Network management architecture and models in Internet of Things

Miroslav Slavchev[1] and Aleksandar Tsenov[2]

Abstract – **The main domain of interest in this paper is the implementation of Internet of Things (IoT) solutions in order to meet the ever-increasing demands on modern network management. As a result, the work presents a conceptual framework in order to demonstrate how to incorporate IoT technologies and to elaborate on the requirements and possible advantages. This article also provides examples with concrete IoT solutions and their possible application in the framework as a final proof-of-concept.**

Keywords – **Internet of Things, network management, architectural framework, out-of-band model**

I. Introduction

Internet of Things (IoT) has presented some emerging technologies [1], which have the potential to provide valuable alternatives and/or additions to common elements of current network management solutions. However, we do not discuss this topic. Thus, the aim of this work is to develop a framework for the seamless integration of IoT in the problem domain.

In general terms, the approach to network management is based on a set of elements, which were used to solve tasks since the first stages of the internet evolution. Examples of these are SNMP, the manager-agent paradigm and in-band management. [2]

Despite modifications (such as SNMP v2 and v3 [3]) in order to meet new demands, these elements impose limitations that may become problematic in the context of modern requirements. A simple example is the shift from reactive to pro-active networking [4], preconditions for which are having accurate enough information about the momentary regional state of the network and low latency by decision-making. The paradigm will result in a delay for data propagation between the agents and a remote manager and it will create severe competition between generated management data (due to finer polling) and primary traffic as well.

However, recent developments have also lead to new technologies and principles, with IoT being the last major step in the evolution of internet. Its applications in various fields, such as the management of households, traffic systems and city resource grids, are already in the focus of research. [5] But the possible benefits of this new design, oriented towards frequent collection, aggregation and distribution of small data volumes with reduced overhead, is not been discussed in terms of applicability for the tasks of network management.

II. Main concept

A. Key requirements

Seamless integration – in order to be commercially viable, the framework must not be mutually exclusive with existing solutions. It should behave as an optional extension, which is applicable to already existing systems. Integration should require minimal expense and modification (preferably only by configuration).

Modularity – For further reduce of transition costs, a modular approach should be applicable. It must be possible to delegate only certain tasks from an existing solution, without having to incorporate other (non-related) elements of the framework. E.g., one should have the option to delegate accounting data collection to an out-of-band module of the framework without having to migrate the fault-management or other tasks. This can also be beneficial in the sense to incrementally incorporation of the framework in an existing solution, further simplifying the migration procedure into manageable steps.

Out-of-band support – in-band management results in competition for resources between business and management tasks. This coupling introduces complexity in the development of a network, as addressing both problem domains simultaneously and compromising the design should between types of requirements. Using the primary network as a media to deliver management messages introduces unreliability. When managed resources cannot accept new messages (e.g. due to a deliberate attack against the system), their embedded agents cannot communicate with the manager. As a result, none of the orders required for restoring the functional state of the network can be delivered.

Interference – In the scenario, that physically separate primary and management networks should coexist together, the possible loss of data (and reduced utilization) due to interference between the two media should be taken into account.

Layered architecture – the design of the presented framework must consider the dynamic nature of emerging technologies and concepts. This translates directly into a requirement for a layered architecture, which abstracts the concrete specifics of a given implementation, lowering coupling between different elements and the dependencies of the global solution.

Decentralized logic – in order to reduce latency and overhead, responsibilities should be distributed over a chain of command. In comparison to already existing manager-agent

[1] Miroslav Slavchev is a Master-Student at the Faculty of German Engineering Education and Industrial Management at Technical University of Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria,
E-mail: miroslav.slavchev@fdiba.tu-sofia.bg.

[2] Alexandar Tsenov is with the Faculty of Telecommunications at Technical University of Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria, E-mail: akz@tu-sofia.bg

solutions, the more complex tasks will remain based in the manager, but problems which require only local situation awareness should be delegated closer to the agents in order to optimize the process. Enabling communication between agents without dependency on the manager is a prerequisite for logic decentralization.

Scopes – local processing does not exclude the possibility to propagate collected data further up the chain for more complex analysis. E.g, when a fault is detected, agents can collect and share data in order to undertake a simple strategy against the problem. These measures may include a set of steps in direct response to the threat. But they may also asynchronously send the same data to the manager in order to analyze the cause of the problem and possibly apply further actions. Introduction of scopes will allow for finer control and aggregation by such propagation.

B. Concept

Fig. 1 depicts the possibility of presenting auxiliary agents as an intermediating layer between agents and managers. First is the common solution – in-band management of network elements (NE) via embedded Agents (AA) and a remote manager (M).

