Network-State-Aware Quality of Service Provisioning for the Internet of Things

Shafique Ahmad Chaudhry
Department of Computer Science
Dhofar University, Salalah, Oman

Jun Zhang
Dept. of Computer Science
UCC, Cork, Ireland

***Abstract*—The Internet of Things (IoT) describes a diverse range of technologies to enable a diverse range of applications using diverse platforms for communication. IP-enabled Wireless Sensor Networks (6LoWPAN) are an integral part of IoT realization because of their huge potential for sensing and communication. The provision of Quality of Service (QoS) requirements in IoT is a challenging task because of device heterogeneity in terms of bandwidth, computing and communication capabilities of the diverse set of IoT nodes and networks. The sensor nodes in IoT, e.g., 6LoWPAN, exhibit low battery power, limited bandwidth and extremely constrained computing power. Additionally, these IP-based sensor networks are inherently dynamic in nature due to node failures and mobility. Introduction of modern delay-sensitive applications for such networks has made the QoS provisioning task even harder. In this paper, we present Network-State-Adaptive QoS provision algorithm for 6LoWPAN, which adapts with the changing network state to ensure that QoS requirements are met even with the dynamic network states. It is a policy-based mechanism, which collaborates with the underlying routing protocol to satisfy the QoS requirements specified in the high level policies. It is simple in its implementation yet minimizes the degradation of the best effort traffic at a considerable level. Our implementation results show that our protocol adjusts well in dynamic 6LoWPAN environment where multiple services are competing for available limited resources.**

Keywords*—*Internet of things; QoS provisioning; 6Lowpan; Policy-based QoS; IP-based Wireless Sensor Network; 6LoWPAN

I. INTRODUCTION

Wireless sensor networks (WSNs) consist of tiny sensor nodes with highly constrained computation, bandwidth and energy resources. WSN are integral part of IoT applications where sensing is necessary to observe the physical world. In most application scenarios, a large number of sensor nodes are deployed in the region of interest to collect and forward the data to a sink or data processing center. The data collection can be periodic, event-driven or query-based [1]. Seamless connectivity of WSNs with traditional IP networks is an essential requirement for realization of the IoT. Transmission of IPv6 over IEEE standard 802.15.4 (LoWPAN) [2] has given rise to the 6LoWPAN standard [3], which defines encapsulation and header compression for transmission of IPv6 over IEEE 802.15.4 networks. The standard enables us to connect these sensor networks with each other and with other IP-networks to maximize the utilization of information which is mainly associated with IP networks. This integration is pivotal to allow users to access the services in sensor networks as well as in the IP networks.

Though the existence of 6LoWPANs brings in great convenience for the users, it also presents novel underlying requirements and new technical challenges for the IoT research community. The main challenge is the dual nature of these networks, i.e., they are sensor networks as well as IP networks at the same time. Therefore, neither traditional IP-based QoS provisioning nor Wireless Sensor techniques are directly applicable to these networks. 6LoWPAN inherently posses dynamic topologies, limited processing power, small memory, and bandwidth constrained wavering capacity links. The introduction of modern real-time services concomitant to delay-agnostic services has further enhanced the complexity of the problem. 6LoWPAN nodes have limited resources to share between competing services, whereas the services require and expect high priority on the network with an objective of meeting high user expectations regarding reliability and quality. The available QoS provisioning algorithms are either inefficient or are embedded into routing protocols adding a high computation and communication load.

Lots of work has been done in QoS provisioning for sensor networks, e.g., [4] [5] as well as for IP networks [6][7]. However, the IP-enabled WSN domain is mostly an unexplored area.

Traditionally, to fulfill the QoS requirements, network managers attempt Hard-QoS or Soft-QoS mechanisms. In Hard-QoS provisioning, managers negotiate, reserve and hard-set capacity for various types of services (hard QoS). On the other hand, in the Soft-QoS case they merely prioritize data without reserving any “capacity setting”. Hard QoS provisioning is not possible in the 6LoWPAN network because of their ad-hoc nature. Other IP-based frameworks like IP DiffServ and IntServ are extremely ‘heavy’ to be deployed directly on WSNs. Service differentiation could be a viable solution but it requires major adaptation to function reliably in the highly dynamic topologies.

In this paper, we propose a novel QoS provisioning mechanism for 6LoWPAN that provides soft QoS and adapts well with the changing network conditions. The solution can be used to implement user-defined high-level policies for services and is independent of the routing protocols. It can be integrated easily with the underlying routing protocols with minimal modifications. Our preliminary results show that our solution mitigates the degradation of best effort traffic and provides minimum QoS guarantees for applications.

