Autonomic management of the HIP-based M2M overlay network

Amine Dhraief\textsuperscript{a}, Abdelfettah Belghith\textsuperscript{a}, Khalil Drira\textsuperscript{b}, Tarek Bouali\textsuperscript{a,b}, Mohmaed Amine Ghorbali\textsuperscript{a,b}

\textsuperscript{a}HANA Research Group
University of Manouba, Tunisia
\textsuperscript{b}LAAS-CNRS, France
Univ. de Toulouse, France

Abstract

In a previous work, we build an M2M overlay network over the Internet in order to ensure private communication between M2M devices, M2M gateway and M2M applications. We propose in this paper to add the autonomic management of this overlay network. We add the self-healing and self-optimization capabilities to the M2M gateway. The M2M gateway is able to detect failures in the overlay path and recover from them. It is also able to dynamically monitor the overlay path and select the best available path in term of RTT. These two features are provided by the reachability protocol (REAP) at the gateway level. We implement our solution on the OMNeT++ network simulator. Results highlight the novel gateway capabilities: it recovers from failures and always select the best available path.

Keywords: Machine-to-Machine, Overlay networks, Autonomic management, REAP, OMNeT++

1. Introduction

According to the Wireless World Research Forum (WWRF), by 2017 we will have 7 trillion wireless devices serving 7 billion people [1]. In another market forecast study elaborated by Juniper Networks in 2011 [2], it was estimated that by 2015, the number of connections between embedded equipments such as sensor or telemeters will reach over 500 millions.

Internet is based on the well-known paradigm: "keep-it simple in the middle, smart at the edge" [3], which survived for the last four decades. Nonetheless, Machine-to-Machine (M2M) communication brings new demands. It 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 from sensors and sends them to the M2M application. It generally communicates with M2M devices via short range communication technologies. The M2M gateway breaks the Internet 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 most fundamental requirement of M2M communications is the privacy of the collected information. This requirement drives us to build an M2M overlay network over the Internet based on the Host Identity Protocol (HIP)[4, 5], named HBMON (HIP-based M2M Overlay Network)[6]. In our previous work, we have addressed the formation and the maintenance of the overlay. In this work, we propose to add the autonomic management of the overlay. We mainly focus on the self-healing and self-optimization autonomic properties. Thus, in our overlay, M2M gateways are able to autonomicallly detect failure of the overlay links and recover from them. M2M gateways are also able to monitor the available overlay paths and dynamically select the best path in term of Round Trip Time (RTT). To achieve these goals, we use the Reachability Protocol (REAP), standardized by the IETF [7]. We enable REAP at the gateway level along with HIP. 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 II gives an overview of our previous work HBMON[6], introduces the autonomic networking properties and details the REAP protocol. Section III highlights our contribution; namely the autonomic management of the HBMON. Section IV presents the simulation results. Finally, Section V 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 focus on the the autonomic networking properties. Finally, we detail the reachability protocol: a failure detection and recovery protocol.

2.1. The HIP-based M2M Overlay Network

Machine-to-machine communication is a novel communication technology under standardization at both the European Telecommunications Standardization Institute (ETSI) and 3rd Generation Partnership Project (3GPP). An M2M communication involves an M2M device communicating with an M2M application via M2M gateways, with no human intervention [8]. The first "Machine" in a Machine-to-Machine communication is a device embedding both 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. The M2M gateway aggregate the 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, all these applications agrees on privacy as an important requirement to be fulfilled. In order to build a secure M2M network we proposed in [6] a HIP-based M2M overlay network called HBMON. Overlay networks are private logical network build 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", overlay networks move intelligent toward the middle. 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 [9]. 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.
To define and manage our private M2M overlay network we use the Host Identity protocol (HIP) [4, 5]. HIP introduces a new layer between the transport and 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 the upper layer protocols and thus providing session survivability upon end-host mobility or failures in the currently used path[10]. M2M devices within our overlay network may embed several access technologies, each one associated with a distinct Internet Service Provider (ISP), so they can 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 viewed as mobile M2M devices as they change their point of attachment to the network while they move.

Our previous work [6] focused on the definition and the addressing techniques based on the HIP protocol used within a private M2M overlay network.

2.2. Autonomic Networking

An autonomic network is formed by a federation of heterogeneous systems self-formed without any managing authorities nor specific infrastructures[11, 12, 13, 14]. An autonomic network is able to self-manage itself and form its own policies and decision according to information gathered from its environment. In order to perform targeted functionalities, individual components as well as the whole autonomic network should provide properties namely self-properties [15, 16]. Each of them is described below:

- **Self-configuring:** Allows the autonomic system to dynamically adapt itself to the deployment of new components or changes in its environment. In our previous work [6], 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.

- **Self-healing:** Allows a system to evaluate its actual state and perform corrective actions without disrupting system operation. Failure detection and recovery done by a device in the network is the most important feature of self-healing as the path management is autonomically performed.

- **Self-optimizing:** This feature refers to the ability of the environment to efficiently maximize resource allocation and utilization to meet end-users needs with a minimal human intervention. Self-optimizing does not only target the complexity of managing system performance but also gives possibility to components to form knowledge and proactively control available paths to ensure the optimal path selection.

