

Pritrans: A Prioritized Converge-cast Scheme using Consensus Ranking in Wireless Sensor Networks

Shao Tao and Mun Choon Chan
National University of Singapore, Singapore
Email: {shaot, chanmc}@comp.nus.edu.sg

Sergey Muravyov and Evgeny Tarakanov
Tomsk Polytechnic University, Tomsk, Russia
Email: muravyov@camsam.tpu.ru, tark@tpu.ru

Abstract—Data packets collected in wireless sensor networks may contain information of different importance, including regular updates and critical data triggered by important events. In a converge-cast tree, packets routed towards the data sink may be dropped due to network contention. In order to improve the reliability of critical packets, we propose a prioritized transmission scheme named *Pritrans*. By overhearing the data packets transmitted by the neighbors, each node will compute the *consensus relation* based on the packet contents and maintain a *consistent ranking* within the neighborhood. The parent node will send feedback messages to adjust the packet rate of child nodes proportional to their ranking, such that the channel can be prioritized for transmission of important packets. A preemptive queuing method is applied in the intermediate nodes, enabling an incoming packet with high priority to evict a low priority one when the queue is full. The simulation results in TOSSIM show that the *Pritrans* scheme can effectively increase the delivery rate of critical packets in a congested network scenario.

Keywords: converge-cast, consensus relation, adaptive rate, preemptive queuing

I. INTRODUCTION

Wireless sensor networks have shown great potential for environment monitoring [1], target tracking and security surveillance tasks [2], where sensor nodes collect information from the deployment space and propagate it back to the data sink. The forwarding path towards the data sink is determined by distance metrics such as hop count, delay or ETX [3]. As the packets approach the sink, the converge-cast traffic may cause hot-spots around the sink node. In order to mitigate the impact of such congestion, packets triggered by important events should be given higher priority to access the network than the packets carrying merely periodic updates.

In this paper, we present a prioritized transmission scheme named *Pritrans*. In *Pritrans*, a node overhears the data packets, computes the *consensus relation* [4] and estimates its relative ranking to its neighbors in real time. As the rankings are consistent within a neighborhood, each parent node can adaptively update their child nodes' packet rate proportional to the *importance of the reported data*. *Pritrans* also incorporates a preemptive queuing method, allowing the critical packets to evict the non-critical ones in the queue, such that the total number and the delivery ratio of the critical packets can be improved.

The rest of the paper is organized as follow. Related work is presented in Section II. The packet rate adaption scheme and the preemptive buffer management algorithm are introduced

in Section III. The performance of *Pritrans* is evaluated in Section IV and we conclude in Section V.

II. RELATED WORK

The spanning tree topology is commonly used to route data packets from sensor nodes to the data sink. The Adaptive Robust Topology control algorithm (ART) [5] can adjust the transmission power according to the link quality, while maintaining the required packet reception ratio with lower power consumption. In order to facilitate point to point data delivery on a tree topology, various coordinate systems can be applied to locate each node. The VPCR [6] protocol assigns polar coordinates to the nodes based on the node position and subtree size. The packet is passed towards the root until the subtree containing the destination is located, from where, the data packet can routed by following the polar coordinates. For many sensor network applications, concurrent transmission without congestion collapse is required. Woo *et al.* [7] proposed an adaptive rate control algorithm in the MAC layer, that is energy efficient and allow the fair share of bandwidth between the route-thru traffic and originating traffic. RCRT [8] is a reliable transport protocol which uses end-to-end loss recovery and perform rate adaption at the sink. Each node will keep a copy of the outgoing packets and check for the cumulative ACK for data retransmission. In the Flush [9] protocol, the minimum inter packet delay is estimated to avoid intra-flow interference. It also uses end-to-end communication to ensure packet delivery and hop by hop rate control to reduce packet loss due to congestion. The Reliable Bursty Converge-cast (RBC) [10] combines windowless block acknowledgement with differentiated contention control. Packets are stored in priority queues where the ones that require retransmission are transmitted first.

Previous works do not address the issue of effective data transmission scheduling. In this work, we address the scheduling problem using voting techniques to determine priority. We believe that this is the first attempt to apply consensus relation based voting techniques in this problem domain.