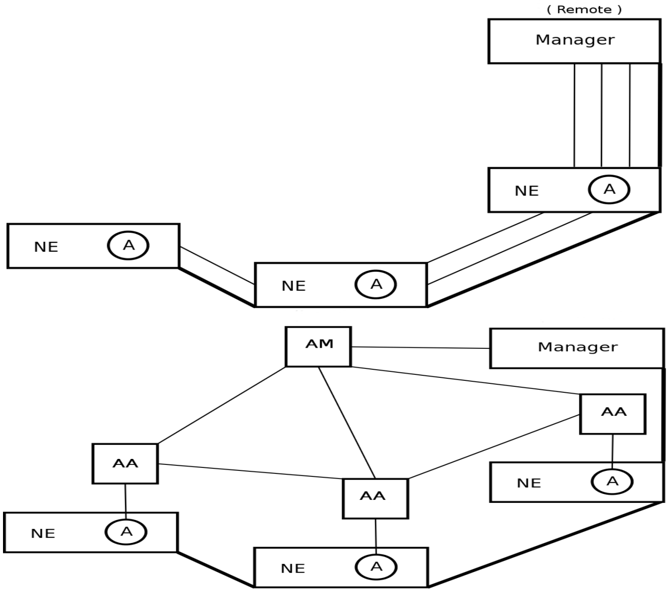

Fig. 1. Model extension via auxiliary elements

The conceptual framework model extends this system via auxiliary agents (AA), directly connected with the original agents. Simpler management and inter-AA communication tasks are incorporated in an embedded logic module (Fig. 2). An auxiliary manager (AM) can be used in order to accommodate more complex policies and coordinate AAs.

Since this is an expansion of an already existing solution, it is known in advance what functionalities the manager exposes to agents (and vice-versa) in order to cover a set of tasks. Therefore, these functionalities can be abstracted into an interface (quasi-manager) and exposed by the auxiliary agent, allowing to extend the chain of command from (NE-A)-M to (NE-A)-AA-M. From an agent's perspective, messages will be sent to and received from a AA which behaves in the same manner as the manager. On the grounds of this, no changes in the inner logic of agents will be necessary. The same applies to remote managers as well. However, this does not eliminate or prohibit in any way the coexistence between a direct A-M and an A-AA-M chain. Fig. 2 illustrates another option for modification of the chain, where AAs can be used as alternatives to Agents. In this case, a subset of the agent's functionalities are embedded in the AA so that it can independently monitor and interact with the managed resources.

This presents the model from a view based on the management aspect. However, concerns should also be distributed in a manner, which is natural to IoT in order to utilize some important advantages. Agents can embody the sensor and actuator roles, as they already incorporate equivalent functionalities in the traditional manager-agent paradigm. AAs will provide the most important asset – device-to-device (“agent-to-agent”) connectivity, and allow for out-of-band communication between local elements.

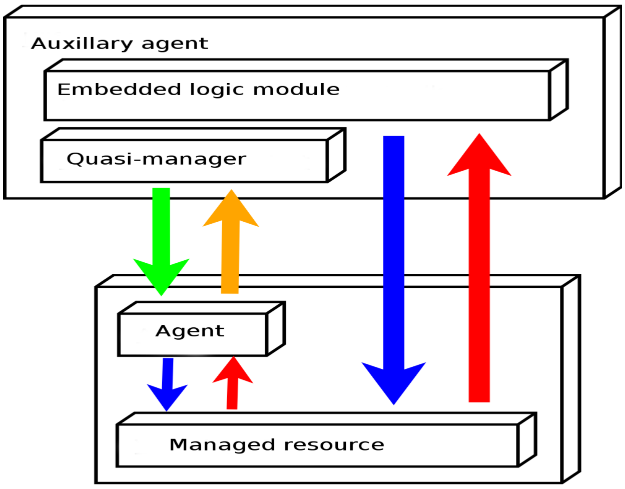

Fig. 2. Network Element– Agent – Auxiliary Agent

Auxiliary managers, if necessary, can be used also as coordinators for larger secondary networks.

The next point of interest is communication within the boundaries of the auxiliary network and between it and the central manager.

Some IoT protocols are more suitable than others for specific tasks. Nevertheless, all of them are designed to support rapid exchange of small data amounts, which is appropriate for most management problems (with the exception of large configuration file distribution). Therefore they are applicable for the purposes of “agent-to-agent” communication. In case some of the data should be further propagated to the remote manager over a standard TCP/IP network, protocol mapping may be incorporated in the AA logical module. If AMs are available, they are the most appropriate candidate for a gateway, adding collection, filtering and aggregation capabilities before data is converted from one protocol to another. This will also reduce the impact of management traffic on network performance in other sectors which have only in-band support.

