Delay Contributing Factors and Strategies Towards Its Minimization in IoT

Wan Fariza Wan Abdul Rahman[1], Md. Rafiqul Islam[2], Aisha Hassan Abdalla Hashim[2]
[1]*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Malaysia.*
[2]*Department of Electrical and Computer Engineering, Kuliyyah of Engineering,*
International Islamic University of Malaysia, Malaysia.
wfariza@kelantan.uitm.edu.my

***Abstract*— Internet of Things (IoT) refers to various interconnected devices, typically supplied with limited computational and communication resources. Most of the devices are designed to operate with limited memory and processing capability, low bandwidth, short range and other characteristics of low cost hardware. The resulting networks are exposed to traffic loss and prone to other vulnerabilities. One of the major concerns is to ensure that the network communication among these deployed devices remains at required level of Quality of Service (QoS) of different IoT applications. The purpose of this paper is to highlight delay contributing factors in Low Power and Lossy Networks (LLNs) since providing low end-to-end delay is a crucial issue in IoT environment especially for mission critical applications. Various research efforts in relevance to this aspect are then presented.**

***Index Terms*— IoT; QoS; Low power and lossy networks; End-to-end delay.**

I. INTRODUCTION

The emerging Internet of Things (IoT) technology is based on the interconnected smart objects (devices) working together. These objects, with Internet Protocol (IP) connectivity, typically embedded with sensors, thus can be assessed, controlled and managed anytime from anywhere regardless of their location. A network formed by this type of devices is compatible with the IEEE 802.15.4 standard and known as the IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN).

Unfortunately, this type of network poses unique challenges to a routing solution. This is due to a large number of constrained nodes with limited processing power and memory. The routers are interconnected by unstable lossy links, typically supporting only low data rates, contributing into relatively low packet delivery rates [1]. This aspect is a major concern as many IoT applications operating in inhospitable setups; e.g. battlefields, whereby the nodes are at risk of getting damaged. Apart, the operation of nodes equipped with limited batteries power will cease upon depleting their onboard energy supply. Consequently, the nodes failure may not only affect the coverage and data fidelity, but also can divide the network into disjoint blocks of nodes.

The dynamicity of IoT environment may cause delay in the process of collecting, storing and manipulating the information [2]. This at the same time poses new challenge to the existing protocol. The more dynamic the topology, the more routing, transport and application layer protocols have to cope with interrupted connectivity and/or longer delays. Thus, in such situation, a transport layer protocol may fail to operate if it expects no route changes during a communication flow [3].

A. Routing Requirements and Common IoT Applications

Common IoT applications include urban networks, building automation, industrial automation, and home automation. A routing protocol therefore should satisfy specific routing requirements considering various applications. Following are list of routing requirements specified by ROLL Working Group [4]:

a. Unicast, anycast, and multicast traffics should be supported
b. Adaptive routing – new paths are dynamically and automatically recomputed based on conditions (link/node failure, mobility, etc.) change in the network
c. Routes optimization considering different metrics (e.g. minimized latency, maximized reliability, etc.)
d. Able to find a path that satisfies specific constraints such as providing a path with a latency lower than a specified value
e. Constraint-based routing – Consider various node constraints (such as energy, CPU, and memory) as well as link attributes (such as link latency)
f. Provide support of (i) multipoint-to-point (MP2P); most of the traffic is from leaf nodes such as sensors to a data collection sink, (ii)point-to-multipoint (P2MP); such as the sink sends a request to all nodes in the network, and (iii) point-to-point (P2P) traffics; communication between devices in the network
g. Scalable – expected to support millions of nodes or more
h. Secure – in most cases such as Smart Grid, building automation and industrial automation, authentication and encryption are necessary

Different types of applications have different delay requirements. In healthcare environments dealing with critical patients, for example, delayed or lost information may be a matter of life or death. In certain other applications (such as a remote control to switch a light on) which do not involve critical data, the response time and network delays must still be maintained to be within a few hundred milliseconds for optimal user satisfaction [5]. For applications such as air conditioning and other environmental-control applications, response delays of ten of seconds or longer may be accepted. On the other hand, alarm and light control applications which regarded as soft-real time systems accept a slight delay. However, the perceived quality of service degrades significantly if response times exceed 250 ms. Similarly, in the telecom industry, if the voice delay exceed 250 ms, users start getting confused, frustrated or annoyed [6].