III. PROTOCOL DESIGN

In this section, we first introduce the consensus relation over a set of packets by calculating the Borda count based on the sensor information in the packets and then show

TABLE I
DATA IN PACKETS OF A FIRE SENSOR NETWORK

	<i>Location</i>	<i>Temp</i> (°C)	<i>Temp_rate</i> (°C/s)	<i>Smoke_density</i>
p_1 :	0.9	300	50	0.3
p_2 :	0.5	100	10	0.4
p_3 :	0.1	150	20	0.2
p_4 :	0.4	200	10	0.3
p_5 :	0.8	150	30	0.6
p_6 :	0.7	300	20	0.6

how Pritrans adjusts the packet rate and performs preemptive queuing according to the consensus relation.

A. Consensus Relation of Data Packets

Suppose one needs to order objects by the value of some attribute manifestation that is used to express some **preference relation** over the object set. Usually, the notation of preference is modelled by means of specially constructed binary relation, which is called a **weak order**. Let us introduce it formally.

Let $A = \{a_1, a_2, \dots, a_n\}$ be a set of alternatives (objects) being compared. We will call a binary relation λ by the preference relation over set A , if

$$\lambda = \rho \cup \tau \quad (1)$$

That is, the binary relation λ is the union of two relations: a **strict order** ρ , which is antireflexive ($\overline{a_i \rho a_i} \forall i$), antisymmetric ($\forall i, j, a_i \rho a_i$ and $a_j \rho a_i$ imply $a_i = a_j$) and transitive ($\forall i, j, k$ from $a_i \rho a_j$ and $a_j \rho a_k$, it follows that $a_i \rho a_k$), and **indifference relation** τ , which is reflexive ($a_i \tau a_i \forall i$) and symmetric ($\forall i, j, a_i \tau a_j$ implies $a_j \tau a_i$). The former is said a_i to be **strictly preferred** over a_j and denoted $a_i \rho a_j$ or $a_i > a_j$. The latter is said a_i and a_j to be **indiscernible** and denoted $a_i \tau a_j$ or $a_i \sim a_j$. In many cases, the indifference relation can be understood as an **equivalence** equation.

Thus, the preference relation λ is reflexive, ($a_i \lambda a_i \forall a_i \in A$), transitive ($\forall i, j, k$ from $a_i \lambda a_j$ and $a_j \lambda a_k$ it follows that $a_i \lambda a_k$) and complete ($\forall a_i, a_j \in A$ either $a_i \lambda a_j$, or $a_j \lambda a_i$ is valid).

The preference relations are usually represented by object rankings of the following form:

$$\lambda_k = (a_1 > a_2 > \dots \sim a_s \sim a_t > \dots \sim a_n) \quad (2)$$

The preference relation λ can also be represented by a $(n \times n)$ matrix $B = [b_{ij}]$, in which the rows and columns of B correspond to alternatives of A , and its elements allow to distinguish the relations $>$, $<$ and \sim .

$$b_{ij} = \begin{cases} 1 & \text{if } a_i > a_j \\ 0.5 & \text{if } a_i \sim a_j \\ 0 & \text{if } a_i < a_j \end{cases} \quad (3)$$

Notice that this matrix is transitive and $b_{ij} + b_{ji} = 1 \forall i \neq j$.

Suppose now that n objects are described by m preference relations. Take the **example of a fire sensor network**. The information collected by the sensors may contain the following: *temperature*, *temperature increasing rate*, *location* of the sensor and *smoke density*. The example in Table I shows the packets p_1 through p_6 received by a node from its child nodes N_1 through N_6 on the spanning tree. The attributes *location*

and *smoke_density* are dimensionless quantities, normalized between 0 (less critical) to 1 (more critical).

For each attribute, we have a ranking of the packets according to its value:

$$\begin{aligned} \lambda_1 \text{ by } Location &: & p_1 > p_5 > p_6 > p_2 > p_4 > p_3 \\ \lambda_2 \text{ by } Temp &: & p_1 \sim p_6 > p_4 > p_3 \sim p_5 > p_2 \\ \lambda_3 \text{ by } Temp_rate &: & p_1 > p_5 > p_3 \sim p_6 > p_2 \sim p_4 \\ \lambda_4 \text{ by } Smoke_density &: & p_5 \sim p_6 > p_2 > p_1 \sim p_4 > p_3 \end{aligned}$$

Thus, we have the set of rankings $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_4\}$ over the set of packets $P = \{p_1, p_2, \dots, p_6\}$. The set of rankings Λ is called a **preference profile** for given m and n . Our aim is to determine a **unique** preference relation β that provides the best compromise for m rankings. This relation β is called **consensus relation**.

