ADHT: Agent-based DHT Architecture for Constrained Devices

Erkki Harjula[1], Teemu Leppänen[2], Timo Ojala[2], Mika Ylianttila[1]
[1] Center for Internet Excellence (CIE)
[2] Department of Computer Science and Engineering (CSE)
University of Oulu, Finland
[1] firstname.surname@cie.fi, [2] firstname.surname@ee.oulu.fi

***Abstract*— In this paper, we propose a novel architecture, called ADHT, that allows the participation of wireless and constrained-capacity M2M devices in a P2P overlay without compromising their battery life. The approach makes P2P-based networking feasible in scenarios where dedicated super-peer nodes are not available or not feasible to use, e.g. when all the participating nodes in a subnet are constrained devices. The concept is based on sharing the peer responsibilities among the nodes within the same subnet using mobile agent-based virtual peers. The mobile agent-based approach enables on-the-fly configurations to further reduce energy consumption in the subnet nodes during runtime. The presented mobile agent composition is both programming language- and platform-independent. Therefore it inherently enables heterogeneous platforms, from the low-power resource-constrained embedded networked devices to the cloud platforms, to operate as the P2P peer. We analyze the power consumption characteristics of the mobile agent-based approach by comparing it with alternative architectures with a numerical analysis.**

Keywords—peer-to-peer, mobile agent, machine-to-machine, internet-of-things, energy-efficiency

I. Introduction

Due to extensive battery life requirements, low power consumption is a dominating design principle in constrained machine-to-machine (M2M) networks, such as wireless sensor networks (WSN), wireless sensor-actuator networks (WSAN) and wireless multimedia sensor networks (WMSN) [1][2][3].

The power consumption requirements concern both node-internal technologies and communication technologies. Node-internal measures to improve energy-efficiency include advanced power management functions and optimal use of sleep modes [4]. The measures concerning the communication between nodes include ultra-low-power short-range wireless technologies, lightweight communication protocols and optimized communication architectures. Wireless radio technologies, such as 6LoWPAN, ZigBee or Bluetooth Low Energy, can be used either as a primary communication channel, or as a low-power solution to wake up higher-capacity wireless interfaces such as WiFi [5]. Lightweight communication protocols and resource-optimized communication architectures are currently being developed to improve energy-efficiency on the system level [6][7][8]. Interoperability between hardware and software vendors requires universal standards on both radio link-layer and transport-layer protocols [9]. Constrained Application Protocol (CoAP) [10] is an essential protocol for providing global access for M2M systems over IPv6 networks.

A predominant trend in M2M systems is the rapid growth in the number of internet-connected M2M devices. It is predicted that M2M device connections worldwide will grow by a factor of 10-20, from around 2 billion devices in 2013 to around 20-50 billion devices in 2023 [11] [12]. The largest M2M systems will require scalability beyond millions of devices, where centralized solutions easily reach their bounds. To achieve universal access, M2M resources need to be globally identifiable, addressable and discoverable [13]. However, most solutions for improved scalability, such as [14], are limited to the scope of single access network.

Peer-to-peer (P2P) technology [15] is a widely used solution for creating highly scalable content discovery and distribution systems that minimize the need for large and expensive server infrastructures. In addition to practically infinite scalability, P2P systems also provide good failure tolerance, low investment cost, and minimal need for maintenance [15]. In P2P systems, peer nodes provide the needed routing, computing and storage resources. Depending on the application, servers are typically needed e.g. as seeding servers for distributed content, or in providing secure user authentication and authorization, but most of the actual data is exchanged directly between peer nodes. DHT-based structured P2P systems [16] have been proven efficient solutions for content distribution between always-connected personal computers. However, utilization of P2P in networks with battery-operated constrained-capacity devices has been so far unfeasible, due to high maintenance overhead that overutilizes their precious hardware resources and diminishes their battery life to unfeasible level [17][18].

In order to take full advantage of the superior scalability of P2P in M2M computing, there is a clear need to introduce such P2P technologies that can better adapt to constrained environment. In this paper, we propose a novel concept that allows participation of constrained-capacity nodes in a P2P overlay without compromising their battery life. The concept is based on sharing the peer responsibilities among nodes within the same subnet by using mobile agent-based virtual peers.

