QoS and Congestion Control for Pervasive Event Driven Applications

Poonam Yadav
Department of Computing
Imperial College London
United Kingdom
pooyadav@doc.ic.ac.uk

Julie A. McCann
Department of Computing
Imperial College London
United Kingdom
jamm@doc.ic.ac.uk

ABSTRACT

Pervasive computing applications consist of many components ubiquitously embedded throughout the environment involving complex interactions to provide the networked service to the user as smoothly as possible. This paper focuses on the class of pervasive services that involve some form of sudden bursty traffic. Pervasive monitoring services that involve sensing are a classic example of this type of system whereby a basic telemetry service communicates periodic data - perhaps accumulating or aggregating that data as it traverses the network to its destination. Moving some of the intelligence to the sensing nodes changes the periodic nature of data delivery to a more bursty one as when an event of note arises, that event is communicated and that communication should receive priority. Nevertheless, the vast majority of sensing network protocols has been viewing traffic in a monolithic way and the concepts of priority and Quality of Service (QoS) classifications are a relatively new departure. In this paper we propose a Priority Based Re-Routing (PBR) scheme with rate control and experimentally evaluate the importance of data congestion control in event delivery. We show that our routing schemes not only ensure improved packet delivery performance for event data but do so more reliability over current schemes.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Distributed networks; C.2.2 [**Network Protocols**]: Routing protocols

General Terms

Experimentation, Algorithms

Keywords

Wireless sensor networks, QoS, Congestion control

1. INTRODUCTION

Pervasive services consist of many components ubiquitously embedded in devices and the environments in which we partake. Such services require collaboration, and involve complex interactions between people, intelligent objects and technology. This collaboration manifests as services that can provide anything from social networking, to caring for the elderly to alert monitoring in urban environments. Here success lies in the ability to provide users with cost-effective services that have the potential to run anywhere, anytime and on any device with minimal user distraction.

This paper is focused on the set of pervasive services that constitute a form of monitoring. Applications such as the intelligent home or UbiCare home require sensing devices that are embedded in the environment and are able to update the applications requiring the sensing services on a periodic basis [2]. This we call basic telemetry whereby an environment is being monitored, with day-to-day periodic data being relayed back, via the typically wireless network, to some node (the sink) that is interested. Typically, the sink is the device that is running the application requesting environmental data. This may be mobile or fixed, wireless or wired or even off-site (in a hospital for example).

There has been a move to push some of the intelligence from some central device to the sensing nodes so that when that node has something interesting to say, it creates an event which is sent to the particular sink interested in it. This has come about because of two reasons. Firstly, increased processing capacity of embedded sensing technologies has allowed more distributed processing to be carried out, thus reducing single points of failure. Also, many monitoring applications require sensing devices to be positioned in close proximity to provide sensing coverage which means the network is dense and with this density comes some very peculiar and less predictable radio effects; result in more data packet collisions and the network performance becoming compromised. Add to this the costs of sending data wirelessly in terms of both the network capacity and battery usage, there is a strong argument for decentralization. Furthermore, this improvement in sensing node hardware is also heralding a move toward a dual operation whereby a node is fitted with more than one sensing device which could e.g. carry out basic telemetry (sending temperature regularly) but also be detecting some event (Granny has not moved in 10 minutes). That event may require some urgency to effect a change such as texting a neighbor to check on Granny or even initiating a self-calibration routine that checks that the

motion sensing device is working properly. Another application type that has similar properties is that of environmental monitoring, e.g. water leakage management. Here the pervasive sensing application is monitoring the water flow, turbidity and even the actuator vibration within the water pipe network. It periodically sends off summaries of this data to the base station (sink), but in parallel it also keeps trend data and when it detects an anomaly it begins to generate event data to indicate a possible leak situation. The application will now require more detailed knowledge of the event (not just that it happened). For example on detection of a leakage event the application will require more details about the turbidity and pressure leading up to and during the suspected leakage event to be able to understand what exactly is happening. Therefore the system will now measure more details; the sensing duty cycle will increase. This data is now more plentiful and must be delivered with some urgency; therefore reliability is of paramount for those packets.

In this paper, we introduce and motivate why we need to better understand pervasive sensing network capacity. We then discuss related work. Our solution spans across the network stack so we discuss optimization in section 3 and then congestion control schemes in section 3.3 and 3.4. We evaluate Priority Based Re-routing scheme using simulations in section 4 then analyze its packet loss behavior in presence and absence of B-MAC congestion control scheme.

2. PERVASIVE SENSING AND NETWORK CAPACITY

In routine sensing applications, each sensing node generates a relatively known traffic or packet rate as it is monitoring phenomena periodically. For simplicity, we examine a single hop wireless sensor network (see fig 2-a). Suppose, there are $'n'$ nodes in the network with a single sink with an achievable capacity of C [1]. If we consider that all nodes can detect and send routine messages at same time to the sink, then the maximum achievable capacity assigned to each node is shared over the network and is C/n. However, in real-world scenarios nodes are likely to detect events that occur with less predictably and which may originate from a specific region in the network. Let us assume 10 % of total nodes $'n'$ generate bursty traffic which is five times the *normal* packets/messages per second during the event detection. In this case the minimum required capacity for these event detecting nodes is increased five times the previous case, therefore a new required capacity for each is $\frac{5*C}{n}$. This means that the total increased traffic in the network is 50% more than that observed in the basic example.