III. Practical approach

A. Management network architecture

The management network itself needs some form of administration. Since a thorough discussion of this problem is not possible within the boundaries of this article, only a generalization will be presented. Based on the aforementioned networking view of the conceptual model, AAs are to be included in one or more groups, subordinated to local auxiliary managers. The responsibilities of the AM also include providing access to AAs to external elements (e.g the remote manager or other AMs).

Devices should also be uniquely identifiable. Regarding their task, AAs need an identity only in the scope of the management group they participate in. Thus, when AAs from two different groups need to exchange information, their full identities are required. For the far more likely scenario of direct communication between AAs within the same group, the overhead of the fully-qualified global identity is neither practical, nor obligatory. Instead, a shorter form of identification can be used. Upon evaluation of the optimal size scalability should be taken into consideration: smaller sizes will directly reduce the overhead by communication between devices from the same group, but it will also reduce the maximum number of elements one subgroup can hold. This will lead to a higher number of groups in order to cover a larger management network. Which also leads to increased probability of communication between AAs from separate groups, requiring the full representation form of both elements. In conclusion, optimal size of the shorter form for smaller domains is 1 Bytes, but mid- and large-scale networks will require 2 Bytes.

A. PHY layer and MAC - 802.15.4

The choice of a physical layer must be carefully considered. The aforementioned motivation for an out-of-band model may be severely compromised by radio frequency interference. RFI may cause significant packet loss or temporally deferred transmissions in both networks, which reintroduces the problem of competition for resources.

802.15.4, by specification, is supported in three following modes: 868 – 868.6 MHz/single channel/20 kbps, 902-928 MHz/10 channels/40 kbps and 2400-2483.5 MHz/16 channels/250 kbps. Of these, only for the last one exists a possibility toS overlap with the standard ISM bands. As prior research suggests, the coexistence between 802.15.4 and 802.11 is an achievable task, but the following measures are highly recommendable in order to minimize risk of RFI: Frequency Division Multiple Access (FDMA) providing 16 channels, FHSS (Frequency Hopping Spread Spectrum) balancing the load between channels and CSMA/SA (Carrier Sense Multiple Access with Collision Avoidance), also known as “listen before talking”.

These strategies are advantageous in providing that simultaneous transmissions in the two networks will not overlap. However, in the scenario that a node from one networks detects that all possible channels are currently occupied, it must defer transmission to a later time period (hence the name "listen before talking"). This may cause unfairness. In order to reduce it and in accordance to the nature of management traffic, 802.15.4 packets can be limited to a size of 127 Bytes. The transmission of a single packet will require 50, 25 or 4 ms, depending on the chosen 802.15.4 mode.

The maximal overhead of the 802.15.4 MAC protocol is 39 Bytes (and 88 Bytes for payload). Of these, three fields are optional and thus candidates for optimization: destination address, source address and auxiliary security header.

The auxiliary header can be either 14 Bytes or fully omitted. However, out-of-band management provides full control over the underlying resources and therefore security is obligatory and cannot be sacrificed in order to achieve smaller overhead. But, as already described, destination and source addresses can be limited only to 2 Bytes. These 16 Bytes mean 41.03% overhead reduction and 18.19% increased payload.

B. Internet layer – IPv6

IPv6 introduces 40 Bytes of maximal overhead. However, a similar optimization is applicable here.

Some fields are obsolete. As the solution will be based only on IPv6, the 4 Bits for version field can be used for other tasks. Flow label (20 Bits) is experimental and not fully supported. It does not provide functionalities that are beneficial for the proposed framework. Finally, source and destination fields, 128 Bits each, can be fully omitted. The motivation behind this decision is that IPv6 is used only for the purposes of communication within the management network boundaries and the PHY layer contains enough information for addressing.

It is also possible to reduce the size of other fields. Halving the traffic class field (4 Bits) will provide 4 values and sufficient control over priorities. With less than 104 Bytes of payload, which is representable by 7 Bits. The "payload length" field is 16 Bits, but it can be reduced only to 11, as with future versions a larger 802.15.4 packet size may become available. Considering the relatively low distance between AAs and other AAs or AMs, 3 Bits will be enough for the hop limit field (currently 8 Bits).

In terms of overhead, these decisions lead to a reduction from 40 to 3 Bytes, which is 92.50%. The payload is increased from 64 to 101 Bytes (57.81%).