The rest of the paper is organized as follows. In section II we outline the QoS provisioning requirements for IoT and

6LoWPAN. The details of our solution architecture are provided in section III followed by network state adaptive algorithm in IV. We present implementation details and preliminary evaluation results in sections V and VI respectively. A brief review of the state of the art is given in VII and paper is concluded in section VIII.

II. QOS CHALLENGES FOR THE IOT

Quality of service (QoS) provisioning is one of the key issues in 6LoWPAN networks' utility and penetration in the IoT market. Traditionally QoS refers to the measure of service quality from the network to the application. QoS is generally measured in terms of related parameters like available bandwidth, throughput, jitter, availability, delay, error rate, etc. In the case of traditional WSNs, application specific aspects such as data accuracy, aggregation delay, coverage, and network life are also considered as measures of QoS.

While QoS provisioning is an established domain in traditional networks, it still remains an open field when it comes to the IoT. In any IoT scenario, where WSNs and IP networks are integrated, a number of contrasting challenges have to be addressed to meet QoS requirements. While traditional networks mainly focus on end-to-end QoS provisioning, it is generally not the case with WSNs. WSNs are traditionally application-centric and therefore QoS parameters are determined not only by the network but also by application's requirements. The application may demand specific set of parameters, for example, data aggregation latency, accuracy of information, network life, coverage, minimum number of active sensors, fault tolerance threshold, etc. The network QoS provisioning is also influenced by the data delivery model being used. Event driven, query driven and continuous data delivery models affect the fault and delay tolerance, redundancy, user-interaction, and other QoS thresholds for sensor networks.

QoS provisioning in 6LoWPANs is different from just WSNs as well as just from IP networks because of the dual nature of these networks. On one hand, 6LoWPANs are IPv6 networks; while on the other hand, these are low power sensor networks with extremely limited resources. It means that we must provide lightweight IP-like solutions that can be deployed on resource constrained devices. Moreover, traditional networks run a diversity of applications as compared to WSNs where the network traditionally executes a single application in a cooperative fashion. However, because of the IP support, 6LoWPANs must support a variety of services as suggested in [8]. Running multiple applications on the highly resourced constrained devices makes QoS provisioning operations even more challenging. Followings are some of the challenges that must be addressed in order to provide a better QoS provisioning solution for 6LoWPAN.

Platform heterogeneity: In an environment where various type of networks are operating together, it is required that a QoS protocol works seamlessly on heterogeneous nodes as well as networks. Tackling such heterogeneity is one of the biggest challenges in realization of IoT. To cope up with the routing protocol heterogeneity, it is essential that the QoS provision mechanism is not embedded in a specific network protocol. Additionally it is important to have a mechanism, which adapts according to the application and network needs.

Dynamic network topology: Node failure and mobility are common phenomena in pervasive environments. Node failure has rather been considered as a 'normal event' for WSN as compared to IP networks where node failure is a rare event. The network topology in a wireless sensor network can change under various scenarios, e.g., a node or part of wireless sensor network moves out or die, joining of a new node, or some node(s) turn(s) to sleep mode to save energy. Such changes in topology badly affect the QoS provisioning. The QoS provisioning mechanism for IoT, therefore, should help network to discover new QoS-guaranteed route after topology changes.

Multiple Applications Support: In contrast to traditional WSNs where only one application is supported by the network, IP-WSNs may support multiple applications. These diverse applications may need to share the available network resources with different QoS requirements. While some applications generate periodic data, others could be driven by specified events. For instance, an application for event detection and target tracking needs real-time information from sensors. Therefore, the WSN must meet the delay and accuracy requirement during packet transmission. Furthermore, sensors for different kinds of physical variables, e.g., temperature, humidity, location, and speed, generate traffic flows with different characteristics which may need to be handled differently.

Resource constraints: WSN nodes are highly constrained in terms of battery, bandwidth, storage and communication capabilities. Specifically, efficient use of available energy conservation is critically important. It is therefore essential for a QoS mechanism to produce as less traffic overhead as possible in order to save energy and extend the network life.

Multiple Sinks: Traditionally WSNs are characterized with only one sink node but multiple gateways and sinks are also possible, especially where multiple applications are running on the network. In such cases QoS provisioning mechanism should be able to provide diversified parameters of service levels to support requirements from different sinks.

III. SOULTION ARCHITECTURE

Traditionally QoS support is application specific and is provided using the model in Figure 1.

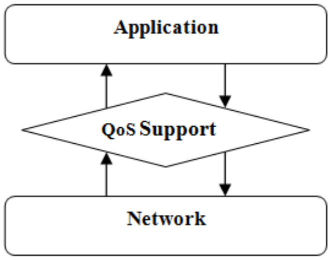


Fig. 1. A simple QoS model

Our solution, lowNETSAQ, is a lightweight policy-based framework which is loosely coupled with the underlying routing protocol and can work with any routing protocol with minimum tailoring. We have incorporated Policy-Based Management framework in our solution in order to allow user-defined configurations with respect to several applications, which may differ in network service requirements. While policy based network management has been in practice for decades, use of policies for QoS provisioning had gained special interest of researchers in recent years. High-level policies can be defined by the administrator to create a metric using path length, link quality, bandwidth, throughput and delay from source to destination node. The QoS is provided for diverse applications according the defined policies.