Two conditions result in network congestion for single-hop scenarios. First if the number of nodes detecting events are more than expected, this obviously leads to extra traffic; secondly if the event nodes generate packets at higher rates than the predefined rate. Generally, the majority of pervasive sensing application solutions are designed to provide a guaranteed QoS for predictable traffic (routine traffic) but in case of events, further unpredictable data is generated that can bring the message rates beyond the expected capacity of the network leading to congestion. Even when total network traffic has not exceeded the network capacity we can experience congestion due to other reasons such as channel con-

---

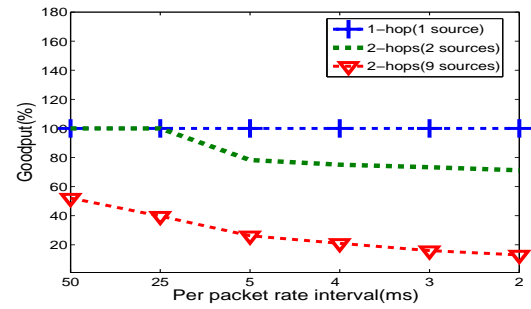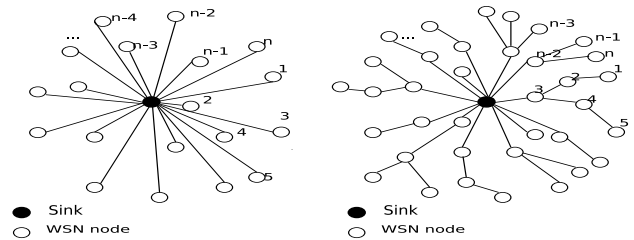[1]Maximum receive rate (pkts/sec) at the sink

Figure 1: Goodput (total data less overhead packets) rate as hop count increases

(a) Single hop   (b) Multi hop

Figure 2: Single-hop and Multi-hop Network showing the Sink node and Sensing (WSN) nodes

tention caused by concurrent transmission[10], buffer overflows and varying wireless channel conditions[16]. Therefore congestion detection and control is important in pervasive sensing applications for resource utilization to avoid degrading the network efficiency as well as throughput; as they not only deteriorate performance but also waste energy.

Figure 1 illustrates the number of packets dropped as the hop count increases. Here we carry out a basic simulation using TOSSIM-2[12] for MICAZ devices [1][18] to show the Goodput rate (total number of data less overhead packets) in a 1-hop network; that is, 2 nodes (a sink and a source). We also measure a 2-hop network (3 nodes, one is sink and two source nodes connected in chain topology), and 10-hop network (one sink and 9 source nodes) respectively. The link quality is assumed to be ideal and external noise is accounted for, hence the 1 hop line is showing that the sink is receiving 100% of the data packets. Here we can clearly see that the Goodput drops considerably; with a 9 hop network barely sending 20% useful packets.

Most current standard congestion control strategies do not differentiate between event and routine traffic and implement the same action for all data.

2.1 Related Work

There are many QoS metrics to determine the quality of a transaction in pervasive sensing applications. These can be in terms of data packet delay, bandwidth efficiency, reliability and power consumption even data accuracy (e.g. sensing precision). A routing algorithm optimizing power consumption during source to destination forwarding using a particle

swarm algorithm is explained in [9]. Energy optimization in pervasive sensing application routing that is based on connexionnist training using self organizing maps, is proposed in [7]. A virtual-Infrastructure based energy efficient framework is designed for routing over dense wireless sensor networks, that considers the load-balancing to enhance the life-time of the network[15]. Other research work proposes ant colony optimization based on a QoS routing algorithm that considers a *delay constraint maximum energy residual ratio* as its optimization objective [5], this algorithm able to achieve the QoS but computationally complex and not suitable for many embedded sensing application hardware. Reinforcement learning routing algorithm is proposed in [14] in which each node maintains information about its optimization objectives(Q's) values for each possible hops. Other QoS metrics, such as network throughput and network lifetime, are optimized in a QoS based multipath routing algorithm [6]. A fuzzy logic based congestion estimation method is proposed in [13] for QoS in pervasive sensing applications, provides the QoS in the absence of all network information or data. A good literature survey on energy efficient routing algorithms can be found at [20], which describes different energy efficient route selection policies. Much research has focused on QoS based routing for event driven applications. The trivial routing solution, random route selection has gained its own importance where simplicity of design algorithm is given paramount importance. [21] propose a sink centric transport protocol. In this protocol, packet loss is detected by the using the discontinuous packet sequence numbers. The sequence numbers of those packets lost are sent back towards their sources. This protocol considers only event packets in the network not the background data packets, therefore assuming each event packet has only one path towards sink; an assumption does not exactly fit in real-time dense wireless sensor networks where the numbers of event nodes can be more than one, and the sink(s) receives different data streams.

An Adaptive Forwarding scheme is proposed in [3], that assumes a number of priority levels n, may be generated by the application and represents the levels of differentiation desired by applications. Each priority level l_i is associated with a reaching probability P_i, and the new forwarding probability F_i is dynamically computed. This approach assumes all priority packets are generated at same rate but allows the event packets to choose routes taking single path, multipath, multi-packet and hybrid forwarding. They show that maximum reliability can be achieved by flooding, But do not consider congested networks in which flooding has known poor performance and high packet-loss. [19] simulated three service differentiation schemes to evaluate the packet delivery ratio and end-to-end delay for high priority packets. Here, a single end-to-end path is reserved for high priority packets. This solution suffers with the major drawback that if some intermediate node fails then all high priority packets will lost. Node failure is considered a *norm* n pervasive sensing applications computing due to the nature of the hardware used (low-cost, battery powered etc.) and the dynamism and harshness of the environments they are typically deployed in.