C. Transport layer – UDP

The source and destination port fields, 8 Bits each, can be reduced to 2 Bits. The management network relies on IoT devices which are constrained and more than 4 ports will not be required. The length field can be reduced from 16 to 12

Bits (considerations are similar to IPv6). With the checksum remaining unchanged (16 Bits), overhead is reduced from 16 to 4 Bytes (75.00%) and payload is increased from 85 to 97 Bytes (18.82%).

D. Application layer – MQTT-SN and CoAP

The MQTT (Message Queuing Telemetry Transport) protocol provides functionalities appropriate for remote telemetry querying. MQTT-SN (MQTT for Sensor Networks) is adjusted to the constraints of the IoT stack, but overhead here can be optimized further. Length and Type fields can be reduces to 11 and 5 Bits respectively. Different MQTT lifecycle tasks, such as automatic network discovery and topic (un)subscription require up to 16 Bytes. Most important is the topic publication task – in this case, overhead is 7 Bytes, leaving up to 90 Bytes for payload.

CoAP (Constrained Application Protocol) provides access capabilities to auxiliary elements based of web services. GET can be mapped to SNMP GET_REQUEST. PUT and SNMP SET_REQUEST are similar. DELETE can be used to remove objects from the MIB. The overhead here is up to 24 Bytes (8 Bytes for source and destination address, since this protocol will be used to provide remote access and 8 Bytes are required to support the core functionalities). This reduces payload to 73 Bytes.

E. Inner data representation

SNMP can be mapped to an inner protocol as a final optimization step as shown on Fig. 3. Three Bits are enough to represent different message types (such as TRAP or GET_REQUEST). Five Bits can be reserved for future usage.

The MIB Object ID is internally represented by 8 Bytes in a "Parameter" field. 1 Byte is enough to represent 256 discrete values, which are enough to cover information such as the current device temperature or port load. Thus the "value" field can contain between 1 and 4 Bytes. For certain events (such as device shutdown) the <parameter, value> tuple can be omitted and a simple code can be used instead. 2 Bytes are enough to represent 65,000 codes.

In order to support messages with different content (variable data length), three extra fields can be applied to every (code, parameter, value) triplet. Thus, 2 Bits are enough to demarcate the code length in Bytes (00-10 for 0-2 Bytes). The parameter field can be presented via 2 Bits, applying the following schema: 00 – no parameter, 01 – 2 Bytes, 10 – 4 Bytes, 11 – 8 Bytes parameter field. As different data types may require finer control over the value field length, 4 Bits can be used to represent its length (again in Bytes). Since the size-descriptors and real data fields are logically connected, they can be viewed as a "data group" entity, which is between 2 and 17 Bytes long.

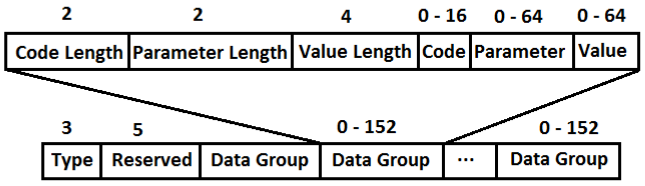

Fig. 3. Inner protocol (fields length in bits)

To sum up, 2 Bytes are enough to represent the SNMP type and future options. This leaves a payload of 71 (CoAP) or 88 (MQTT-SN) Bytes. Within these limitations, a data group can be fitted easily. This leaves enough space to embed several data groups within a single packet in order to reduce the overhead from other layers and emulate BULK_* operations.

IV. Results

The given practical approach covers all layers, required to implement and fulfill the requirements set in chapter II. It also provides a motivated set of decisions in order to minimize overhead from standard protocols. The possibility of coexistence (and collaboration) between standard (802.11) and IoT networks is proved via a constructional approach. In the practical approach no step depends on the particular management task that is to be executed and therefore this solution is widely applicable.

V. Conclusion

IoT is capable of improving and replacing current network solutions and also prepared to meet future requirements. The technology stack is still under a very dynamic development and this article presents a solution based only on current state of the art. But the framework has the potential to utilize future developments in the field of IoT.

References

[1] Haller S., S. Karnouskos, C. Schroth "The Internet of Things in an enterprise context", Vienna: Springer (Berlin-Heidelberg), pp. 14-28, 2008

[2] Pavlou G., "OSI Systems Management, Internet SNMP and ODP/OMG CORBA as Technologies for Telecommunications Network Management", Telecommunications Network Management: Technologies and Implementations, pp. 63-109, IEEE Press, 1998

[3] Stallings W., "SNMPv3: A security enhancement for SNMP", IEEE Press, 1998

[4] Bush S. F., A. B. Kulkarni, "Active Networks and Active Network Management", Kluwer Academic, New York, 2001

[5] Tan L., N. Wang, "Future Internet: The Internet of Things" in 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), August 2010