The mechanism operates between the underlying routing protocol and the application layer. QoS is separated from routing protocols to provide independence and flexibility in QoS mechanism. The QoS Protocol Stack is shown in Figure 2.

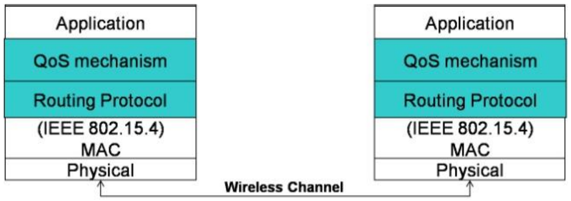


Fig. 2. QoS Implementation Stack

The QoS provisioning mechanism is lightweight and independent of routing protocol. But it can be integrated with most of the routing protocols. For obtaining the link information, the original routing table is customized. The mechanism incorporates additional metrics to the routing table data of the routing protocol. For instance, Figure 3 shows addition of two extra fields to routing table which hold information of QoS routing mechanism.

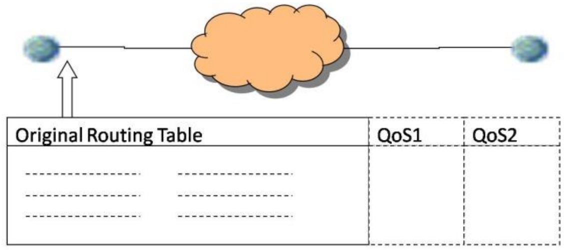


Fig. 3. Extra fields in routing table

The big picture of the architecture is given in Figure 4 and shows main components of the lowNETSAQ framework.

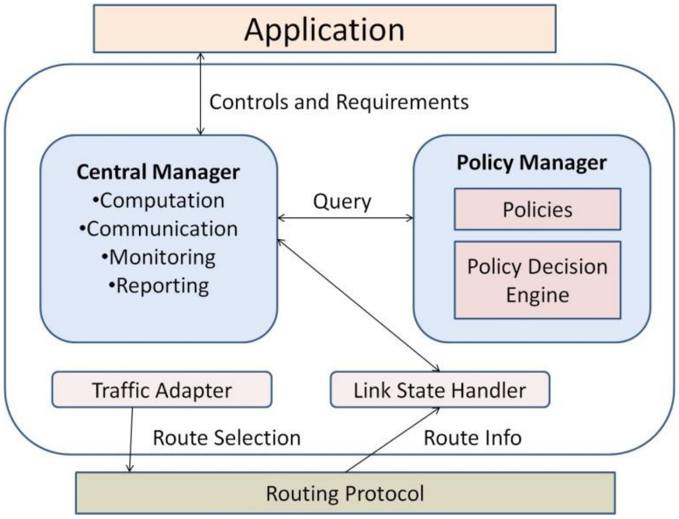


Fig. 4. Architecture of framework

A. Policy Manager

The policy manager can be used by the administrator to establish list of policies based on application requirements. A policy can be defined as a set of rules and decision strategy to be applied under specific circumstances to achieve a goal. These policies can be created using different parameters, such as, path length, link quality, bandwidth, throughput and delay from source to destination node, to provide diverse metrics for QoS provisioning. Policy-based approach enables the system to use the passive network monitoring information for active QoS provisioning. Policy decision engine is used to monitor the configurations using the central manager for deployment of the policy. The details of policy manager and policy decision engine design and implementation are subject of another paper and beyond the scope of this paper.

B. Central Manager

The central manager, in collaboration with other components, is responsible for monitoring the network state. In wireless sensor network, the network state is dynamic and unpredictable because of node mobility, node failure and battery exhaustion. A node of part of wireless sensor network moves out or even die, new node can join into the network or a node turn to sleep mode to save energy. All those situations make the network topology vary and the QoS mechanism must monitor these changes to provide an adaptable. Central manager corresponds with other components to perform such functions as monitoring link state, configuration of packet transmission metrics and reporting feedback to application.

C. Link State Handler

Link state handler gathers the route information of the network from the underlying routing protocol through routing table. This information is used by the QoS manager in route selection for a packet flow. As WSN state changes so dynamically, the solution must be able to adapt as network state change. lowNETSAQ makes sure that minimum QoS guarantees are met even when network state changes due to mobility, node failure or external interference.