The work that most resembles what we present here is the event driven geographic routing algorithm (EDR) [11] uses local information to provide transport layer and routing layer QoS support. For QoS support at the transport layer, it suggests the use of two separate buffer queues for routine data packets and event data packets at each node respectively. For QoS support at the routing layer, it uses a cost function to decide on which nodes is the next-hop. Here three parameters are gathered from neighboring nodes: their position relative to base station, their current buffer queue size and their remaining energy [11]. This algorithm requires the knowledge of geographical information either in term of coordinates or angular offset from the direction towards the base-station, which significantly increases the complexity of the algorithm. Furthermore, while selecting the neighbor, this algorithm does not consider link quality, a significant oversight as packet loss may still occur over a congested medium.

3. CROSS LAYER OPTIMIZATION FOR QOS SUPPORT

Priority Based Re-Routing (PBR) protocol that supports congestion control and also provide priority based delivery to event traffic[22]. Typically periodic data makes use of the best path possible over the multi-hop network to send the data from the source to the sink. When an event occurs this means that the event traffic will be in competition to use the same shortest paths therefore in PBR the low priority traffic is essentially redirected or dispersed over a wider area away from the shortest paths and only event traffic is sent down the best route. If congestion occurs due to the rate of high priority packets being received by downstream nodes, then back-off information is cascaded backwards down the route to reduce the number of event packets being generated (as they will only be lost in the network). Each node is told what rate it is allowed to work at so as to minimize collisions at topology initialization time.

To this end, we have explored two different congestion control strategies with our priority based routing algorithm. First is a traffic differentiation scheme and other is an adaptive rate control scheme. The implementation of both these schemes require an efficient Self-Stabilized Adaptive Spanning Tree topology. The spanning tree is created and updated using a broadcast message sent by each node called a Topology update message (TUM). Before we explain the spanning tree formation and each type of congestion control scheme, we show the network and event model that describes the network assumptions and terminologies used in this work.

3.1 Network and Event Model

The wireless sensor network is composed of n nodes of which one of them is the sink node. We assume a radio link model where each node(n_i) has a certain transmission range(r_i) and uses omni-directional antennas. For convenience we use symmetric links, i.e., for any two nodes u and v, u reaches v if, and only if, v reaches u. Thus, we represent the network by the graph of nodes and links as G = (N,L) with the following properties:

N = (n_1 , n_2 , ... , n_n) is the set of sensor nodes, such that $|N|$ = n and n1 is the sink node(S);

E = (e_1 , e_2 , ... , e_m) is the set of event sensor nodes, i.e., nodes detecting an event, such that $|E|$ = m and and M $\subseteq$ N;

$$L = \bigcup_{i,j \in \{0,1,...,n\}} \langle i,j \rangle$$

iff e_i and e_j are neighbors.

The k-hops neighborhood N^k is composed of the node itself and its k-hops neighbors. Thus, the 1-hop neighborhood of the node n_i is given by

$$N_i{}^1 = \{n_i\} \bigcup \left(\bigcup_{\langle i,j \rangle \in L} \right) \{n_j\}$$

The parent of node n_i is given by

$$Parent(i) = nl : \langle l, i \rangle = Best(\forall \langle j, i \rangle); \langle j, i \rangle \in N_i{}^1$$

Here, Best() returns the best 1-hop neighbor satisfying the predefined parent constraint. The children(i) of node n_i is given by

$$Children(i) = \bigcup_{\langle i,j \rangle \in N_i{}^1} n_j : Parent(j) == n_i$$

3.2 Self-Stabilized Adaptive Spanning Tree

Fixed Sink initiated multi-hop routing is very popular in pervasive sensing networks [8] [17] [22]. Here the sink initiates a Breadth-First Search to form a spanning tree of routes from the nodes to the sink. Figure 2-b shows a multi-hop spanning tree. Algorithm 1 shows the formation of this tree using synchronous topology formation(for simplicity as the receive is always able to receive management packets) the spanning tree is formed periodically. Each node in the spanning tree manages the information required for topology management, congestion control, rate control and priority based re-routing. Here we set up the parameters in the initialization round and again in subsequent rounds updating parameter values to adapt to network changes (such as a hole being caused by the death of a node on the route). In wireless networks the Spanning tree is formed by exchanging Topology Update Messages (TUM) locally with neighborhood nodes. Each node in the spanning tree require the parent information as well as children information. The information update phase is completed in two sub-phases called top-down phase and bottom-up phase. Parent information is propagating from root to end nodes (top-down using TUM messages) whereas information from child nodes propagate in the reverse direction (bottom-up using TUM-Ack messages); after the top-down phase is complete. TUM-Messages share information with the neighborhood, whereas the TUM-Ack message is unicast from the child node to its parent only. A TUM can be of two types, synchronous (periodic) or asynchronous (on-demand). In cross-layer wireless networks the selection of the TUM timing has impact on the performance of a congestion control algorithm. This is because the messages used to propagate beacons regarding the congestion control rate allocation as well as other overhead messages may be missed if the node is not in receive mode (for asynchronous messaging for example).