Delay is one of the major concerns as it is a significant indicator to calculate response time. Response time can be defined as the time taken for a message to travel from the sending device, through the sensor network to the receiving device. From the query-based WSN applications point-of-view, end-to-end delay refers to the time taken to send a query to the source and receive an answer by the transmitting base station (sink) in response to the request query [7]. Delayed data transmission may lead to the execution of incorrect actuation command and inappropriate management decision. Therefore, maintaining a specified delay in accordance to the application requirements is really a matter in IoT to ensure Quality of Service (QoS).

Before going into details about delay issues in IoT, the following subsection will first highlight about IoT standard routing scheme - RPL.

B. Routing Scheme - RPL

The IETF ROLL working group has defined an IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [3] as a standard routing for LLNs. RPL is based on Directed Acyclic Graph (DAG) whereby all paths are oriented toward and terminating at root node(s) called DAG root. The term Destination-Oriented DAG (DODAG) refers to a DAG rooted at a single destination with no outgoing edges. DODAG Information Object (DIO) messages are sent among nodes to construct and maintain these DODAGs. Two important elements in RPL are Rank and Objective Function (OF). The node’s individual position relative to other nodes (with respect to a DODAG root) is stated as Rank. Rank decreases from leaf nodes towards DODAG root, and increases from DODAG root towards leaf nodes. Rank is determined from routing metrics, optimization objectives, and related functions defined by OF. The stability of the routing topology is influenced by the stability of the Rank [3].

Each node sends link-local multicast DIO messages to other nodes to advertise itself, its routing cost and related metrics. The information in DIOs is used to join a new DODAG, select DODAG parents or to maintain the DODAGs. Besides, nodes use Destination Advertisement Object (DAO) messages to propagate destination information Upward along the DODAG through next-hop destinations called DAO parents. The Downward traffic, on the other hand, can either be forwarded in Storing or Non-Storing mode. In Storing mode, the stored routing states are used by a common ancestor of the source and the destination to direct Down the packet towards the destination prior to reaching a DODAG root. In contrast, Non-Storing mode requires the packet to travel all the way to a DODAG root before travelling Down [3].

II. RPL CHALLENGES - STORING VERSUS NON-STORING MODE

In practical, as the size of LLN deployments increase, a homogeneous Non-Storing mode network will incur a high level of communication overhead, and a homogeneous Storing mode network will require too much memory resources. The Non-Storing mode requires little memory for storing routing states. Thus, it provides an advantage for the devices with limited processing and storage capabilities. However, the Non-Storing mode requires a source routing header (SRH) to be attached to all packets. With this attachment, the packet size not only increases, but also becomes variable depending on the path length. This in turn decreases the effective maximum transmission unit (MTU) of the packet and an IPv6 packet might be fragmented into more than 16 fragments at the 6LoWPAN sublayer. Considering a multihop IoT network, hop-by-hop re-composition at each hop (to reform the packet and route it) contributes additional latency. The intermediate nodes are forced to store packets for an undetermined time, thus, giving impact on critical resources such as memory and battery [8].

On the other hand, the Storing mode does not have the long-route problem since it does not need for SRH. However, each RPL node is required to store route information for all destinations in its own subtree. Considering limited memory constraints of small embedded devices, this may lead to another problem. Although initial RPL standard does not support mixed-mode of operation where some nodes source route and other store routing tables, routing pathology issues in a mixed network of storing and non-storing nodes has been discussed in [1].

A. RPL Enhancements

A More Memory-efficient Storing mode RPL; MERPL has been proposed in [9] by suggesting that the number of routing table entry stored in a node should not exceed a pre-specified factor of N. When the number of routing table entries to be stored is larger than N, the node will transfer part of its responsibility to the selected child. However, no clear approach on how to determine the value of N has been defined.

A more efficient network with reduced delay and memory consumption can be achieved by allowing a mixed of computationally powerful nodes with route storing capabilities and low-cost nodes that do not need to maintain a routing table [1]. However, a mix of nodes operating in Storing and Non-Storing modes to form a single network can cause a routing pathology. Routing pathology can partition the network due to the scenarios where nodes cannot send packets to the root and the root cannot send packets to the nodes even though they have plenty of multi-hop physical connectivity in the network.