D. Traffic Adapter

Traffic adapter is the core component of the system. It runs network state adaptive algorithm, which in collaboration with central engine makes route selection decisions. As WSN state changes so dynamically, the QoS provisioning mechanism needs to adapt with the network state changes. In order to meet end-to-end QoS requirements, the network will adapt to topology dynamically to guarantee minimum QoS requirements for either real-time or non-real-time applications.

IV. Network State Adaptive Algorithm

In a network, that provides services to multiple user applications, some applications may spell their QoS. We assume that, any application that starts has certain QoS requirements with minimum and maximum QoS bandwidth constraints. Our mechanism finds and uses the routing path which can meet the requested QoS specifications.

At the start of any application which specifies its requirements, the originator sends out route-discover message. When the metric values, bandwidth and delay are successfully obtained, route-discover-reply is sent to the originator.

Let's assume the bandwidth required by application is *Br*. For a given path *B1, B2, B3, …,Bn,* the maximum bandwidth available on the path is the minimum bandwidth available on a link on the path. It mean

$Bmax\ =\ min\,\{B1, B2, B3\,\ldots, Bn\}\ \ldots$ (1)

The application's QoS requirements are considered met if $Br\ <=\ Bmax$. otherwise an alternative path is to be found. During the path discovering and sending discover packet, the protocol updates the bandwidth field called *Bmax* in the routing table. Let *i* and *j* to be two positive integers (i.e. indexes of nodes) and i $>$ j. When packet is received at each intermediate node, field *Bmax* is updated to *Bi* if condition *Bi* $<$ *Bj*. Otherwise do nothing with the routing table.

Similarly, assume the delay required by application is *Dr*. The total delay that occurs of the path with *n* links/hops can be given by

$$Dtotal\ =\ D1\ +\ D2\ +\ D3\ +\cdots+\ Dn$$

Dtotal = $\sum_{i=1}^{n} D_i$ ▯

Then we assert that

If *Dr* >= *Dtotal*, the path meets QoS requirement.

If *Dr* $<$ *Dtotal*, then find another path.

During the path discovery phase, the protocol computes the sum of delay experienced at each node. It accumulates the *Dtotal* field of routing table when discover packet passes through each node. Let *i* to be the index of a node.

When route discovery packet reaches at each intermediate node, computing *Dtotal = Di + Dtotal'*. Where *Di* is delay experience on the current node, and *Dtotal'* is value of *Dtotal* field in routing table when packet is received on current node (i.e. *Dtotal* computed on last node).

For a single node in the network, delay is dependent on queue length and can be modeled as M/M/1 queue.

$$Q = \frac{1}{\mu - \lambda} \qquad (2)$$

Where μ is processing rate and λ packet arrival rate. Processing rate μ is assumed as constant. In general, λ is equal to the sum of the average message flow rates of all paths traversing this node

$$\lambda = \sum_i \sum_j \gamma_{ij}$$

Here γij is average message flow from node *i* to node *j*. To simplify the calculation, we assume all paths have same message flow γ. Let *m* to be the number of paths via a node. Then average arrival rate for a node is given by:

$$\lambda = \sum_m \gamma = m \cdot \gamma$$

To determine this we need gather information from the routing table or query on each node (how many source node pass this node corresponding to same number of path traverse it).

Equation (2) provides delay on a single node base on M/M/1 queue theory, and we can deduce the delay of a path.

$$D = \sum_{i \in path} Q = \sum_{i \in path} \frac{1}{\mu - \lambda}$$

After substitution we achieve

$$D = \sum_{i \in path} \frac{1}{\mu - \sum_i \sum_j \gamma_{ij}} \qquad (3)$$

For finding out the route to meet the requirements in terms of *Bmax* and *Dtotal* given in equations (1) and (3) respectively, routing data and routing table information is used. Two additional columns, Band_width_for_route (*Br)* and Delay_for_route (*Dr)* are added to the routing table in 6LoWPAN. These columns are used for route selection based on requirements. To create and update these columns, Route Reply (RREP) message has been modified to get the link information. When a node sends out a RREP back to the originator, it adds these extra fields to the message. Figure 5 illustrates the flow of the protocol.