Figure 3 shows the latency in the flow of top-down parent information in terms of rounds. The root node (Sink) is represented by S where d1, d2, d3 represent the nodes at depth 1, 2 and 3 (1-hop, 2 and 3 hops away) from the sink. The TUM is broadcast after a periodic time interval of t0, t1 and t2 and so on. In first round parent information reaches the nodes which are 1-hop away from the sink. Here time A represents the initial randomised time for first
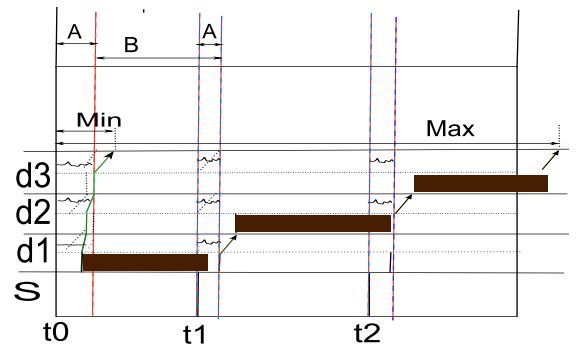
Adaptive Spanning Tree Formation():
for $\forall$ i, $i \in N$ **do**
  Round 0: Initialization
  $Self\_id(i) = Predefined\ constant$;
  $Parent(i) = Self\_id$;
  $N_i{}^1 = \emptyset$;
  $Children(i) = \emptyset$;
  $Good - Neighbour(i) = \emptyset$;
  $TotalChildern(i) = 0$;
  $TotalNeighbour(i) = 0$;
  $Goodness(i) = 0$;
  $Depth(i) = \infty$;
  $Rate(i) = Predefined\ constant$;
  $GlobalReAllocation = 0$;
  $Round = 1$;
  while $Round$ **do**
    TUM Send:
    if $TUM\text{-}Period\ over$ **then**
      Broadcast TUM;
      //TUM has following parameters:
      $Self\_id(i)$, $Parent(i)$,
      $TotalChildern(i)$, $TotalNeighbour(i)$,
      $Goodness(i)$,$Goodness(i)$,
      $Depth(i)$,$Rate(i)$,$GlobalReAllocation = 0$;
      Round++;
    end
    TUM Received:
    Call UpdateParent();
    Call UpdateNeighbourTable();
    Call UpdateRate();
    TUM-Ack Send:
    if $TUM\ Received\ from\ Parent$ **then**
      Send TUM-Ack to Parent, TUM-Ack contain GlobalReAllocation field
    end
    TUM-Ack Receive:
    Call UpdateChildren();
    Call UpdateRate();
  end
end

Algorithm 1: Adaptive Tree formation through Synchronous Periodic Topology Management

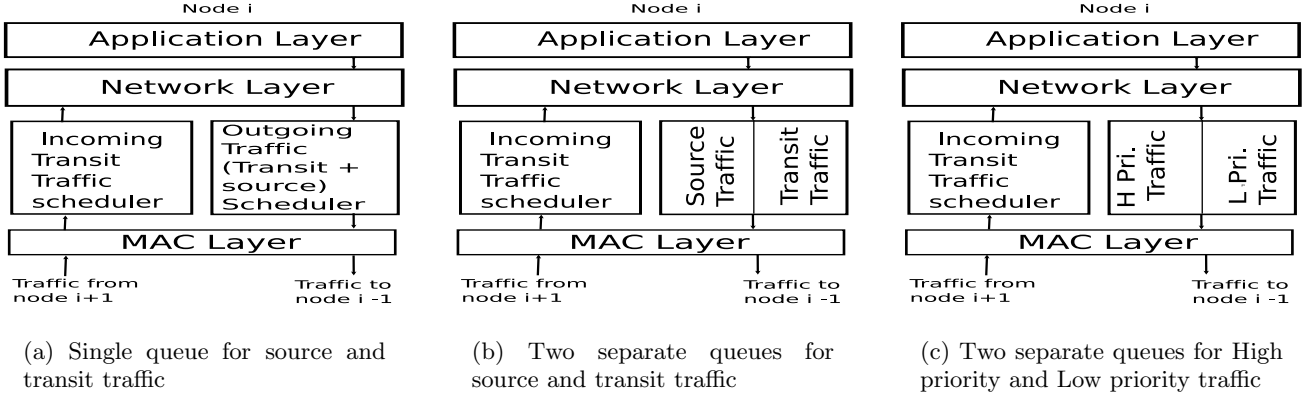**Figure 3: Beacon communication for Spanning Tree route discovery for n hops.**