- **Self-protecting:** The main goal of this property is to give the system the possibility to protect itself from intrusion, hostile behavior, viruses or denial of service attacks. HITs used by the Host Identity protocol with the private addresses used within the overlay are the features used with M2M devices in the HBMON to protect themselves from attacks.

In our previous work [6], we focused on the self-configuration and self-protection properties when we build the overlay network benefiting from the HIP functionalities. In this actual work we are enhancing our solution HBMON with the self-healing and self-optimization properties using the reachability protocol REAP [7].

2.3. 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. As an example, let us assume a multihomed source S having two IP addresses \((IP_{src1}, IP_{src2})\), and a multihomed destination D having also two IP addresses \((IP_{dest1}, IP_{dest2})\). In this case, we count four available paths between S and D as listed in the following: (i) path 1 between \(IP_{src1}\) and \(IP_{dest1}\), (ii) path 2 between \(IP_{src1}\) and \(IP_{dest2}\), (iii) path 3 between \(IP_{src2}\) and \(IP_{dest1}\), (iv) path 4 between \(IP_{src2}\) and \(IP_{dest2}\).
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 result in out-of-order TCP segments. Upon receiving an out-of-order segment, destination’s TCP immediately sends a duplicated acknowledgment. Three duplicated acknowledgments result in the reduction of the TCP congestion window. 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 terminal re-home their ongoing session to a backup path[17]. The IETF has standardized a protocol for path monitoring, failure detection and recovery named the reachability protocol (REAP) [7]. In an autonomic architecture, the REAP protocol is in charge of the self-healing aspect of the network.

REAP relies during its functioning on two timers and a state machine assuming that the communicating nodes have a prior knowledge about their addresses before starting the communication.

The protocol defines a send timer and a keepalive timer. The send timer is started whenever an outgoing payload is generated to reflect that the peer receiving the traffic should return something in response. This means that if no response from the peer is received during a timeout equal to the send timer the host detects a failure in the currently used path. In case of unidirectional communication, when the peer does not have any payload to send, the keepalive timer is fired and a keepalive message is generated when the timer expires. The REAP specification recommends that the keepalive timer should 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.

During a communication each node maintains a state machine having three states: Operational, Exploring and InboundOK [7]. If the communication is not experiencing any problem the state is Operational which means that each node is receiving either data packets or keepalive signals from the other. The node switches to the Exploring state if a failure occurred and the send timer expires. Reaching this state, the node begins to send probes to the peer host exploring possible paths and each probe include its state. The third state, InboundOk is reached if a host receives packets from its peer, but there is an indication that the peer does not receive packets. A host may join the InboundOk state from Operational when it receives an exploring probe or from Exploring if it receives anything from its peer. Fig.1 illustrates a case of REAP failure detection and path exploring.

![Diagram of REAP failure detection and recovery](image)

Fig. 1: Example of failure detection and recovery

In this scenario, the host A has data to send to host B in a unidirectional communication. Host A has two addresses (A1,A2) and B has two addresses (B1,B2). At the beginning A and B are exchanging packets
using the pairs \((A1,B1)\) till the first packet from A was lost. B continues to send keepalive messages to A until A send timer expires resulting into a switching to the Exploring state by A and the sending of probes exploring possible paths. The first probe was lost which means that there is a failure in that path. The second path is a working path, B receives the first exploring probe, changes its state to InboundOk and sends an InboundOk probe which reaches A. Receiving the InboundOk probe from B, A returns to the Operational state, sets the new source and destination addresses and restarts the payload exchange using the new path.

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 [8]. 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. From an autonomic networking perspective, failures detection and recovery capability is provided by the self-healing property. Similarly, the path monitoring functionality is provided by the self-optimization property. 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 HBMON

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 monitor the existing paths, 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 from 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. It sends each keepalive timer 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. Consequently, we build a self-healing HIP-based M2M overlay network by adding the failure detection and recovery capability to the M2M gateway.
3.2. Self-optimization 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. The targeted test bed consists of an M2M device connected to a multihomed 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. 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 high 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.

4.1. Failure detection and recovery time

We evaluate in this section the self-healing 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.2 and Fig.3. The x-axis is the send timer value while the y-axis is the measured ART. La Oliva et al. have already measured the ART of both TCP and UDP traffic in [18]. By these two 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 [18]. 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 RTO timer expiry. TCP does not distinguish between a failure recovery process and the congestion in the currently used path. It adjusts the RTO timer as if it has experienced a congestion phase which explains the plateaux in Fig.3.

4.2. 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.

Firstly, we focus on the following scenario: the currently used overlay path has an RTT of 20ms 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.4 shows the obtained results for different values of probing intervals. 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 (20ms). After the failure recovery, REAP detects a new working overlay path having the second best RTT (50ms).
As soon as the best overlay path (20ms) recovers forms its failure, M2M communication switches to this new 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 connection 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.5 shows this dynamic selection of the most stable path. During the first 20ms, the M2M communication flows via the path having the lowest RTT (20ms). 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 and 4, 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. Furthermore, it is able to detect non stable path and select a more stable one for an ongoing M2M communication.

5. Conclusion

M2M communication is a new paradigm under standardization at both the ETSI and 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 network. 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 and with autonomic path selection capabilities. We used for this purpose the reachability protocol. 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. We are currently implementing this solution on the phidget\(^1\) testbed.

References


\(^1\)http://www.phidgets.com/