Taking into account Equation 3, the preference profile can be represented by the $(n \times n)$ **electoral matrix** (or **tournament table**) $S = [s_{ij}]$, where

$$s_{ij} = \sum_{k=1}^m b_{ij}^k, \quad i, j = 1, \dots, n; \quad (4)$$

$$b_{ij}^k = \begin{cases} 1 & \text{if } a_i^k > a_j^k \\ 0.5 & \text{if } a_i^k \sim a_j^k \\ 0 & \text{if } a_i^k < a_j^k \end{cases} \quad (5)$$

The matrix S possesses the evident property: $s_{ij} + s_{ji} = m$. For our example, the corresponding tournament table is shown in Table II.

TABLE II
TOURNAMENT TABLE FOR THE EXEMPLARY 4 ATTRIBUTES AND 6 PACKETS

	p_1	p_2	p_3	p_4	p_5	p_6	Counts
p_1	0	3	4	3.5	3	2.5	16
p_2	1	0	2	2.5	0	0.5	5.5
p_3	0	2	0	1	0.5	0.5	4
p_4	0.5	1.5	3	0	1	0	6
p_5	1	4	3.5	3	0	2.5	14
p_6	1.5	4	3.5	4	1.5	0	14.5

The problem of consensus relation determination has been under intensive investigation in the framework of Social Choice Theory since the late XVIII century (see, for example, [11], [12]). It was shown that the best consensus relation can be found by **Condorcet method**: in each comparison, the preferred alternatives is the alternative preferred by a majority of votes, that is, $a_i > a_j$ iff $s_{ij} > s_{ji}$. In other words, if the alternative a_i obtains a majority of votes in pair-wise contests against every other alternative, the alternative is chosen as the winner. By this method, the consensus relation for our example is $p_1 > p_5 > p_6 > p_2 > p_4 > p_3$. Unfortunately, a difficulty with the Condorcet winner is that in general it needs not exist due to a common intransitivity of preference profile (so called Condorcet Paradox).

It is known that the best approximation to Condorcet consensus relation is so called **Kemeny Median** [13], [14] defined by the formula:

$$\beta = \arg \min_{\lambda \in \Pi} D(\lambda, \Lambda) \quad (6)$$

TABLE III
SCORES OBTAINED FROM EACH ATTRIBUTE IN A PACKET

	Location	Temp	Temp_rate	Smoke_density	Total	Rank
p_1	5	4.5	5	1.5	16	1
p_2	2	0	0.5	3	5.5	5
p_3	0	1.5	2.5	0	4	6
p_4	1	3	0.5	1.5	6	4
p_5	4	1.5	4	4.5	14	2
p_6	3	4.5	2.5	4.5	14.5	3

where $D(\lambda, \Lambda)$ is a distance between ranking λ and profile Λ , Π is a set of all $n!$ strict orders \succ on A . However, in this case, there are at least two difficulties: first, the Kemeny Median Problem is NP-complete and second, the Kemeny consensus relation can be non-unique (and the number of the problem solutions can be considerably high).

Accounting for the above considerations, in the calculation of consensus relation, we have selected the so called **Borda Count** due to its two favorite properties: it is a good approximation of Condorcet method and the corresponding consensus relation can be found in polynomial time. Jean-Charles de Borda in 1784 proposed the approach in which each alternative is given a number of points depending on its position in the ranking: 0 point for placing last, 1 point for placing next to last, and so on, up to $n-1$ points for placing first. The scores of each alternative are aggregated and its position in the consensus ranking is defined by the obtained sum.

It is clear that the total sum obtained by each alternative equals the sum of elements of the corresponding row in tournament Table II. The formal description of the procedure is presented by Algorithm 1. These sums are shown in the rightmost column in Table II for our example. It follows from these data that the final ranking (consensus relation) of the packets is $p_1 \succ p_5 \succ p_6 \succ p_4 \succ p_2 \succ p_3$, according to their total score. The total score attribute distribution for the 6 packets and appropriate ranks are given in Table III.