Node i

| Application Layer |
| Network Layer |

| Incoming Transit Traffic scheduler | Outgoing Traffic (Transit + source) Scheduler |

| MAC Layer |

Traffic from node i+1   Traffic to node i -1

(a) Single queue for source and transit traffic

Node i

| Application Layer |
| Network Layer |

| Incoming Transit Traffic scheduler | Source Traffic | Transit Traffic |

| MAC Layer |

Traffic from node i+1   Traffic to node i -1

(b) Two separate queues for source and transit traffic

Node i

| Application Layer |
| Network Layer |

| Incoming Transit Traffic scheduler | H Pri. Traffic | L.Pri. Traffic |

| MAC Layer |

Traffic from node i+1   Traffic to node i -1

(c) Two separate queues for High priority and Low priority traffic

Figure 4: Ingress-egress traffic at Intermediate forwarding node

beacon message and B(t1-t0) presents the periodic topology update time interval. We can clearly see that the maximum set-up time for the top-down (parent setting phase) and bottom-up (children setting phase) phases depends on route update period and depth of the network. However the minimum time depends on the initial randomised time. Figure 5-b shows the topology set-up time for the topology shown in figure 5-a. In this figure, one-hop neighbors(nodes 2,3 and 4) from sink update their parent information by receiving TUM message in negligible time(the time required to extract information from received TUM and internal parameter updation). The two-hop neighbors(nodes 5,6,9,10) from sink update their parent information only when they receive valid TUM from their respective parents. We can see the latency in this updation because they received valid TUM only after TUM time period. And for three-hop neighbors(nodes 7 and 8), it's randomized time lesser than TUM period.

3.3 Traffic Differentiation Scheme

Once the spanning tree is in place we then provide a facility that allows the event traffic to be routed down the shortest path to the sink while other traffic is dispersed to longer paths. This Priority Based Routing technique is shown in Algorithm 2. To ensure Quality of Service, traffic differenti-

---

while $Node_i$ *receives Priority Traffic* **do**
  if *Congestion detected at* $node_{i-1}$ *due to event traffic* **then**
    | Congestion notified back to $node_i$;
    | $Node_i$ takes action to reduce traffic towards $node_{i-1}$;
 **end**
end

Algorithm 2: Priority Based Re-Routing with Congestion Control

---

ation and queue management strategies have been explored. Figure 4-a describes the single outgoing queue without any traffic differentiation; Figure 4-b describes the two separate queues one for locally generated traffic from the local sensor and forwarded traffic (generated at children nodes which is being routed through a given node). To provide better Quality of Service to event traffic, at each node a separate

outgoing queue is maintained for event traffic and periodic traffic. Traffic from the periodic queue is only transmitted when the event queue is empty. This reduces the latency in the event traffic, but at an obvious cost to the non-event traffic packet delivery time potentially.
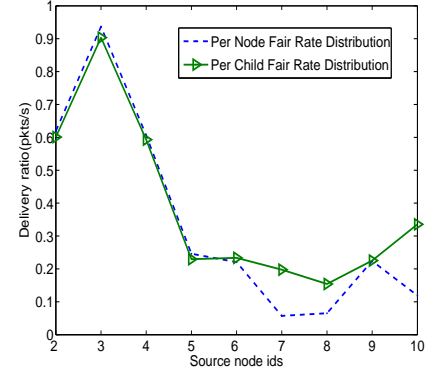
3.4 Adaptive Traffic Rate Control Scheme

Here we present our priority based adaptive rate control scheme. In priority based rate control, from figure 1 we were able to derive the maximum receiver capacity for each node. That is, it is clear from the 2-hop network that as nodes start generating packets at a high rate after 25pkts/sec, the goodput drops significantly. Therefore we have a rough approximation how much traffic a node can receive. That is, each node can handle a finite number of incoming packets, therefore the child nodes feeding this node must set their data generation rate such that if all child nodes are sending data simultaneously the parent can cope with the volume. The receiver capacity (receive packets/second) for a given parent node dictates the rate (send packets/second) of its child nodes. There are a number of ways to approach this. We evaluated two mechanisms namely per node fair rate distribution (NFR) and per child fair rate distribution (CFR) using the topology presented in Figure 5-a. NFR essentially counts the number of child nodes that are downstream from a given node and divides the rate amongst them. For example in our figure 5-a if we understand that node 4 has 100 packets/second receive capacity we divide the rate evenly amongst 4 child nodes downstream from it (i.e. their send rate is 25 packets/second each). On the other hand, CFR just looks at the child nodes one hop away and divides the rate amongst them. For node 4 we would divide the rate into 50 packets/second for both nodes 10 and 9. Then node 10 would divide the 50 packets/second into two for node 7 and 8 who get a rate allocation of 25 packets/second respectively. Figure 5-a shows the network topology considered in all experiments. To understand which scheme would be more useful we simulated a general event driven scenario, where all nodes periodically generate data, routing it to sink. An event is then detected by the node (id =7) and now intermediate nodes on the shortest path between this event node and the sink should be forwarding both priority and routine traffic towards sink. Figure 5-c shows the delivery ratio(pkts/sec) for the all individual nodes (2 to 10)
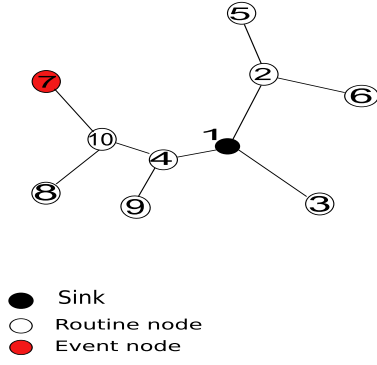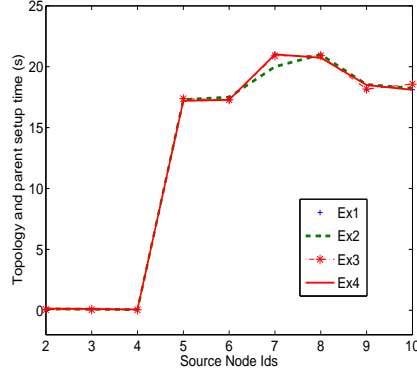
(a) Topology   (b) Topology Set-up Time   (c) Delivery Ratio

Figure 5: Figure-a shows the network topology considered in all experiments; note node 7 generates events. Figure-b topology set-up and parent discovery time in a three hop network where the global Beacon Interval(TUM period) is set at 16 seconds. Figure-c delivery ratio(pkts/sec) for all nodes (2 to 10) for the Per node fair rate distribution scheme and per child fair rate scheme with maximum receiving capacity assumed to be 0.5pkt/s with 25 bytes packet length.

for the two rate allocation schemes: first the per node fair rate distribution scheme and then, the per child fair rate scheme. The receivers maximum receiving capacity is assumed 0.5pkt/s whereas each packet is 25 bytes long.

