new working path and monitor the remaining paths.

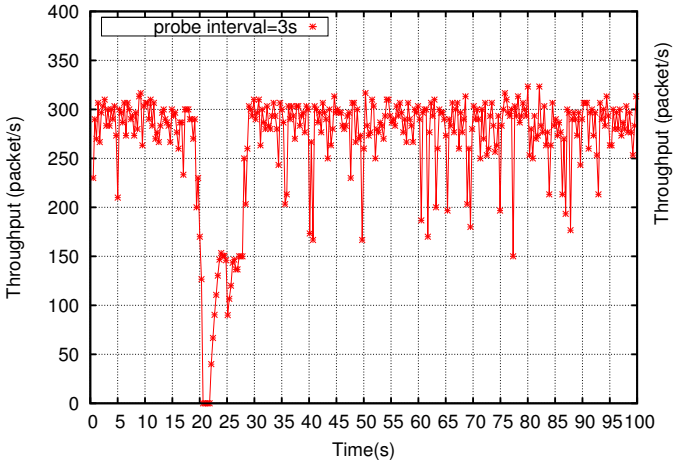

Figure 5: FTP recovery time

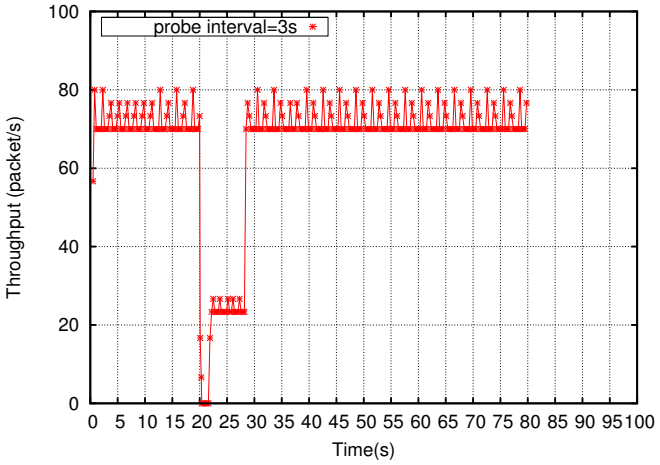

Figure 6: UDP recovery time

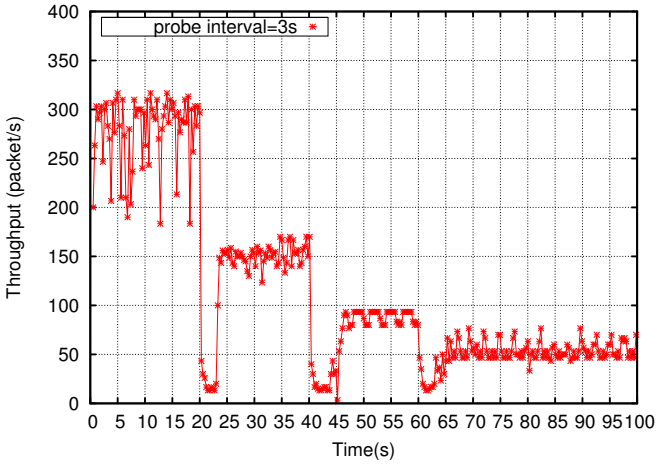

Figure 7: FTP dynamic path selection

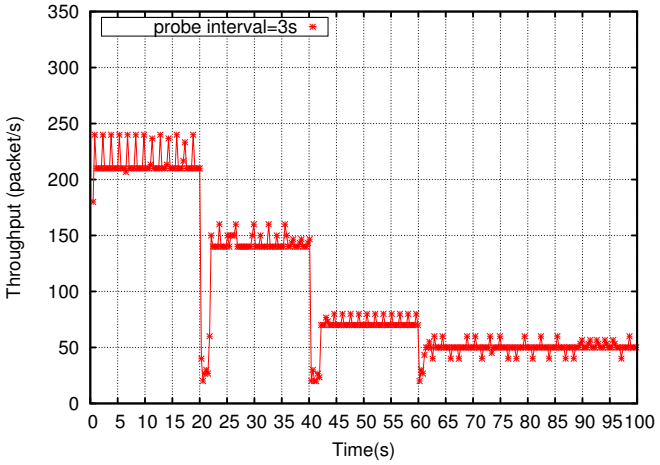

Figure 8: UDP dynamic path selection

5 Conclusion

M2M communication is a new paradigm under standardization at both the ETSI and the 3GPP. This novel technology breaks the end-to-end principle as it introduces a novel element in the network architecture namely the M2M gateway. The M2M gateway aggregates the data collected from the M2M devices and sends them to a correspondent M2M application. In a previous work [6], we have designed a HIP-based M2M overlay network over the existing Internet. This overlay ensures a private communication between M2M devices and their corresponding M2M applications. In this work, we added the autonomic management of our M2M overlay net-

work. We focused mainly on the self-healing and the self-optimized autonomic properties. We enhanced the M2M gateway with the failure detection and recovery mechanism, M2M device mobility management and with autonomic path selection capabilities. We implemented and evaluated our solution on the OMNeT++ network simulator. Results shows that the gateway is able to switch from one overlay path to another either due to failure or due to the path characteristic degradation. Results also show that M2M devices running sessions survive to the mobility episode. We are currently implementing this solution on the phidget[1] testbed.