II. Related Work

A. P2P optimizations

The current workaround for allowing constrained-capacity devices to participate P2P networks is using super-peer architectures (Fig. 1a), where the most capable devices operate as *super-peers* that constitute the P2P *overlay*, and the rest of the devices operate as *clients* [16][19]. The clients connect to the overlay using peers as their gateways, and they neither provide resources to the overlay nor participate in the overlay management. Super-peer architecture can be optimized for M2M communications. For instance, [20] introduces a mechanism, where super-peers are used as "command mailboxes" in order to enable asynchronous communication with M2M nodes sleeping most of their time. The problem with super-peer approach is that it requires at least one higher-capacity device in a subnet to act as a super-peer. This kind of centralized structure within a subnet is prone to failures of the selected peer node. Alternatively, the super-peer can be selected among the wireless sensor or actuator nodes in M2M networks. In this case, the node would most likely be overloaded and its battery life would become unfeasibly short.

Multi-level super-peer architectures [21] have been used to cope with the single point-of-failure problem. More advanced node hierarchies help distributing the load with higher granularity between nodes with different hardware capabilities, such as the nodes in M2M systems. However, hierarchical architectures require more complex implementations for maintaining more complex overlay structures, whereas simple and lightweight implementations would be preferred to spare the precious hardware resources.

mDHT [22] (Fig. 1b) is an architectural enhancement for structured P2P networks. In mDHT, P2P host nodes in a subnet appear in a DHT overlay as a single node. Queries are routed as usual until they reach the destination subnet, where they are resolved among the hosts using multicast. The model is not suitable for M2M as is, since multicast query routing causes high signaling overhead in subnet. However, the fundamental idea of sharing peer responsibilities among all P2P nodes in a subnet provides an interesting basis for further optimizations.

B. Mobile agents in distributed computing

In this paper we use mobile agents to compose peer entities in M2M-optimized P2P systems. The mobile agents are autonomous programs capable of controlling their own execution and migration between system devices. The agent composition may also include the data required for the computation and the current state of the computation. As the state migrates, the computation can be continued in the new device hosting the agent.

The use of mobile agents in distributed systems provides several benefits [13][23]. First, computational load can be distributed in the system devices to reduce network load and communication costs. Second, mobile agents provide dynamic adaptation to the system configuration changes. Third, instead of updating the software in every device in the system, the agent functionality itself can be modified. Finally, agents provide some robustness against network, device or system failures. The current mobile agent implementations and applications are somewhat programming language, hardware platform and operating system-specific, where notable examples include the Java-based JADE framework [24]. However, Java implementations require Java virtual machines to execute the agents, which may be too burdensome to run in networked embedded devices with limited hardware resources.

In our recent research, interoperability has been studied in order to enable mobile agents in heterogeneous embedded networked systems over disparate networks [13]. We believe that the previously presented mobile agent composition can be extended to present the peer functionality in P2P systems, as described in this paper.

III. AGENT-BASED DHT (ADHT) ARCHITECTURE

Our Agent-based DHT peer concept (ADHT) allows the participation of low-capacity nodes, such as sensors and actuators, in P2P overlays as peers, without compromising their battery life. The concept is based on sharing the peer responsibilities among low-capacity nodes within the same subnet using mobile agent-based virtual peers. In ADHT architecture, a cluster of *sub-peer* nodes in a subnet collaboratively take care of the peer responsibilities (Fig.1c)