To eliminate the network partition problem and preserve the high bidirectional data delivery performance, works in [1] and [10] have proposed modifications on (i) DAO transmission and its format, (ii) SRH support, and (iii) storing mode flag. The purpose is to allow all DAO messages, from both Storing and Non-Storing mode nodes to be processed at intermediate nodes, and to resolve the ‘SRH’ problem for packets sent by Storing mode nodes.

Another work in [11] also proposed some RPL modifications to improve end-to-end delay estimation in WSN, involving a few changes in RPL metrics and Objective Function (OF). Two dynamic metrics; path delay and processing delay are to be considered in the selection of Preferred Parent.

III. DELAY CONTRIBUTING FACTORS AND POSSIBLE HANDLING APPROACHES

A. *RPL Loop and Repair*

When DIO messages lost, a DODAG loop may occur in RPL. In this case, a node detaches from the DODAG and reattaches to a device in its prior sub-DODAG [3]. To reduce control traffic overhead, RPL has been designed to repair loops only when detected by a data traffic transmission. However, repairing loops only when triggered causes delays in forward progress of data packets. This increases end-to-end delays. Additionally, buffer usage may also increase due to the buffered data packet during repair. For a number of IoT applications such as data acquisition in smart metering applications, an increased delay may be acceptable, but for applications such as alarm signals or in home automation, increased delay may be undesirable. In addition, due to memory constraint of IoT devices, buffering incoming packets during the route repair may not be feasible for all incoming data packets, leading to dropped packets. These packets may require retransmissions, or may be definitely lost (depending on the transport layer protocol) [12]. The impact of reactive repairing mechanisms used by RPL on the convergence time, power consumption and packet loss is shown in [13].

Careful implementation of loops avoidance can minimize the impact of loop detection in RPL [13]. The idea of loop prevention, detection and avoidance has been emphasized in [14]. However, each approach involved with certain costs. The loop prevention approach requires a node to wait for a sequence number update by the DODAG root (global repair) before increasing its Rank in order to choose new parents. The loop detection approach puts costs on the already small available space for carrying the data by requiring the node tags to be carried in the packet. On the other hand, the loop avoidance causes dismantling the sub-DAG rooted at the node performing the Rank increase. If not carefully handled, this action can be too pricy with a minor change in DAG structure. Based on simulations performed in [14], loop avoidance in a DAG based routing protocol is not recommended. The turmoil caused by dismantling of the sub-DAGs can be much more than what the routing loops themselves will cause. This is due to the generation of large number DIOs during the stabilization times resulting from large number of affected nodes.

Another method to repair DODAG locally without causing any DODAG loops has been proposed in [14]. Similar to standard RPL, when a DODAG parent becomes unreachable, a node may switch to another DODAG parent for upward traffic. However, the node first has to transmit a DODAG Repair Request (DRQ) message via link-local multicasting to all nodes and wait for a DODAG Repair Reply (DRP) message.

Further, P2P-RPL [15] could be considered for certain applications such as home automation and building control networks since it provides a reactive mechanism for quick, efficient and root independent route discovery or repair. Data traffic can be avoided from going through a central region around the root and drastically reduces path length. This at the same time substantially decreases unnecessary network congestion around the root as well as delay [6].

B. *Mobility*

Frequent topology changes due to mobility puts another routing challenge. Mobility in the IoT platform refers to the data producer mobility (due to the location change), data consumer mobility, network mobility, and disconnection between the data source and destination pair. Hence, the IoT mobility support should be capable to deliver data with respect to the acceptable delay constraint considering the above cases [16].

Mobility of IoT devices such as mobile gadgets or physical objects (living or non-living) requires sufficient Mobility Management Schemes for data transmission. Host based mobility protocols; MIPv6 [17] and its extensions (e.g. HMIPv6 [18] and FMIPv6 [19]) are not suitable for resource constrained devices. Since the Mobile Node/Mobile Host moving from one network to another is involved in all signalling related process, resource restricted devices such as sensor nodes may experience more complexity and consume more power. The focus of the research is to reduce signalling cost, packet loss and particularly handover latency. Handover latency is caused by L2 (channel scanning, authentication and association) and L3 (movement detection, duplicate address detection and registration delay) handoffs. Therefore, Network Based Mobility Management Schemes (such as NEMO [20]) have attracted the researchers due to the potential in satisfying the QoS for real-time applications. A study of network based mobility management schemes, 6LoWPAN mobility and associated challenges can be found in [21] and [22]. Network based mobility protocols are found to be useful for handling 6LoWPAN mobility with limited energy resources, memory and computational power.