Acknowledgment

This work has been partially funded by the project ITEA2-A2NETS.

References

[1] ARKKO, J., AND VAN BEIJNUM, I. Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming. RFC 5534 (Proposed Standard), June 2009.

[2] BAILEY, D. A. Moving 2 Mishap: M2M's Impact on Privacy and Safety. *IEEE Security and Privacy 10*, 1 (2012), 84–87.

[3] CLARK, D., LEHR, B., BAUER, S., FARATIN, P., SAMI, R., AND WROCLAWSKI, J. Overlay Networks and the Future of the Internet. *COMMUNICATIONS & STRATEGIES 63*, 3 (2006).

[4] DHRAIEF, A., AND BELGHITH, A. Multihoming support in the internet: A state of the art. In *International Conference on Models of Information and Communication Systems (MICS 2010)* (2010).

[5] DHRAIEF, A., AND BELGHITH, A. An experimental investigation of the impact of mobile ipv6 handover on transport protocols. *The Smart Computing Revue, KAIS 2*, 1 (2012).

[6] DHRAIEF, A., GHORBALI, M. A., BOUALI, T., BELGHITH, A., AND DRIRA, K. HBMON: A HIP-Based M2M Overlay Network. In *The 2012 Third International Conference on the Network of the Future (NoF 2012)* (2012).

[7] DHRAIEF, A., MABROUKI, I., AND BELGHITH, A. A service-oriented framework for mobility and multihoming support. In *Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean* (march 2012), pp. 489 –493.

[1] http://www.phidgets.com/

[8] DHRAIEF, A., AND MONTAVONT, N. Rehoming decision algorithm: Design and empirical evaluation. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 02* (Washington, DC, USA, 2009), CSE '09, IEEE Computer Society, pp. 464–469.

[9] DHRAIEF, A., ROPITAULT, T., AND MONTAVONT, N. Mobility and multihoming management and strategies. In *14th Eunice Open European Summer School 2008* (2008), IFIP, Ed., RSM Dept (Institut TELECOM ;TELECOM Bretagne).

[10] ETSI TECHNICAL COMMITTEE MACHINE-TO-MACHINE COMMUNICATIONS (M2M). Machine-to-Machine communications (M2M); Functional architecture. TS 102 690, Oct. 2011.

[11] JUNIPER NETWORKS. Machine-to-Machine (M2M) The Rise of the Machines. white paper, 2011.

[12] LA OLIVA, A., BAGNULO, M., GARCÍA-MARTÍNEZ, A., AND SOTO, I. Performance analysis of the reachability protocol for ipv6 multihoming. In *Proceedings of the 7th international conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking* (Berlin, Heidelberg, 2007), NEW2AN '07, Springer-Verlag, pp. 443–454.

[13] LAGANIER, J., AND EGGERT, L. Host Identity Protocol (HIP) Rendezvous Extension. RFC 5204 (Experimental), Apr. 2008.

[14] MOSKOWITZ, R., AND NIKANDER, P. Host Identity Protocol (HIP) Architecture. RFC 4423 (Informational), May 2006.

[15] MOSKOWITZ, R., NIKANDER, P., JOKELA, P., AND HENDERSON, T. Host Identity Protocol. RFC 5201 (Experimental), Apr. 2008. Updated by RFC 6253.

[16] NIKANDER, P., HENDERSON, T., VOGT, C., AND ARKKO, J. End-Host Mobility and Multihoming with the Host Identity Protocol. RFC 5206 (Experimental), Apr. 2008.

[17] NORDMARK, E., AND BAGNULO, M. Shim6: Level 3 Multihoming Shim Protocol for IPv6. RFC 5533 (Proposed Standard), June 2009.

[18] SALTZER, J. H., REED, D. P., AND CLARK, D. D. End-to-end arguments in system design. *ACM Trans. Comput. Syst. 2*, 4 (Nov. 1984), 277–288.

[19] TECHNICAL SPECIFICATION GROUP SERVICES AND SYSTEM ASPECTS. Study on facilitating machine to machine communication in 3GPP systems. 3GPP TR, Mar. 2007.

[20] UUSITALO, M. A. Global vision for the future wireless world from the wwrf. *Vehicular Technology Magazine, IEEE 1*, 2 (2006), 4 –8.