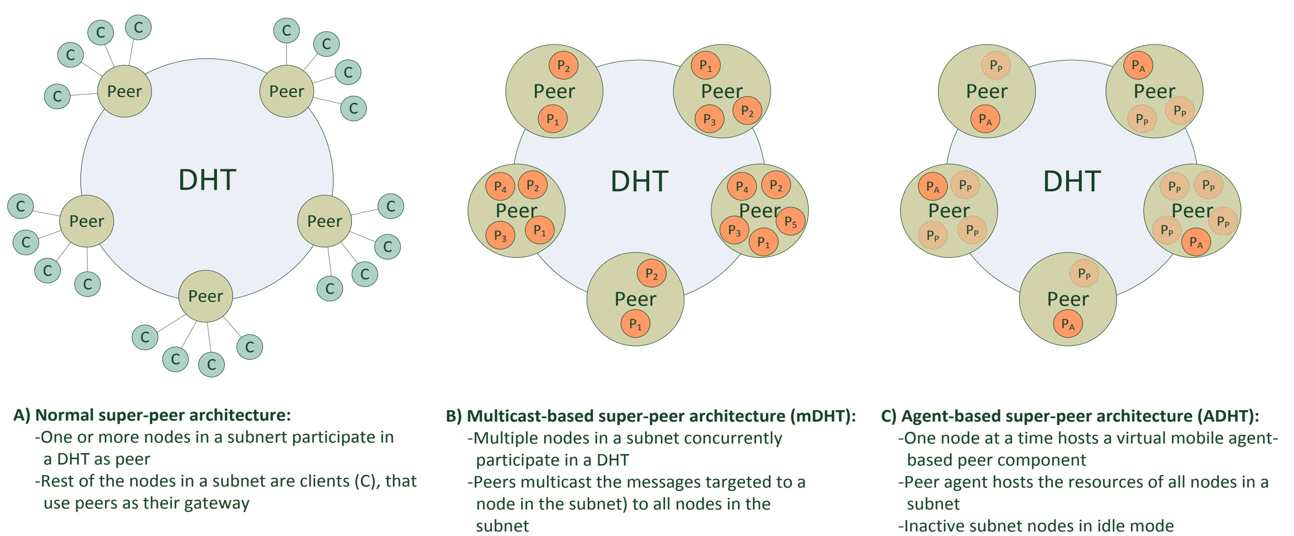

Fig. 1. Super-peer architectures including our Agent-based DHT Architecture (ADHT).

Sub-peer refers to a node which is allocated for acting as a peer when needed. The basic principle is that one node among all the participating sub-peer nodes in a subnet takes care of the peer responsibilities while the other nodes are in sleep mode (or act as clients, depending on the used *data delivery mode*, described in Section III B).

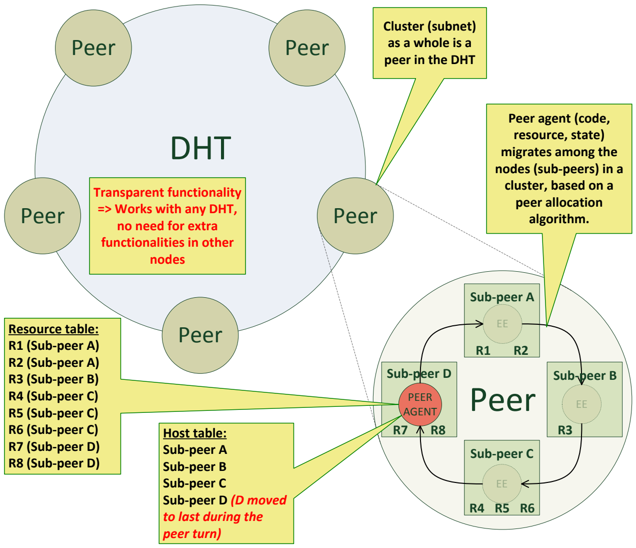

Fig. 2. Architecture of the ADHT.

The novelty of the architecture is in circulating the peer role between the sub-peer nodes in a subnet by using a mobile agent (*peer agent*) that migrates among the sub-peers. *Peer* in our system is a virtual peer, represented by the peer agent, identified by a virtual IP address, and located in one of the sub-peers. During runtime, different sub-peers within the same subnet host the peer on their turns, based on the *peer allocation algorithm*. The allocation algorithm can be a simple time share-based algorithm, where each sub-peer hosts the peer at its turn for a pre-defined time period, or it can be more sophisticated dynamic algorithm, see Section III A.

While active in a sub-peer node, peer updates its resource tables concerning that node, and keeps the data available in resource tables when migrating to other nodes. The principle is illustrated in Figure 2.

In order to maintain successful operation of a DHT, the peer agent should be reachable with a single identity on the overlay. Thus, the basic IP-based node ID does not work in this case, since all sub-peers have their own IP-addresses. There are two ways to solve the problem, of which either can be used with ADHT. The first option is to configure the subnet gateway to use "virtual server" function in order to redirect the inbound messages for the currently active sub-peer node. In this option, the gateway should be notified every time the peer agent has been successfully handed over to a new sub-peer. Most off-the-shelf routers provide this functionality. This option is transparent for the overlay, i.e. no overlay actions are involved in the peer-agent handover. The second option is that sub-peers join the overlay on their turns and leave the overlay when handing over the peer agent to next sub-peer, but share the same node ID. In this case, the peer-agent handover is not transparent for the overlay. However, the handover is seen from the overlay like a peer changing its network. Thus, the extra overhead for the overlay is minimal since the overlay topology does not change.

A. *DHT Mobile Agent Composition*

The virtual peer is represented by a mobile agent, based on the composition presented in [13], where the composition includes *code*, *resource* and *state* segments. The composition migrates between the subnet nodes, according to their available resources. In this paper, we extend this composition to consider P2P peer functionality. As the composition is dynamically modifiable during runtime, we can enable dynamism and reflect to changes in the subnet topology. Additional benefit is that the DHT and peer allocation algorithms, as well as routing and resource tables can be utilized remotely as system resources, reducing the size of the agent composition in very large P2P systems. The peer agent composition is illustrated in Table I, and the segments are illustrated in the following Subsections.