To further support mobility in RPL, another work in [23] proposed an enhancement by modifying DIO messages to include the node's mobility status, and the preferred parent selection procedure in favor of fixed nodes. These modifications are supported by a dynamic DIS management procedure to provide a quick update of the DODAG information.

IV. OTHER DELAY HANDLING STRATEGIES

A. *Service Differentiation and Scheduling*

The most commonly used Best Effort service model is unsuitable for IoT devices. Dealing with delay sensitive traffic, the heterogeneity of IoT devices with limited buffer capacity requires effective buffer management scheme and differentiated service priorities. To meet the QoS requirements, a cost-effective analytical model for a finite capacity queuing system has been proposed in [24]. The proposed model employs pre-emptive resume service priority and push-out buffer management scheme. The arriving traffic is classified into low priority (normal traffic) and high priority (emergency traffic). In case of a full buffer, the lower priority class traffic will be pushed-out by the highest priority class traffic packet in order to avoid data loss of the delay sensitive traffic. Service scheduling follows the priority discipline whereby higher priority traffic is serviced according to First Come First Served (FCFS) prior to lower priority traffic. In other words, lower priority traffic is served only in the absence of higher priority traffic and is immediately pre-empted upon arrival of emergency data packets. Such similar approach can also be seen in [25]. Another work in [26] also proposed a QoS message scheduling algorithm with the idea of service differentiation to differentiate emergency messages from the non-mission critical messages. This is done through classification of messages into high priority (HP) and Best Effort (BE).

On the other hand, research work in [27] highlighted IEEE 802.15.4e Deterministic and Synchronous Multichannel Extension mode, considering IoT devices that send periodic and aperiodic data. Mechanisms that consider a tradeoff between diverse needs of periodic and aperiodic IoT applications as well as maintaining QoS requirements of delay-sensitive have been proposed. A multi-superframe structure can be potentially assigned to a routing device, supporting either Contention Free Period (CFP) or Contention Access Period (CAP) dynamically. IoT devices with periodic data to be sent may use CFP, while devices with aperiodic data may use CAP. The compensation factor carried in MAC packets will be used by the intermediate nodes to perform dynamic management of resources.

Another paper, [28], also proposed a model considering both traffic prioritization and service differentiation in order to provide low bound delay for real-time traffic, while at the same time maintaining a lower packet dropped for delay-tolerant traffic.

B. Sensor/Node Deployment

In [29], the effect of the sensor geometric configuration for the query-based WSN applications has been presented by considering a few real-time applications, namely; volcanic surveillance, patient monitoring, and vitals monitoring and analysis and alarms. Comparing random and two plan-based sensor placement (uniform and star distribution), it is claimed that uniform distribution results in a better round-trip delay due to the construction of the shortest routes between the source and the route.

C. Relay Node Placement

Apart from having a suitable sensor deployment, a strongly connected network topology should be maintained at all times. This is to avoid network partitioning which affects the connectivity goal [30]. In case of node failure, a subset of the actor nodes may be autonomously repositioned as a recovery method to restore connectivity. However, the decision should be carefully handled since it might impose high node relocation overhead or extend some of the inter-actor data paths. A Least-Disruptive topology Repair (LeDiR) in [31] proposed an algorithm of careful node repositioning to recover from a single node failure at a time. The work in [32] and [33] highlighted the issue of relay node locations to improve delivery ratio and end-to-end delay as well as connectivity.

A more detail survey and thorough analysis of network topology management techniques for tolerating/handling node failures in WSNs can be found in [34]. The existing techniques are classified into two broad categories; reactive and proactive methods. Proactive methods are performed before the failure takes place, whereas reactive methods are performed only when the failure is detected.

D. Caching

Due to resource constraints, avoiding unnecessary transmissions among IoT devices while retrieving and distributing the data is very important to save bandwidth and devices’ battery power. In addition, IoT real-time applications requiring shorter delays can benefit from local caches to reduce end-to-end delay [35].