UpdateRate():
if $Node_i$ *receives Total Traffic* $\geq PredefinedRate$ **then**
    if $Node_i$ *receives Event Traffic* **then**
        NewRate = $(PreviousRate * TotalChildren - IncomingEventTraffic)/TotalChildren$; **if** $NewRate \geq 0$ **then**
            Broadcast Topology Update Message with NewRate;
        end
        GlobalReAllocation=1;
        Broadcast Topology Update Message(TUM) with GlobalReAllocation field set.
    end
end
if $TUM\text{-}Ack.GlobalReAllocation == 1$ **then**
    // For received TUM-Ack- from Child NewRate = $(PreviousRate * (TotalChildren - 1))/TotalChildren$;
    GlobalReAllocation=1;
    Broadcast Topology Update Message(TUM) with NewRate
end
Algorithm 3: Rate Control in Congested Network
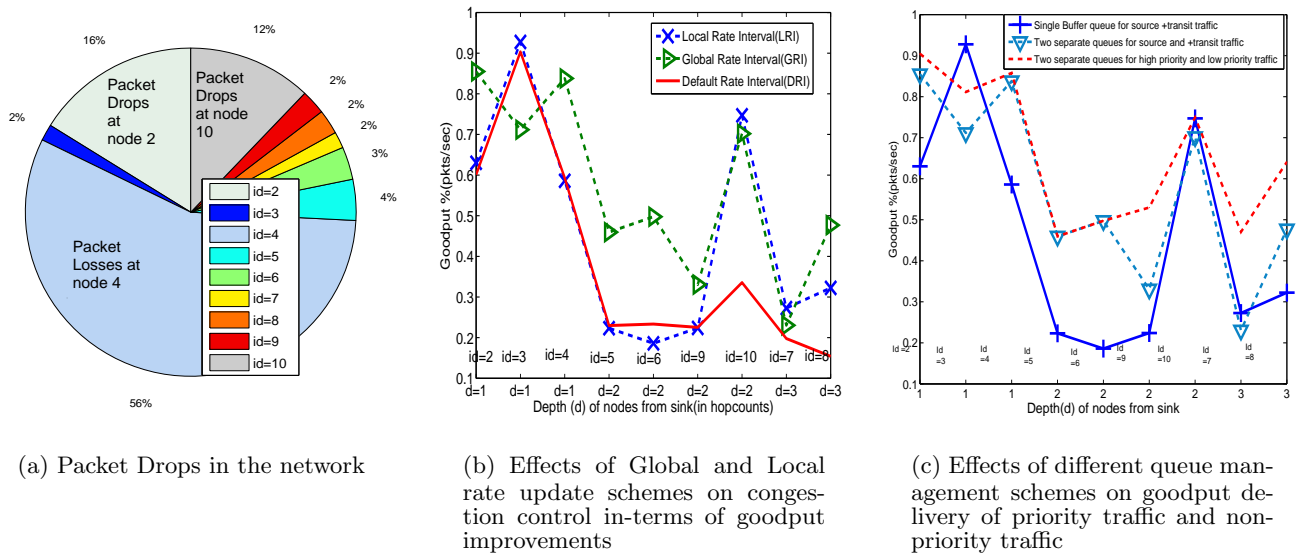

Figure 5-c clearly highlights that for node 7, the event generating node, we observe a better delivery ratio with CFR because the allocation is distributed for that particular branch reflecting its number of child nodes and the node depth; which provides a better allocation for unbalanced trees [4]. Given that under single hop networks or those that exhibit a balanced tree, both schemes behave in a similar way. As all other factors are almost equal, we select CFR as it gives advantages when the topology of the

network is less balanced. This leads to more packet loss and delay in the arrival of event data. Therefore, we present an alternative devolved decision making scheme that handles the decision making down at the event's parent node level and if that node cannot handle the traffic it recursively requests help in the form of receiver capacity rate re-allocation from its immediate parents. That is, when a node receives an event packet from one of its children it can reduce the traffic emanating from the other (non-event nodes in its sub-tree) children and/or its own traffic. Then if the packet rate remains so high that the parent node is exceeding its allocated receiver capacity then it recursively requests a global capacity re-allocation from the node above (which will be biased toward the sub-tree from which the event occurred to allocate it a higher rate). This has the advantages of forcing faster and focused decision making around the links that are involved with the event (which is typically clustered) and its message propagation to the sink. This happens recursively until there is enough allocated receiver capacity for that node, or we reach the sink. At this latter point we are in effect taking a global decision but without the lag time observed in previous work. Algorithm 3, shows both local rate control decisions and global rate control decisions based on the traffic congestion caused by the event traffic using Priority based re-distribution of Per-child allocation scheme (CFR).