TABLE I. PEER AGENT COMPOSITION

Segment	Content				
Code	***DHT algorithm***				
	{ Defined by the used DHT algorithm }				
	Peer allocation algorithm				
	Sequence[0..n] <- Resource[0..n]				
	Data preprocessing algorithms				
	Name = { … }				
Resource	***DHT routing table***				
	{ Routing/neighbor tables defined by the used DHT algorithm }				
	Subnet resource table				
	Seq. number	***Identifier***	***Address***	***Data access***	***Status***
	0	Name	Node IP address	{Local | RR | PS}	{Active | Sleeping | Disconnected}
State	***Identifier***		***Value***		
	Name		{ Value representation }		
Metadata	{ … }				

PS = Publish-subscribe RR = Request-reply

1) *Code segment*

The distinct capability of mobile agents is that they are able to control their own execution and migration dynamically. The code segment contains the DHT algorithm and the peer allocation algorithm. The DHT algorithm runs the P2P protocol and takes care of peer responsibilities for the overlay. Peer allocation algorithm determines the most suitable sub-peer as the next host for the agent (peer). The ranking of the most suitable sub-peers is represented in the *host table*. The modifications to the host table are reflected to the resource segment below. Separating the peer allocation algorithm from the resource tables and caching enables runtime updating and adjustments for the algorithm, based on for example machine learning methods. Several algorithms, whether application-specific or not, can be run in parallel and even new algorithms introduced into the system during runtime as a mobile agent. We can, for example, include data preprocessing and filtering or machine learning algorithms to further reduce the resource access in sub-peers. These capabilities may assist in the evaluation of the different algorithms and additionally improve energy efficiency. The modifications to the data are reflected to the state segment.

2) Resource Segment

The resource segment contains the DHT routing table and the subnet resource table, which is a dynamic table of the identifiers and addresses of the stored resources in sub-peers. This table is updated whenever a new sub-peer joins or leaves the subnet. The segment also lists the sub-peer state, whether it is “active”, “sleeping” or “disconnected”. The data delivery mode to the subnet nodes is determined by the sub-peer and included in the resource table, when it joins the subnet.

3) State segment

The state segment contains the cache of the overlay and sub-peer data. The sub-peer data may be either raw or refined, in order to reduce the communication costs in the subnet peers and to enable extensive periods of sleep mode in the subnet nodes. The cache update mechanisms are resource-specific.

The agent migration in the subnet is based on sequential numbers of the resources as an array. The supported data delivery modes are “time-share” where the peer hosts the data, “publish-subscribe” where the peer subscribes to the resource updates from the sub-peers and “request-reply” where the peer requests the sub-peers for current data. “Request-reply” and “publish-subscribe” data delivery modes are feasible options with heterogeneous systems, where the sub-peer is considered as a client for a server (peer) in the subnet. This is additional factor for reducing energy consumption of the sub-peers.

The sub-peer status is “active” when it communicates outside the subnet, “sleeping” when the sub-peer is idle and “disconnected” when disconnected from the subnet.

B. Data delivery modes

Based on the peer agent composition, ADHT provides three data delivery modes that have different characteristics concerning the data freshness and hardware resource consumption. Different nodes can have different data delivery mode within the same subnet: nodes can set up their data delivery mode while hosting the peer agent or when joining the subnet.

1) Time-share

In this mode, the sub-peer data is refreshed to the peer agent’s state segment the next time the peer agent migrates to the sub-peer hosting the resource. When data request is received, the peer returns the data on the mobile agent state on behalf of the sub-peer. Figure 3 illustrates the described scenario. The main principle is that the sub-peer does not need to interact with other sub-peers while the peer agent is located in another sub-peer, allowing it to sleep most of the time. This enables significant savings in power consumption in wireless nodes where frequent signaling substantially increases the power consumption. Table II illustrates the relevant parts of the mobile agent composition for this mode. The next peer is determined by the order of the nodes in the resource segment, where the current peer is listed as the last item (Resource[n]) and is the only active sub-peer in the subnet. The current peer will be moved as the last item in the segment when the agent migrates to the next sub-peer.

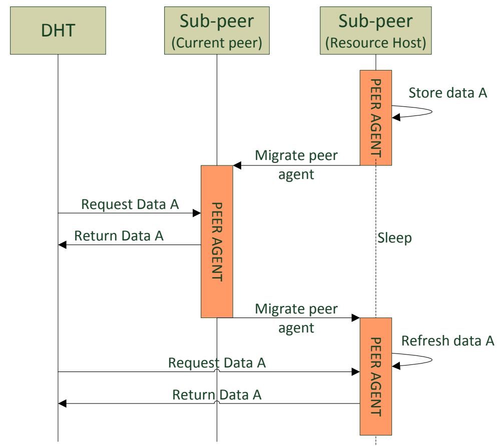

Fig. 3. Time-share data delivery mode.