Survey in [2] highlighted that the time constraint is very delicate in real-time sensor systems especially for long-running and instant queries. Unfortunately, the techniques previously used in traditional databases are difficult to apply due to limited resources of the sensors. Thus, caching can greatly reduce the content access latencies. However, caching policy need a careful implementation since data generated also subjected to privacy and security guidelines. While struggling for the efficient resolution of cached copies, certain content should not be cached anywhere. A balance between caching, content security/privacy and regulations should be achieved [16]. Instead of session-based security mechanisms such as TLS/DTLS prevalent in the traditional Internet, the concept of Object Security should take place [35].

V. CONCLUSION

This paper highlights delay contributing factors in IoT and relevant research efforts to minimize it. Focusing on RPL, delay can be contributed by source routing in Non-Storing mode, and loop repair process. A hybrid network of Storing and Non-Storing nodes can be a better solution to tackle the delay caused by source routing as well as limited memory capability. However, the routing pathology should be handled wisely to ensure a smooth interoperability. Regarding the delay caused by loop repair, if handled properly, loop avoidance could enhance the performance of RPL in terms of end-to-end delay, memory consumption for packet buffering, and routing overhead. However, the signaling cost should be carefully taken into account. Similarly, P2P-RPL and the consideration of delay in RPL parent selection could also be explored in reducing the delay. Moreover, mobility of nodes is another delay contributing factor. However, the integration of mobility aspect in RPL is still an ongoing research. Apart from the RPL mechanism itself, some strategies have been identified to potentially reduce delay such as service differentiation and scheduling, sensor deployment strategy, relay node placement and caching. It is hoped that the highlighted aspects in this paper would be beneficial for further research in providing a more reliable and practical environment for common IoT applications, especially the real-time ones.

REFERENCES

[1] Ko, J., Jeong, J., Park, J., Jun, J., Kim, N., Gnawali, O. , "RPL Routing Pathology In a Network With a Mix of Nodes Operating in Storing and Non-Storing Modes," *draft-ko-roll-mix-network-pathology-04*, pp. 1–8, 2014.

[2] "Real-time data management on wireless sensor networks: A survey," *Journal of Network and Computer Applications*, pp. 1013 – 1021, 2012.

[3] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., Alexander, R. , "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," *RFC* . vol. 6550, pp. 1 – 157, 2012.

[4] Kaufmann, M., Interconnecting Smart Objects with IP. DOI: 10.1016/B978-0-12-375165-2.00025-9 (2010)

[5] Kim, E., Kaspar, D., Vasseur, JP. , "RFC 6568 - Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)," pp. 1 – 28, 2012.

[6] Brandt, A., Baccelli, E., Cragie, R., van der Stok, P. , "Applicability Statement: The use of the RPL Protocol Suite in Home Automation and Building Control," *draft-ietf-roll-applicability-home-building-10*, pp. 1 – 32, 2015.

[7] Lindsey, S., Raghavendra, C., Sivalingam, K. , "Data Gathering in Sensor Networks using the Energy Delay Metric," *In: IEEE Computer Society. Proceedings of the 15th International Parallel & Distributed Processing Symposium*, pp. 2001 – 2008, 2001.

[8] Thubert, P., Watteyne, T., Struik, R., Richardson, M. , "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e," *draft-ietf-6tisch-architecture-05*, pp. 1 – 40, 2015.

[9] Gan, W., Shi, Z., Zhang, C., Sun, L., Ionescu, D. , "MERPL: A More Memory-efficient Storing Mode in RPL," *In: 19th IEEE International Conference on Networks (ICON)*, pp. 1–5, 2013.

[10] Ko, J., Jeong, J., Park, J., Jun, J., Gnawali, O., Paek, J. , "DualMOP-RPL: Supporting Multiple Modes of Downward Routing in a Single RPL Network. In," *ACM Transactions on Sensor Networks*, vol. 11, no. 2, Article 39, pp. 1–20, 2015.

[11] Pinto, P., Pinto, A., Ricardo, M.: RPL Modifications to Improve the End-to-End Delay Estimation in WSN. pp. 868 – 872.

[12] Clausen, T., Verdiere, A. C., Yi, J., Herberg, U., Igarashi, Y. , "Observations of RPL: IPv6 Routing Protocol for Low Power and Lossy Networks.,"*draft-clausen-lln-rpl-experiences-09*, pp. 1 – 31, 2015.

[13] Khelifi, N., Kammoun, W., Youssef, H. , "Efficiency of the RPL repair mechanisms for Low Power and Lossy Networks," *In: International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 98 – 103, 2014.