The formal description of the procedure is presented by Algorithm 1. The packets' ranks (their places in the final ranking) define their priorities during packet transmission and they are also used to adjust the packet forwarding intervals at their respective hop senders.

Algorithm 1 Determine the consensus ranking by Borda Count

- 1: Let S be the tournament table with m rankings of n packets.
 - 2: Let $RowSum_i$ be the i -th tournament row sum, $i \in [1, n]$
 - 3: Let $CR = \{cr_1, cr_2, \dots, cr_n\}$ be the consensus ranking.
 - 4: **for** $i \leftarrow 1, n$ **do**
 - 5: $cr_i \leftarrow i +$; $RowSum_i \leftarrow 0$;
 - 6: **end for**
 - 7: **for** $i \leftarrow 1, n$ **do**
 - 8: **for** $j \leftarrow 1, n$ **do**
 - 9: $RowSum_i + = s_{ij}$;
 - 10: **end for**
 - 11: **end for**
 - 12: Sorting $RowSum_i$ and cr_i in descending order.
-

Next, we will present the algorithm by which a node regulates its packet interval according to its rank. We also

explain how packet eviction is applied in the preemptive queue.

B. Adaptive Transmission Rate

To prevent the converge-cast traffic from overloading the network, each node N_i sends packets at the forwarding interval T_i , leading to an outgoing packet rate of $\frac{1}{T_i}$. As the packets flow towards the data sink, nodes closer to the data sink will require a lower forwarding interval thus higher rate than the nodes located around the network perimeter. Such a requirement leads to a natural need to priority scheduling at the nodes.

Once the data collection tree is established, the parent node will maintain a child list by overhearing the beacons. The child list consists of all the one hop neighbors that report data back to the node. After receiving the first round of packets, the parent will rank the child nodes based on the data reported. Child nodes reporting more critical attribute values will be assigned smaller packet forwarding intervals and vice versa. The previously received data from each child is stored in the parent's cache such that the relative ranking can be refreshed upon the arrival of new packets. The packet forwarding interval allocated to each child is derived from the parent's interval, in order to ensure that each child only consumes a fraction of the available bandwidth and that the aggregated data traffic will not overwhelm the network capacity. From the data sink to the leaf nodes, the packet forwarding intervals are regulated at each level, such that the data traffic generated in the network will not incur severe congestion at the root.

Algorithm 2 Adjust the Packet Forwarding Interval

- 1: Let T be the forwarding interval of the parent node.
 - 2: Let $N_i(i \in [1, k])$ be the k child nodes of a parent node.
 - 3: Let r_N be the rank of child node N .
 - 4: Initialize r_{max} and sum_rank to 0.
 - 5: **for** $i \leftarrow 1, k$ **do**
 - 6: $sum_rank += N_i.rank$;
 - 7: **if** $N_i.rank > r_{max}$ **then**
 - 8: $r_{max} = N_i.rank$;
 - 9: **end if**
 - 10: **end for**
 - 11: $sum_rank' = k \times (r_{max} + 1) - sum_rank$.
 - 12: forwarding interval of child N is $T_N = T \times \frac{sum_rank'}{r_{max} - r_N + 1}$
-

Periodically, a parent node sends *feedback* messages to the child nodes to announce the updated packet intervals that each child node should adopt. Each feedback message is a broadcast, containing the packet intervals for multiple children. When the child list does not fit in the payload of a single message, the feedback will cycle through the list and the packet interval updates will be received over a longer period.

The packet interval of a child node is computed by Algorithm 2. The initial packet interval T of each node is set to 100 ms, defining the maximum incoming packet rate that the data sink will accept. The number of child nodes k is determined by the network density and the spanning tree topology, which can be learned from periodical beacons. The node's rank r_N is computed by the method introduced in Section III-A. The

packet interval T_N of a child N is calculated by Equation 7,

$$T_N = \frac{T \times \text{sum_rank}'}{r_{\max} - r_N + 1} \quad (7)$$

where T is the forwarding interval of the parent and r_{\max} is the maximum rank value (lowest priority) among the child nodes.