TABLE II. PEER AGENT COMPOSITION IN TIME-SHARE MODE

Code	*Sequence[0] <- Resource[0]*
…	

2) Publish-subscribe

In this mode, sub-peers frequently update their data to the peer, in order to achieve higher freshness of the data. The refresh rate is defined by sub-peers. Figure 4 and Table III illustrate this scenario. The sub-peers are sleeping, except while acting as peer or while publishing their resource state into the peer. The selection of the next peer shall be based on an algorithm that favors the sub-peers hosting more local resources as the next peer, in order to minimize the interactions between the peer and sub-peers.

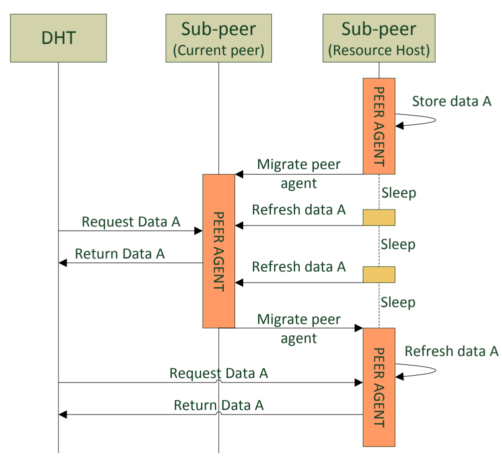

Fig. 4. Publish-subscribe data delivery mode.

TABLE III. PEER AGENT COMPOSITION IN PUBLISH-SUBSCRIBE MODE

Code	Sequence[0] <- max { Local resources }
…	

3) Request-reply

In this mode, the peer forwards all incoming requests to the target sub-peers in order to achieve high freshness of the data. However, since the resource hosting sub-peers sleep when they are idle, they need to wake-up from the sleep mode in order to reply to these requests, which cause some delay. The operational principle of this data delivery mode is close to traditional super-peer model, with the exception that the static (super-) peer is replaced by the dynamic virtual peer. This mode is also suitable for scenarios where the resources require a lot of storage space, and therefore would cause too heavy agent migration cost. Figure 5 and Table IV illustrate the described scenario. The migration sequence shall be based on the resource access frequency, resource size or minimizing the hop distance from the peer node to the sub-peers, in order to minimize the data transfers between the peer and sub-peers.

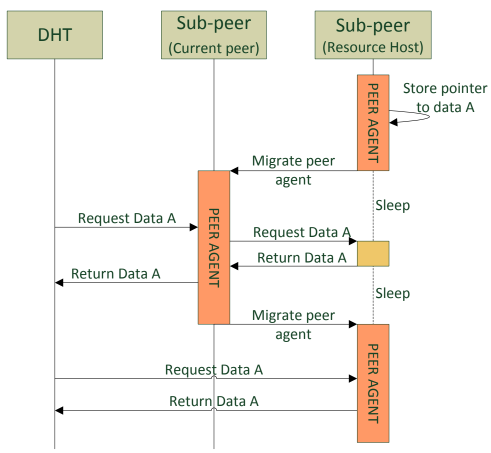

Fig. 5. Reqest-reply data delivery mode.

TABLE IV. PEER AGENT COMPOSITION IN REQUEST-REPLY MODE

Code	*Sequence[0] <- max { Access Freq / Size} / min { Distance }*
…	

IV. PERFORMANCE ANALYSIS

A. *Overhead*

Since the target platforms for ADHT are embedded networked M2M systems, the communication overhead of ADHT needs to be carefully considered. Our previous work [15] presented a mapping of the mobile agent composition into a CoAP message, targeted for low-power resource-constrained embedded nodes. In Table V we approximate the size of the agent composition migrating between the sub-peers. To mitigate the migration cost of the code segment, it can be excluded from the composition as soon as the agent has been hosted at least once by each peer, and the code has been programmed into their memory. However, a reference to the original code location and a checksum may be needed to address the code updates. Additionally, the peer agent needs to keep track of the different data delivery modes of the nodes and their state. However, additional overhead is introduced by caching the rarely requested resources in the composition. In Table V, The IPv6 network address prefix has been omitted from the address, as we assume the resources are within the same network. The remaining address size will be 8 bytes and we reserve 8 bytes for individual resource name.