4. EVALUATION

All algorithms presented in this paper are implemented in TinyOS2.x and evaluated in the Tossim-2 [12] Simulator for MICAZ devices [1][18].

In Figure 6-a, we study the data queues for nodes that are involved in forwarding packets through the network(for topology shown in figure 5-a). We calculate the packet drop percentage at individual nodes with respect to the total packets dropped in the network. It is very clear that nodes which has larger sub-tree (large number of descendants) are drop-

(a) Packet Drops in the network

(b) Effects of Global and Local rate update schemes on congestion control in-terms of goodput improvements

(c) Effects of different queue management schemes on goodput delivery of priority traffic and non-priority traffic

Figure 6: In Figure-a, percentage packet drop per node. Figure-b goodput with rate control. Figure-c three queue management scheme.

ping more packets as compared to other nodes. Figure 6-b illustrates the improvement in the goodput at all nodes when event traffic nodes detect congestion and take local and global rate control decisions respectively. The x-axis represents the nodes which are at depth 1, 2 and 3 from the sink. The Y-axis represents the goodput percentage share of each node in the total traffic received at sink. The local and Global rate updation schemes provides better throughput share for all nodes by 5-35% depending on location and sub-tree size of the node. To further understand the behavior of Congestion Control and Priority Based Routing (PBR), we implemented the algorithm 2 for Micaz motes and simulated in TOSSIM-2 using a small topology shown in figure 7-a. Here only one node generates events (node 7, Grey), the results are shown in figures 7-b & 7-c. Figure 7-b, shows the priority data throughput(%) observed over variable traffic loads in the network [4]. It is very clear from the figure that an efficient Congestion Control scheme improves the delivery of event data and also we notice that in a two hop network, PBR is improves throughput by ($\sim 10\%$) compared to TinyOS's standard Multihop-LQI Algorithm (shown as B-MAC on the graph). Similarly, 7-c, records that total traffic that reaches the sink for a simulation of duration of 93 seconds. Here we find that the total traffic throughput is also better in PBR with the added help from the congestion control mechanism from B-MAC compared to without any congestion control mechanism.

To further improve the efficiency of the PBR, we are proposing a middle layer exists between the network layer and the MAC layer, that acts as scheduler for incoming and outgoing traffic while also providing congestion control as shown in figure 4 and evaluated for topology shown in figure 5-a and results are presented in figure 6-c. Figure 6-c represents the queue management scheme; that is a single queue for all traffic is used initially. The second strategy is to implement two different queues, one for traffic that is from the local sensor and needs communicating and other one is for the

traffic that is being forwarded by the node. The last scheme aims to provide high-priority to event traffic, therefore two separate queues are held but now one is for high priority event data and the other one for periodic traffic. Here we see that an improvement of 5-35% (depending on location and sub-tree size of the node) in goodput is observed by placing the data in different queues per node allowing the event data to be sent first.

5. CONCLUSION

In this paper we briefly examine QoS and congestion control requirements for event driven pervasive sensor networks and summarized our QoS and congestion control strategies showing that through a combination of rate control and traffic differentiation using multi communication queues we can significantly improve both the reliability and latency of high-priority packets. Future research aims to answer the question: Does the QoS strategy interfere with congestion control strategies when the QoS strategy is independent of congestion control? Here the behavior of the underlying MAC layer in pervasive networking can potentially cause significant differences in the behavior of our simple rate control and traffic re-routing algorithm. We wish to explore this and perhaps better understand cross-layer effects therein.

6. REFERENCES

[1] Micaz datasheet. http://www.xbow.com/Products/Product. Accessed on 21/09/08.

[2] R. Belecheanu, G. Jawaheer, A. Hoskins, J. A. McCann, and T. Payne. Semantic web meets autonomic ubicomp. In *Workshop on Semantic Web Technology for Mobile and Ubiquitous Applications*, 2004.

[3] S. Bhatnagar, B. Deb, and B. Nath. Service differentiation in sensor networks. In *WPMC'01:Proceedings of Wireless Personal Multimedia Communications*, 2001.

[4] M. Breza, P. Martins, J. A. McCann, E. Spyrou, P. Yadav, and S. Yang. Simple solutions for the second decade of