Given k child nodes, the aggregated packet rate R can be computed by Equation 8.

$$\begin{aligned} R &= \sum_{i=1}^k \frac{r_{\max} - r_i + 1}{T \times \text{sum_rank}'} = \frac{k \times (r_{\max} + 1) - \sum_{i=1}^k r_i}{T \times \text{sum_rank}'} \\ &= \frac{k \times (r_{\max} + 1) - \text{sum_rank}}{T \times \text{sum_rank}'} = \frac{1}{T} \end{aligned} \quad (8)$$

Thus, the incoming packet rate at the parent node will add to $\frac{1}{T}$, no more than its allocated bandwidth. For two child nodes N_i and N_j with ranks r_i and r_j , the ratio of their packet intervals will be $\frac{r_{\max} - r_j + 1}{r_{\max} - r_i + 1}$, showing that the packet forwarding rate is assigned proportional to the node's relative ranking in the cluster.

Suppose a parent node has 4 child nodes N_1 through N_4 with rank $r_1 = r_2 = 1$ and $r_3 = r_4 = 2$. Nodes with smaller rank values have higher priority. The maximum rank value r_{\max} is 2. The variable $\text{sum_rank}'$ is computed as in Equation 9. The packet interval for N_1 is given in Equation 10.

$$\begin{aligned} \text{sum_rank}' &= \text{num_nodes} \times (\text{max_rank} + 1) - \\ &\quad \text{sum_rank} \\ &= 4 \times (2 + 1) - (1 + 1 + 2 + 2) = 6 \end{aligned} \quad (9)$$

$$T_1 = T \times \frac{\text{sum_rank}'}{r_{\max} - r_N + 1} = T \times \frac{6}{2 - 1 + 1} = 3T \quad (10)$$

Similarly, we can compute $T_2 = 3T$ and $T_3 = T_4 = 6T$. The aggregated packet rate of N_1 to N_4 is $\frac{1}{3T} + \frac{1}{3T} + \frac{1}{6T} + \frac{1}{6T} = \frac{1}{T}$, equivalent to the parent node's forwarding rate. Node N_1 and N_2 achieve a forwarding rate of $\frac{1}{3T}$, twice of that of N_3 and N_4 .

C. Preemptive Queuing

Each node maintains an output queue to buffer the packets that cannot be delivered to the next hop in time due to network congestion or limited link bandwidth. As the wireless sensor nodes have stringent memory constraints (e.g. 4 KB for MICAz), when the queue is fully occupied, subsequent incoming packets will be dropped. Instead of applying the traditional FIFO packet forwarding scheme, Pritrans incorporates a preemptive queuing mechanism to provide high delivery ratio to more important packets. The details are summarized in Algorithm 3.

Pritrans gives the highest priority to the *beacon* messages, as the network topology is maintained by the periodic exchange of beacons. The *feedback* messages have the second highest priority, as they help to regulate the network traffic and avoid collision collapse. A *data* message will only compete with other *data* messages. The priorities of the data packets are computed based on the attribute values through the consensus relation function described in Section III-A.

Algorithm 3 Insert Packets to Preemptive Output Queue

```

1: Let  $Q$  be the output radio queue with capacity  $k$ 
2: Let  $p'$  be the new packet to insert
3: Let  $p_i$  be the existing packets in  $Q$ 
4: if  $Q.size < k$  then                                      $\triangleright$  the queue is not full
5:   insert  $p'$  to the end of queue  $\square$ 
6: else                                                        $\triangleright$  the queue is full
7:   if  $p'.type$  is Beacon then
8:     for each packet  $p_i$  in queue  $Q$  do
9:       replace  $p_i$  with  $p'$ , if  $p_i$  is Feedback or Data  $\square$ 
10:    end for
11:   end if
12:   if  $p'.type$  is Feedback then
13:     for each  $p_i$  in queue  $Q$  do
14:       replace  $p_i$  with  $p'$ , if  $p_i$  is Data  $\square$ 
15:     end for
16:   end if
17:   if  $p'.type$  is Data then
18:     for each  $p_i$  in queue  $Q$  do
19:       Replace  $p_i$  with  $p'$ , if  $p_i$  is Data with lower priority  $\square$ 
20:     end for
21:   end if
22: end if

```

Upon receiving an incoming packet, if the outgoing packet queue has free slots, the packet will be appended to the tail of the queue. When the queue is full, the incoming packet will be compared to the existing ones starting from the head of the queue. A beacon packet will replace the first feedback or data packet encountered. A feedback packet will replace a data packet but not feedback packets. A data packet with the most critical information will be placed ahead of other data packet in the output queue. If a packet eviction cannot be performed in the queue, the received packet will be dropped. It is also possible to implement priority queues to separate the control communication from the data traffic. Due to memory constraint, the current design of Pritrans utilizes a shared queue.