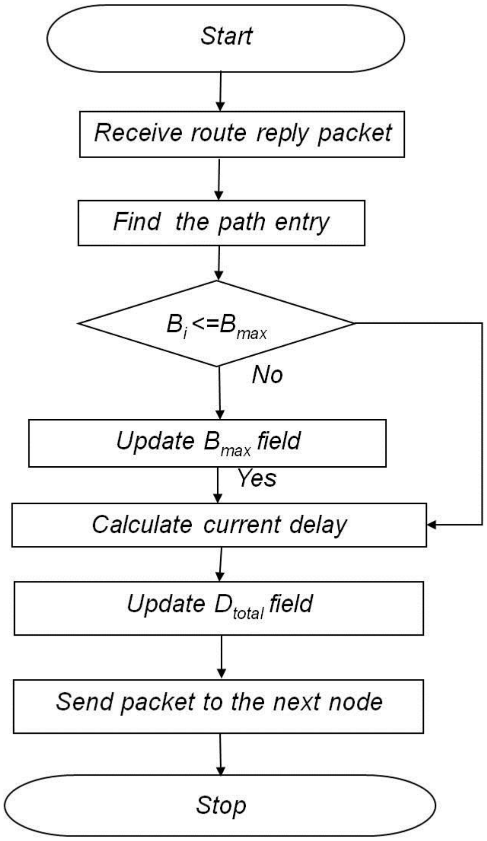
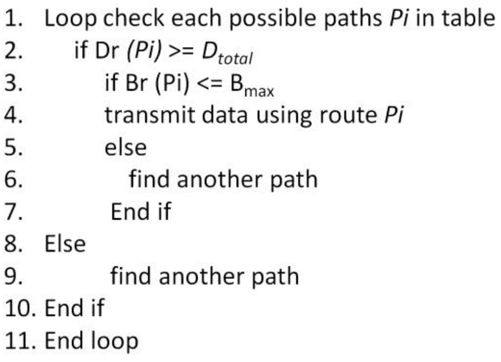


Fig. 5. Link State Updating Process

Every time an application specific data is to be sent, the objective is to find a path that meets QoS requirements of the application. Figure 6 outlines an algorithm for QoS provisioning.



```
Loop check each possible paths $Pi$ in table
    if Dr $(Pi)$ >= $D_{total}$
        if Br (Pi) <= $B_{max}$
        transmit data using route $Pi$
        else
            find another path
        End if
Else
        find another path
End if
End loop
```

Fig. 6. Route Selection Algorithm

After examining all feasible paths if no path meets QoS requirements, then best effort delay and bandwidths are provided.

V. Implementation

We have implemented lowNETSAQ over Berkley 6LoWPAN stack [9] running on Advantics XM-1000 sensor nodes. The implementation consists of several modules based on the NesC and TinyOS frameworks. These modules cooperate with b6lowpan to provide routing with QoS support. Each component is lightweight and loosely coupled with others. Figure 7 shows the components and the interfaces between them.

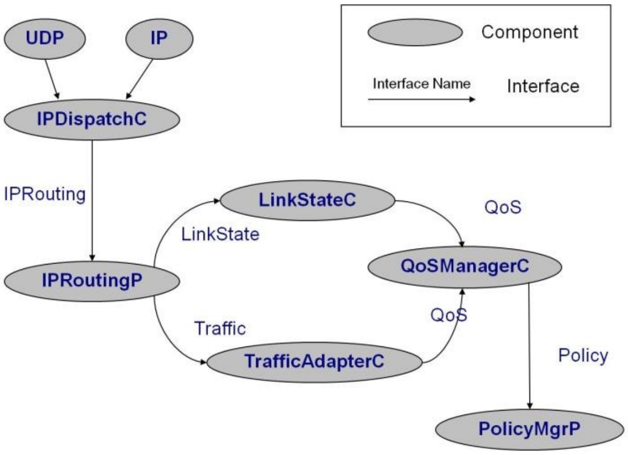


Fig. 7. Component Diagram

TrafficAdapterC:- A component for controlling route selection has been developed. This component contains TrafficAdapterP module which provides interface Traffic and uses interface QoS .TrafficAdapterP that cooperates with IPRoutingP and QoSMananger C. When sending a packet, TrafficAdapterP retrieves each route in routing table and queries QoSManangerC to check if it meets the QoS requirements. It returns the valid route to IPRoutingP after found a candidate route. LinkStateC component collects the network state information for QoS manager. The LinkStateP module has been implemented, which provides interface LinkState and uses infterfaces QoS, IP and IPAddress (see figure 8). The major job of this module is to maintain the network (links) state on each individual node. It calls QoS manager to calculate current network state via interface QoS. The calculation is based on the metrics collected from IP layer such as buffer size, number of neighbour and so on. When a route-discover message is received on a node, IPDispatchC invoke an event IP.recvfrom in the LinkStateP module. It decides to send the message back to origination node if message is not for this node, or update the link state information.