TABLE V. THE SIZE OF A MOBILE AGENT COMPOSITION

Code	Reference	IPv6 address + Name		16B
Resource	0	IPv6 address + Name	Status byte	17B
	…	…		…
	n	IPv6 address + Name	Status byte	17B
State	Name[0]	{ Value representation }		x_0 B
	…	…		…
	Name[n]	{ Value representation }		x_n B
Metadata	…			

To illustrate a typical agent composition size, we approximate its segment sizes for a M2M subnet of 10 wireless sensor nodes, each providing simple numerical sensor data and storing the measurement history for some time. With simple time share-based peer allocation algorithm, the size of the code segment, including the DHT and peer allocation algorithms, would be in magnitude of some kilobytes. Furthermore, the resource segment would be in magnitude of some kilobytes and the state segment in magnitude of tens of kilobytes. Thus, a realistic agent composition size in this scenario would be in magnitude of a hundred kilobytes.

A. *Power consumption*

In order to analyze the performance of the ADHT model, we made a comparative analysis of it against a flat P2P architecture, a super-peer architecture and a multicast-based super-peer architecture (mDHT). Based on the operational principles of different architectural models and their organization of the networks, we calculated an approximation for their power consumption per node in a stabilized overlay.

The following scenario was used in the analysis: A M2M subnet of 10 wireless sensor nodes, each hosting 5kB sensor data on average, connected to a large (>1k node) overlay, resulting in the peer agent composition sized around 100 kB (see Section IV A) The sensor nodes were assumed to be equipped with advanced power management functions, WiFi (802.11g) connection and sufficient hardware to act as a peer in a P2P overlay. We also assumed that the nodes have the ability to enter deep sleep mode and wake-up with a feasible delay, triggered by incoming request through the WiFi.

Table VI presents an approximation of the power consumption for each architecture when WiFi is used, based on the literature [1-3][5][25] and our previous knowledge of the radio interface basic characteristics [26]. The average power consumption of different types of power-consuming elements are presented on the 4th row, and the average number of these elements in a 10-node subnet in the rows 6-11.

Peer: The approximated power consumption of a peer node in flat P2P architecture. The peer functions include e.g. overlay topology maintenance, resource storage and query and message routing. In practice, peer communicates intensively with the overlay. The maintenance messaging dominates the power consumption, since it consists of frequent incoming and outgoing small packets that keep the radio interface active even though the amount of data transferred is relatively low.

TABLE VI. POWER CONSUMPTION OF DIFFERENT P2P ARCHITECTURES

		Power consumption elements								Power/node [mW]
		Peers	Super-Peers (&Req/ reply ADHT peers)	Clients (&Req/ reply sub-peers)	Collaborative Peers	ADHT peers	Pub-Sub sub-peers	Sleeping sub-peers	Peer Migrations	
		Average power consumption [mW]								
		400	450	30	375	425	10	1	20	
Architectural model		**N/Scenario**								
Flat		10	0	0	0	0	0	0	0	**400**
Super-Peer		0	1	9	0	0	0	0	0	**72**
mDHT		0	0	0	10	0	0	0	0	**375**
ADHT	**Time-share**	0	0	0	0	1	0	9	10	**63.4**
	Pub-sub	0	0	0	0	1	9	0	10	**71.5**
	Req-reply	0	1	9	0	0	0	0	10 * 0.1	**74**

Super-peer: The approximated power consumption of a super-peer node in super-peer architecture. In addition to the peer functions, super-peers provide gateway functions for their clients, including query/message forwarding, resource storage/retrieval on the DHT on behalf of the client nodes, and client maintenance. In practice, this means additional burden on the node. However, the effect of the additional communication on the power consumption is relatively low, since the peer role-related messaging dominates the traffic.

Client & Req/reply sub-peer: The approximated power consumption of a client node in super-peer architecture and a sub-peer in the ADHT architecture with request-reply data delivery mode. Client nodes serve the local applications by functions such as reception and processing of incoming messages, sending outgoing messages, and storing/retrieving resources. In practice this means scattered messages and data transfers, i.e. for most of the time, the radio interface is idle.

Collaborative peer: The approximated power consumption of the nodes that share the peer responsibilities in the mDHT architecture. The power consumption of these nodes is lower than of a normal peer, but since mDHT optimizes routing only when the message/query is targeted to the subnet, the difference is low. The maintenance messaging and query routing to nodes from outside the subnet are intact, and since they form major part of the messaging of a peer, the total power consumption is not much lower than of a regular peer.

ADHT peer: The approximated power consumption of a peer agent node in the time-share and publish-subscribe modes of our ADHT model. The maintenance overhead caused by mobile agent-based approach is marginal, since the system was originally designed for very constrained-capacity embedded devices [13]. In the request-reply mode, the functionality of ADHT peers is very close to traditional super-peers, thus we have listed the ADHT peer in request-reply mode in the “super-peer” category. ADHT peer in the time-share and publish-subscribe modes works as a gateway node for the sub-peer nodes, as with super-peer nodes, but the interaction with them is less intensive. Thus the estimated power consumption is lower than with regular super-peers.