(a) Topology   (b) Event(Priority data) through-   (c) All traffic
put rate
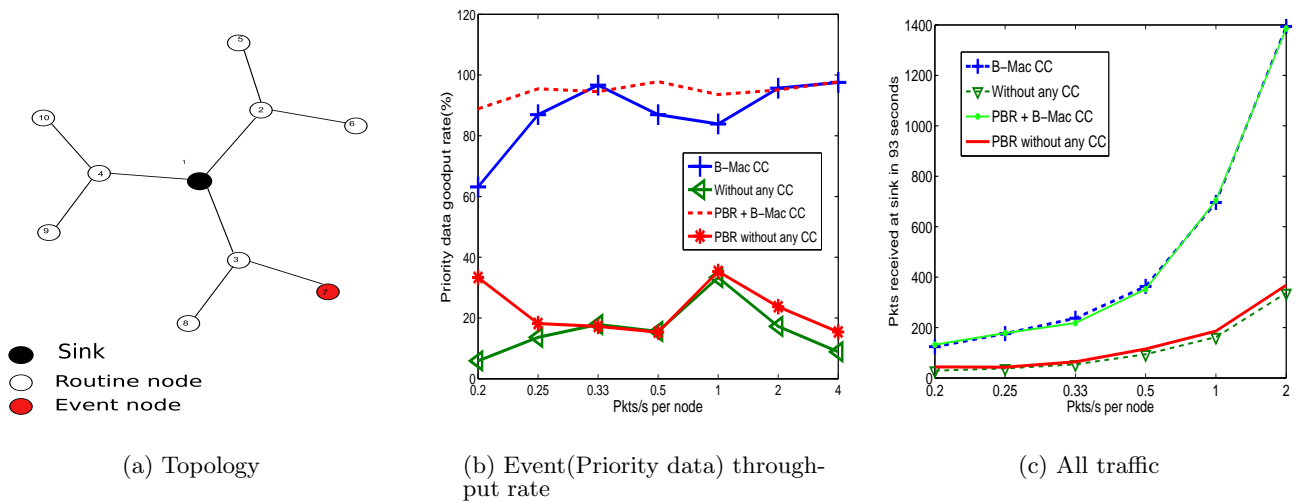
Figure 7: (a) Topology ; (b) Comparison of event throughput rates at the sink when Priority Based Routing is used with and without the B-MAC Congestion Control (CC) scheme with varying the packet generation rates per node. (c) Total throughput (Priority +non-priority) received by the sink when Priority Based Routing is used with and without the B-MAC Congestion Control (CC) scheme

wireless sensor networking. In *ACM-BCS Visions of Computer Science 2010*, Edinburgh, UK, April 2010.

[5] W. Cai, X. Jin, Y. Zhang, K. Chen, and R. Wang. Aco based qos routing algorithm for wireless sensor networks. In *UIC*, pages 419–428, 2006.

[6] Y. Chen and N. Nasser. Enabling qos multipath routing protocol for wireless sensor networks. pages 2421–2425, May 2008.

[7] L. Dehni, Y. Bennani, and F. Krief. *Mobility Aware Technologies and Applications*, chapter LEA2C: Low Energy Adaptive Connectionist Clustering for Wireless Sensor Networks, pages 405–415. Springer Berlin / Heidelberg, 2005.

[8] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *SenSys '09: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 1–14, New York, NY, USA, 2009. ACM.

[9] X. huang Zhang and W. bo Xu. *Agent Computing and Multi-Agent Systems*, chapter QoS Based Routing in Wireless Sensor Network with Particle Swarm Optimization, pages 602–607. Springer Berlin / Heidelberg, 2006.

[10] B. Hull, K. Jamieson, and H. Balakrishnan. Mitigating congestion in wireless sensor networks. In *SenSys '04: Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems*, pages 134–147, Nov 2004.

[11] H. A. L. Savidge, H. Lee and A. Goldsmith. Event-driven geographic routing for wireless image sensor networks. In *COGnitive systems with Interactive Sensors (COGIS)*, March 2006.

[12] H. Lee, A. Cerpa, and P. Levis. Improving wireless simulation through noise modeling. In *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*, pages 21–30, New York, NY, USA, 2007. ACM.

[13] S. Munir, Y. W. Bin, R. Biao, and M. Man. Fuzzy logic based congestion estimation for qos in wireless sensor network. pages 4336–4341, March 2007.

[14] N. Ouferhat and A. Mellouck. Qos dynamic routing for wireless sensor networks. In *Q2SWinet '06: Proceedings of*

the 2nd ACM international workshop on Quality of service & security for wireless and mobile networks*, pages 45–50, New York, NY, USA, 2006. ACM.

[15] A. Papadopoulous, A. Navarra, and J. A. McCann. Vibe: a virtual-infrastructure-based energy-efficient framework for routing over scalable wireless sensor networks. In *International Conference on Distributed Computing in Sensor Systems(DCOSS)*, 2008.

[16] R. H. Perrott, B. M. Chapman, J. Subhlok, R. F. de Mello, and L. T. Yang. Third international conference on high performance computing and communicationsHPCC'07, houston, usa, september 26-28, 2007,. In *HPCC*, volume 4782 of *Lecture Notes in Computer Science*. Springer, 2007.

[17] A. Sridharan and B. Krishnamachari. Explicit and precise rate control for wireless sensor networks. In *SenSys '09: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 29–42, New York, NY, USA, 2009. ACM.

[18] W. Su and M. Alzaghal. Channel propagation characteristics of wireless micaz sensor nodes. *Ad Hoc Networks*, 2008.

[19] W. L. Tan, O. Yue, and W. C. Lau. Performance evaluation of differentiated services mechanisms over wireless sensor networks. In *VTC'06: Proceedings of 64th IEEE Vehicular Technology Conference*, pages 1–5, 2006.

[20] D. J. Vergados, N. A. Pantazis, and D. D. Vergados. *Mobile Networks and Applications*, volume Volume 13, chapter Energy-Efficient Route Selection Strategies for Wireless Sensor Networks, pages 285–296. Springer Netherlands, 2008.

[21] X. Wang, X. Zhao, Z. Liang, and M. Tan. Deploying a wireless sensor network on the coal mines. pages 324–328, April 2007.

[22] P. Yadav and J. A. McCann. Qos based event delivery for disaster monitoring applications. In *Proceeding of the 5th International Conference on Wireless Communication and Sensor Networks(WCSN'09)*, India, December 2009.