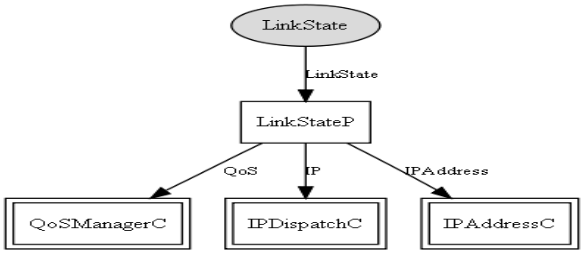


Fig. 8. Link State component

The QoSManagerC component is central to the mechanism. It does important computations and coordinates with other component. For the purpose of adopting QoS support for routing protocol, some modification has been made in IPRoutingC and IPDispatchC components of blip. The IPRoutingC component provides routing function. For decide an ideal route it needs to ask QoS manage to select route via TrafficC when sending packet at each time. It also implements a mechanism to send route discovery message periodically to gather network state information, and update routing table when receive route discovery message. This mechanism helps to maintain the link state information used for QoS manager. Another component that needs to be modified is IPDispatchC which manage send/receive packet, packet fragmentation and reassembly. Additional it need to capable to pass link information to LinkStateC component and forward route discovery message. Figure 9 shows the collaboration between each component.

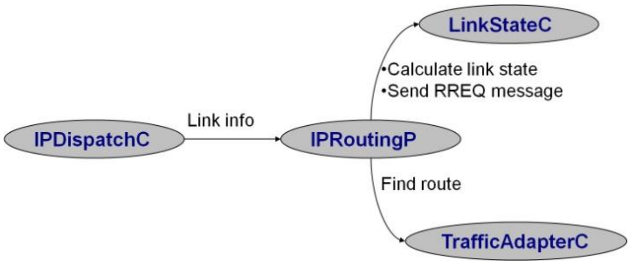


Fig. 9. Interaction of routing component

For the purpose of testing blip network and QoS provisioning mechanism, two applications have been developed in blip project which are UDPEcho and IPBaseStation. We have enhanced the code to support QoS provisioning.

VI. Evaluation

Our preliminary testbed comprised of 25 Advantics XM-1000 nodes with the transmission speed of 250kbps. The XM1000 is the new generation of mote modules, based on "TelosB" technical specifications, with upgraded 116Kb-EEPROM and 8Kb-RAM and integrated Temperature, Humidity and Light sensors. For examining the performance of QoS mechanism, the originator node sends message to base station on different data send rate. The larger data sending rate is selected in order to make congestion on the traffic. Sometimes turn off or move out of range the intermediate node to test the QoS mechanism to reroute the transmission due to dynamic network topology.

The evaluation results are obtained by comparing performance of best-effort traffic and QoS traffic (traffic with QoS support). Average delays, Average Throughput, Packet loss are the metrics used for evaluation.

Average delay is defined as the time escaped between transmitting a packet from originator and successfully receiving it at destination node. Figure 10 shows the average delay of both traffic increases sharply because of the contention of higher data rate. When data rate is 1 packet/s, there is no congestion in network, so the average delays of two traffics are similar. But after that the QoS traffic suffers less delay than best-effort traffic, because it redirect to another route when detecting the congestion or link quality decrease because of node mobility.

When sending rate is 8 packet/s, the delay of QoS traffic is prone to close to the value of best-effort traffic. This can be explained by the congestion that often occurs when send rate is large. As close to the bottleneck of traffic, the delay is keep in a static state that is almost equal to the time-out value.

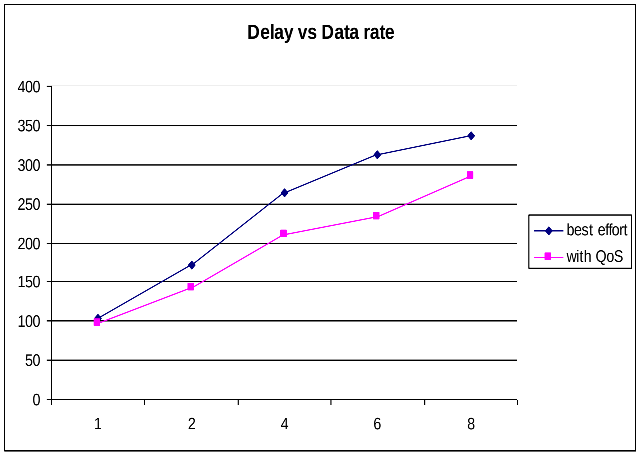


Fig. 10. Average delay vs. Data send rate

Fig. 11 presents the packet delivery ratio under different data sending rates. The experiment result shows the packet delivery probability is slightly improved by adopt the proposed QoS provisioning mechanism. But the network topology of test bed is small, and there are only two hops on each route. The packet delivery probability would be worse if there are more hops on the route, and the QoS traffic performs better than best-effort by choosing to use a QoS-guaranteed alternative route to transmit packets.