Publish-subscribe sub-peer: The approximated power consumption of sub-peers in the publish-subscribe mode of ADHT, that periodically wakes up to e.g. read sensor information and send resource update messages to their subnet’s ADHT peer node.

Sleeping sub-peers: The approximated power consumption of (sleeping) sub-peers in our ADHT model that do not have any interaction with other nodes between the periods of hosting the peer agent.

Peer migration: The approximated average power consumption caused by peer agent migrations from a sub-peer to another. We assume the size of peer agent computation to be around 100kB (see Section IV A). In time-share and request-reply modes, the shared resources of each sub-peer that have changed during the hosting period are moved along with the peer agent. In request-reply data delivery mode, only the peer agent with the resource tables with pointers to the resources in the sub-peers is moved. Based on this, we estimate the migration cost of request-reply mode to be around 10% of the migration cost of other data delivery modes.

From the results, we can see that the average total power consumption per subnet node in our ADHT architecture is on the same or on lower level than with the traditional super-peer architecture, depending on the data delivery mode in use. When compared to the flat P2P architecture, the power consumption savings are significant. The more important difference is, however, the even distribution of the load among the subnet nodes over time, due to the frequent peer migration from node to node. The benefit is clear when compared to traditional super-peer architecture, where the (static) super-peer node consumes up to 450mW while the clients consume 30mW. It is obvious that if the super-peer is one of the M2M nodes, its battery life would be unfeasibly short. Thus, ADHT clearly helps removing the super-peer overload problem of a super-peer architecture while maintaining the scalability and fair load balance among other benefits of the P2P approach.

V. CONCLUSION

In this paper, we proposed a novel architecture, called ADHT, which allows the participation of wireless and constrained-capacity M2M devices in a P2P overlay without compromising their battery life. This approach makes P2P-based networking feasible in scenarios where dedicated super-peer nodes are not available or not feasible to use, e.g. when all the participating nodes in a subnet are constrained devices. The concept is based on sharing the peer responsibilities among the nodes within the same subnet using mobile agent-based virtual peers. The mobile agent-based approach has the capability to distribute the adjustable super-peer functionality dynamically in runtime and enable different data delivery modes within the same subnet, while simultaneously enabling caching to further reduce energy consumption in the subnet nodes. The presented solution is both programming language- and platform-independent. Therefore it inherently enables heterogeneous platforms, from the low-power resource-constrained embedded networked devices to high-end devices, to operate as the super-peer.

We analyzed the power consumption characteristics of the proposed architecture, and compared it with alternative architectures with a numerical analysis. The results indicate that our approach removes the super-peer overload problem, while preserving or even improving the average power consumption on the subnet level, when compared with the most feasible existing architecture.

In our previous work, the mobile agent-based approach has already been proven feasible in WSN scenarios through evaluations with real-life prototypes. The future work includes implementing the P2P part of the ADHT model for real-world devices, in order to further evaluate the model's performance in real-world distributed M2M scenarios. These evaluations would give further information on e.g. in what degree the agent size, the subnet size and the migration latency affect the hardware requirements and energy efficiency.

Acknowledgments

The financial support from TEKES, Infotech Oulu, and foundations of HPY, Nokia, TeliaSonera, Walter Ahlström, Seppo Säynäjäkangas, KAUTE, Riitta & Jorma J.Takanen, EIS, Tauno Tönning and TES is gratefully acknowledged.

References

[1] G. Xing, C. Lu, Y. Zhang, Q. Huang, and R. Pless. "Minimum power configuration for wireless communication in sensor networks," ACM Trans. Sen. Netw. 3, 2, Article 11, June 2007.

[2] P. De Mil, T. Allemeersch, I. Moerman, P. Demeester,W. De Kimpe, "A Scalable Low-Power WSAN Solution for Large-Scale Building Automation," Communications, 2008. ICC '08. IEEE International Conference on , pp.3130-3135, May 2008.

[3] M. Yang, D. Wang and N. Bourbakis, "Optimization of Power Allocation in Multimedia Wireless Sensor Networks," International Journal of Monitoring and Surveillance Technologies Research, vol. 1, no. 1, 2013.

[4] A.G. Jurdak, Ruzzelli, and G.M.P. O'Hare, "Radio Sleep Mode Optimization in Wireless Sensor Networks". IEEE Transactions on Mobile Computing 9, 7, pp. 955-968, July 2010.

[5] H. Qin; W. Zhang, "ZigBee-assisted Power Saving Management for mobile devices," Mobile Adhoc and Sensor Systems (MASS), 2012 IEEE 9th International Conference on, pp.93-101, 8-11 Oct. 2012.