[14] Guo, J., Orlik, P., Bhatti, G.: Loop Free DODAG Local Repair. draft-guo-roll-loop-free-dodag-repair-01, pp. 1–19, 2013.

[15] Goyal, M., Baccelli, E., Philipp, M., Brandt, A., Martocci, J.: RFC 6997 - Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks, pp. 1 – 40, 2013.

[16] Zhang, Y., Raychaduri, D., Ravindran, R., Wang, G.: ICN based Architecture for IoT-draft-zhang-iot-icn-architecture-01, pp. 1 – 23, 2014.

[17] Perkins, C., Johnson, D., Arkko, J.: RFC 6275 Mobility Support in IPv6. July 2011.

[18] Soliman, H., Castelluccia, C., El Maliki, K., Bellier, L.: RFC 4140 Hierarchical Mobile IPv6 Mobility Management, August 2005.

[19] Koodli, R.: RFC 4068 Fast Handovers for Mobile IPv6, July 2005.

[20] Devarapalli, V., Wakikawa, R., Petrescu, A., Thubert, P.: RFC 3963. Network Mobility (NEMO) Basic Support Protocol, January 2005.

[21] Khan, R. A., Mir, A. H. , "A Review of Network Based Mobility Management Schemes, WSN Mobility in 6LoWPAN Domain and Open Challenges," *In: International Journal of Future Generation Communication and Networking*, vol. 7, no. 5. pp. 85 – 104, 2014.

[22] Bouaziz, M., Rachedi, A., *A survey on mobility management protocols in Wireless Sensor Networks based on 6LoWPAN technology*, pp. 1 – 13, 2014.

[23] Korbi, I. E., Brahim, M. B., Adjih, C., Saidane, L.A., *Mobility Enhanced RPL for Wireless Sensor Networks*, pp. 1 – 8, 2010.

[24] Awan, I., Younas, M. , "TowardsQoS in Internet of Things for Delay Sensitive," *In: MobiWIS 2013 Workshops*, CCIS 183. pp. 86 – 94, 2013.

[25] Li, W., Xu, J., Jiang, H., "Queuing States Analysis on a Hybrid Scheduling Strategies for Heterogeneous Traffics in IOT," *In: International Conference on Computer Science and Service System*, pp. 1007 – 1008, 2012.

[26] Abdullah, S., Yang, K., *A QoS Aware Message Scheduling Algorithm in Internet of Things Environment*, pp. 175 – 180, 2013.

[27] Taneja, M., "A Framework to Support Real-Time Applications over IEEE802.15.4 DSME," *In: IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP),* 2015.

[28] Ancillotti, E. Bruno, R., Conti, M. , "Reliable Data Delivery With the IETF Routing Protocol for Low-Power and Lossy Networks," *In: IEEE Transactions on Industrial Informatics*, vol. 10, no. 3, pp. 1864 – 1877, 2014.

[29] Abbasy, M. B., Barrantes, G., Marin, G., *Time Delay Performance Analysis of Sensor Allocation Strategies on a WSN*, pp. 135 – 140.

[30] Rault, T., Bouabdallah, A., Challal, Y. , "Energy efficiency in wireless sensor networks : A top-down survey," *In: Computer Networks*, vol. 67, pp. 104 – 122, 2014.

[31] Abbasi, A. A., Younis, M. F., Baroudi, U. , "A. Recovering From a Node Failure in Wireless Sensor-Actor Networks with Minimal Topology Changes," *In: IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 256 – 271, 2013.

[32] Caro, G. A. D., Flushing, E. F.: *Optimal relay node placement for throughput enhancement in wireless sensor networks*, pp.1 – 6, 2011.

[33] Nigam, A., Agarwal, Y. K. , "Optimal relay node placement in delay constrained wireless sensor network design," *In: European Journal of Operational Research*, pp. 220 – 233, 2014.

[34] Younis, M., Senturk, I.F., Akkaye, K., Lee, S., Senel, F. , "Topology management techniques for tolerating node failures in wireless sensor networks: A survey," *In: Computer Networks*, vol. 58, pp. 254 – 283, 2014.

[35] Lindgren, A., Abdesslem, F.B., Ahlgren, B., Schelen, O., Malik, , "A.: Applicability and Tradeoffs of Information-Centric Networking for Efficient IoT," *In: draft-lindgren-icnrg-efficientiot-01*, pp. 1 – 19, 2014.