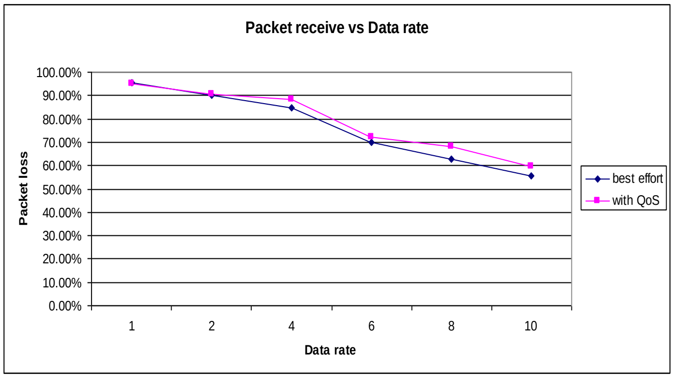


Fig. 11. Packet delivery ratio vs. Data rate

In order to test the max throughput of traffic, the size of packet is increased to 1 KB. This packet size is large enough to fill up the traffic capacity when data rate is faster. The throughput is calculated by counting how many packet is received divide by the time period, which gives the number of packet receive per second on the base station node. In QoS traffic, there is still degrading throughput of the node with larger sending rate. But the QoS provisioning mechanism works as expected, it does improve the performance of network.

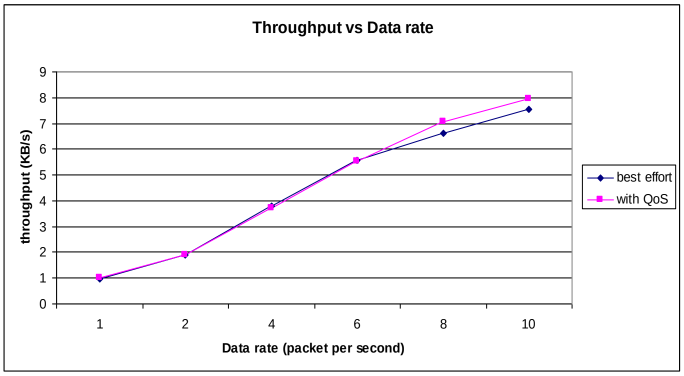


Fig. 12. Average Throughput vs. Data rate

From the comparison the performance in figures 10,11 and 12, it reveals that traffic flow with QoS provisioning mechanism performs better owing to route discovery, contention detecting, rerouting, and high-level policy control on sensor nodes.

VII. Related Work

Quality of service provisioning has been an important topic in recent years and a survey on research efforts in QoS in WSN is presented in [10].

Various efforts are carried out on multi-path approach for QoS provisioning. The work called SPEED [11] proposed a location-based routing protocol for soft end-to-end real-time guarantee for a desired delivery data speed in the network. This proposal does not take into consideration energy metric. An extension of SPEED is the MMSPEED - Multi-Path and Multi-SPEED Routing Protocol [12] which provides reliability and timeliness based on multi-path routing. The multiple paths are chosen depends on the required level of reliability and delivery speed. Another soft-QoS provisioning scheme is presented in [4], which considers multi-constrained QoS in WSN. This scheme utilizes the multiple paths between the sources and sink pairs for QoS provisioning. But as introducing IP infrastructure into WSN, those approaches are not applicable to provide end-to-end QoS provision.

Another effort for QoS provisioning protocol [13] focuses on reliability and energy-aware. This work is based on extension of 6lowpan ad hoc on-demand distance vector routing protocol (LOAD) proposed in [14]. The problem however is that LOAD has not been accepted as a generic routing protocol for 6lowpan. The QoS-aware protocol proposed in [15] implements a priority system that classifies the network traffic into real-time and non-real-time. The protocol is aim to find a least-cost, delay-constrained path for real-time data and maximize the throughput for non-real-time data. In [16] a service polling model base on two queues has been implemented to provide guaranteed QoS in WSN. This model also support for classify network traffic to real-time and non-real-time.

Another routing protocol is proposed in [17]which considers collisions and provides multipath routing to increase the network lifetime and throughput while decreasing latency.

In [18] forward error correction technique is used to provide fault recovery, balance the energy consumption over sensor nodes, and increase the reliability of data transmission.

The work in [19] proposed a QoS routing for time sensitive data delivery. However, this approach provides QoS guarantee at the cost of network life.

A delay guaranteed routing and MC protocol (DEGRAM) [20] proposes a joint-duty cycled MAC and routing protocol, which is based on contention free TDMA. The proposal, however, suffers from the inherent TDMA based MAC synchronization problem.

Source directed multipath routing [21] proposes unification of MAC and routing by using Wyner-Ziv lossy coding on application layer. This work targets hard QoS provisioning by reserved path, hop-by-hop QoS agreements, and admission control mechanisms.

However, those above approaches do not take into account the network state dynamic. The dynamic topology is universal phenomenon in wireless ad-hoc and sensor network. Hence, The QoS provisioning also need to be network state adaptive.