[6] D. Baghyalakshmi, J. Ebenezer, S.A.V SatyaMurty, "Low latency and energy efficient routing protocols for wireless sensor networks," Wireless Communication and Sensor Computing, 2010. ICWCSC 2010. International Conference on , pp.1-6, 2-4 January 2010.

[7] D. McIntire, T. Stathopoulos, S Reddy, T Schmidt, and WJ. Kaiser, "Energy-Efficient Sensing with the Low Power, Energy Aware Processing (LEAP) Architecture," ACM Trans. Embed. Comput. Syst. 11, 2, Article 27, 36p, July 2012.

[8] J. Kim, J. Lee, J. Kim, J. Yun, "M2M Service Platforms: Survey, Issues, and Enabling Technologies," Communications Surveys & Tutorials, IEEE , vol.16, no.1, pp.61-76, 2014.

[9] D. Lee; J-M. Chung; R.C. Garcia, "Machine-to-machine communication standardization trends and end-to-end service enhancements through vertical handover technology," Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on , pp.840-844, Aug. 2012.

[10] Constrained Application Protocol (CoAP), draft-ietf-core-coap-18 (Accessed 21st November 2013), https://ietf.org/doc/draft-ietf-core-coap/

[11] "M2M device connections and revenue: worldwide forecast 2013–2023" (Accessed 25th November 2013), Analysys Mason - Forecast report, http://www.analysysmason.com/Research/Content/Reports/M2M-connections-forecast-Aug2013_RDME0/

[12] "The Internet of Things" (Accessed 25th November 2013), CISCO Infographic, http://share.cisco.com/internet-of-things/

[13] T. Leppänen, M. Liu, E. Harjula, A. Ramalingam, J. Ylioja, P. Närhi, J. Riekki, and T. Ojala. "Mobile Agents for Integration of Internet of Things and Wireless Sensor Networks," IEEE International Conference on Systems, Man and Cybernetics, Manchester, UK, pp. 14-21, October 2013.

[14] C-Y. Hsu, C-H. Yen, C-T. Chou, "An adaptive multichannel protocol for large-scale machine-to-machine (M2M) networks," Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International , pp.1223-1228, July 2013.

[15] S. Androutsellis-Theotokis and D. Spinellis. "A survey of peer-to-peer content distribution technologies," ACM Comput. Surv. 36, 4 pp.335-371, December 2004.

[16] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, S. Lim, S, "A survey and comparison of peer-to-peer overlay network schemes," *Communications Surveys & Tutorials, IEEE* , vol.7, no.2, pp.72-93, 2005.

[17] Z. Ou, E. Harjula, O. Kassinen, M. Ylianttila. "Performance Evaluation of a Kademlia-based Communication-oriented P2P System under Churn," Elsevier Journal of Computer Networks, vol. 54, no 5, pp. 689-705, 2010.

[18] I. Kelenyi and J.K. Nurminen, "Energy Aspects of Peer Cooperation Measurements with a Mobile DHT System," Communications Workshops, 2008. ICC Workshops '08. IEEE International Conference on, pp.164-168, May 2008.

[19] B. Yang, H. Garcia-Molina, "Designing a super-peer network," International Conference on Data Engineering. Bangalore, India: 49–60.

[20] J. Jiménez Bolonio, M. Urueña, G. Camarillo, "A Distributed Control Plane for the Internet of Things Based on a Distributed Hash Table," Mobile Networks and Management, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 125, pp. 108-121. Springer, 2013.

[21] Z. Ou Z, "Structured peer-to-peer networks: Hierarchical architecture and performance evaluation," Dissertation, Department of Electrical and Information Engineering, University of Oulu, Finland, 2010.

[22] J-W. Lee, H. Schulzrinne, W. Kellerer and Z. Despotovic, "mDHT: Multicast-augmented DHT Architecture for High Availability and Immunity to Churn", IEEE CCNC, Las Vegas, Nevada, USA, January 2009.

[23] I. Satoh, "Mobile agents," In: Nakashima et al. (Eds.), Handbook of Ambient Intelligence and Smart Environments, pages 771-791, Springer, 2010.

[24] F. L. Bellifemine, G. Caire and D. Greenwood, 2007. Developing multi-agent systems with JADE, Vol. 7, John Wiley & Sons. 2007.

[25] E.J. Vergara and S. Nadjm-Tehrani, "EnergyBox: A Trace-Driven Tool for Data Transmission Energy Consumption Studies," EE-LSDS, Lecture Notes in Computer Science, vol. 8046, pp. 19-34. Springer, 2013.

[26] E. Harjula, O. Kassinen and M. Ylianttila, "Energy consumption model for mobile devices in 3G and WLAN networks," IEEE CCNC, Las Vegas, Nevada, USA, pp.532-537, January 2012.