VIII. Conclusion

In this paper we presented a network state adaptive light-weight QoS provisioning mechanism for 6LoWPAN. The solution has been implemented on real test bed and preliminary results have been presented. The solution is loosely coupled with underlying protocol making it flexible to be used with any underlying protocol with minor modifications. The policy-based feature assures the scalability of system. In future works, we plan to test the performance on large test bed. Another important feature is to add more parameters to the QoS metrics.

References

[1] D.Chen and P.K. Varshney, “QoS support in Wireless Sensor Networks: A Survey” International Conference on Wireless Networks, Las Vegas, USA, June 2004.

[2] IEEE LoWPAN, 2003. Standard 802.15.4-2003, https://standards.ieee.org/about/get/802/802.15.html

[3] 6Lowpan working group, http://datatracker.ietf.org/wg/6lowpan/charter

[4] X. Huang and Y. Fang, “Multi-constrained Soft-QoS Provisioning in Wireless Sensor Networks”, 3rd international conference on Quality of service in heterogeneous wired/wireless networks, Ontario, Canada, 2006

[5] E. Felemban, C. Lee and E. Ekici, “MMSPEED: multipath Multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks”, IEEE Transactions on Mobile Computing, Volume 5, Issue 6, pages 738-754, June, 2006

[6] Xiao H., Seah, W.K.G., Lo, A., Chua, K.C.: A Flexible Quality of Service Model for Mobile Ad-hoc Networks. *IEEE VTC*’00, Tokyo Japan, 2000.

[7] Jawhar, I., Wu, J.: Quality of Sevice Routing in Mobile Ad Hoc Networks. *Resource Management in Wireless Networking*, Kluwer, 2003.

[8] Ali Hammad Akbar, Ahmad Ali Iqbal, Ki-Hyung Kim: Binding Multiple Applications on Wireless Sensor Networks. GPC 2006: 250-258

[9] Berkley 6lowpan implantation, 2010. Available from: http://smote.cs.berkeley.edu:8000/tracenv/wiki/blip [Accessed May 10, 2015]

[10] Lin, F., Zhang, D., & Li, W. (2011). Research on quality of service in wireless sensor networks. In Software engineering and service science (ICSESS), 2011 IEEE 2nd international conference on, IEEE, pp. 312–315.

[11] Tian He, John A. Stankovic, Chenyang Lu, and Tarek Abdelzaher, “SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks”, *23rd International Conference on Distributed Computing Systems* (ICDCS '03). IEEE Computer Society, Washington, DC, USA, 2003.

[12] E.Felemban, C. Lee and E. Ekici, “MMSPEED: multipath Multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks” IEEE Transactions on Mobile Computing, vol. 5. No.6. pp: 738-754, 2006.
DOI: 10.1109/TMC.2006.79

[13] C. Weston, P. T. Tu, B. Baiteli and S. Y. Ma, “Energy-aware and QoS Routing in 6LoWPAN Mesh Networks”, ZRDC conference, Zambia, April 30, 2007.

[14] K. Kim, S. D. Park, G. Montenego, S. Yoo and N. Kushalnagar, “6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD)”, IETF Internet Draft, March 2006.

[15] K. Akkaya and M. Younis, “Energy and QoS aware Routing in Wireless Sensor Networks” Cluster Computing Volume 8 , Issue 2-3 (July 2005) Pages: 179 – 188

[16] M. Y Aalsalem, M. Iftikhar, J. Taheri and A. Y. Zomaya, “On the Provisioning of Guaranteed QoS in Wireless Sensor Networks through Limited Service Polling Models”, 2006

[17] Y. Chen, N. Nasser, T. El Salti and H. Zhang, “A multipath QoS routing protocol in wireless sensor networks,” International Journal of Sensor Networks, Vol. 7, Issue. 4, pp: 207–216. 2010. DOI: 10.1504/IJSNET.2010.033204

[18] J. Ben-Othman and B. Yahya, “Energy efficient and QoS based routing protocol for wireless sensor networks”, Journal of Parallel and Distributed Computing, Vol. 70, issue. 8: pp. 849–857.2010. doi:10.1016/j.jpdc.2010.02.010

[19] T. Houngbadji, and S. Pierre, “QoSNET: an integrated QoS network for routing protocols in large scale wireless sensor networks,” Computer Communications, vol. 33, issue 11: pp. 1334–1342. 2010.DOI: 10.1016/j.comcom.2010.03.017

[20] C. Shanti and A. Sahoo, “Dgram: A delay guaranteed routing and mac protocol for wireless sensor networks,” IEEE Transactions on Mobile Computing, Vol. 9 No. 10, 1407–1423.2010.

[21] G. Shah, W. Liang and O. Akan9,“Cross-layer framework for qos support in wireless multimedia sensor networks”, IEEE / ACM Transactions on Multimedia, Vol. 14. No.5, 1442–1455. 2012.