IV. PERFORMANCE EVALUATION

To evaluate the effects of prioritized transmission on converge-cast, we implement Pritrans in TinyOS-2.1 [15] and perform simulation using TOSSIM. The simulation parameters are listed in Table IV. We use 241 nodes to form 15 clusters as depicted in Fig. 1. Node 1 is the sink and node 2 to 16 are the cluster heads. Each cluster head collects the data packets from the sensor nodes and propagates the packets to the sink. The initial packet interval is set to 100 ms and the data packet size is 60 bytes. The CC2420 radio in TOSSIM has a bandwidth of 256 kbps and radio buffer size is set to 10 packets. The feedback packets are sent every 1 second, in order to respond to the dynamic behavior of target events quickly.

We assume the nodes form a flat tree topology where the leaf nodes align linearly. A mobile fire source moves from node 24 towards node 234. The temperature sensed by nodes within 3 hops of the fire source, named *hot nodes*, is 500 °C. For nodes 4 to 7 hops away from the fire source, the sensed temperature decreases linearly from 500 °C to the normal temperature of 25 °C. When the distance between the fire source and the leaf nodes is more than 7 hops, the temperature

TABLE IV
SIMULATION PARAMETERS

Name	Value
Number of Nodes	241
Number of Clusters	15
Cluster Size	16
Beacon Interval	10 s
Feedback Interval	1 s
Forward Interval	init/min: 100 ms, max: 15 s
Data Packet Size	60 bytes
Routing Protocol	Collection Tree
Radio Bandwidth	CC2420, 256 kbps
Radio Buffer Size	10 packets

readings will return to the nominal value of 25 °C.

The entire simulation duration is 1740 seconds, among which the initial 30 seconds is the warm up period for parent selection and no data packets are sent during that time. The fire source starts at node 24 and moves towards node 234 at the speed of 1 hop/s. It stops at nodes 24, 129 and 234 for 500 seconds each and nodes around the fire source will detect the highest temperature.

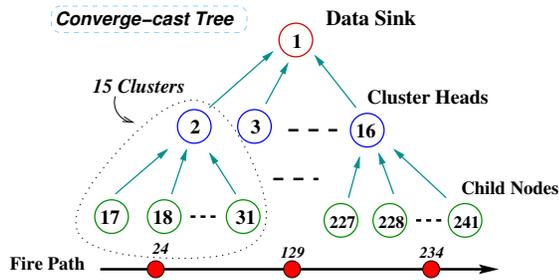


Fig. 1. Converge-cast Tree with 241 nodes grouped in 15 clusters

The sensor nodes report data packets to the parent nodes at the specified forwarding rate. After putting the received packet in the queue, the parent will update the ranking and recompute the packet forwarding rate for each child. The updated packet rates are sent back to the child nodes as feedback messages in a batch. The feedback messages from the cluster heads will increase the packet rate of the *hot* nodes and suppress the packet rates of the rest. The queue management scheme in Pritrans also gives preference to high priority data packets during the store and forward process. We measure the number of packets received from each node at each level and compute the perceived packet rate as the performance metrics. Although temperature is the only attribute considered in the simulation, Pritrans can handle more complex scenarios where multiple modalities are used for consensus relation calculation. Therefore, it is more adaptable to sensor applications compared to a threshold based method.

As the fire source stops at node 24, 129 and 234 for 500 seconds, the nodes within 3 hops around it will be assigned high packet rates and return more packets. The cluster head returning the data packets with higher temperature will obtain more bandwidth from the feedback messages of the sink. Subsequently, as the fire source moves at the leaf level, the

total number of packets and average packet rate of the nodes in the hot area will remain consistently higher than the nodes in the cold area. The incoming packet rates for both high and low priority packets at the sink are depicted in Fig. 2(a). The low priority packets achieve an average packet rate of 0.067 packets/s. With Pritrans, the high priority packet rate remains between 0.21 and 0.25 packets/s, with a much higher average of 0.23 packets/s. For comparison, the packet rates without Pritrans are illustrated in Fig. 2(b). When all packets randomly contend for network access regardless of their importance, both high and low priority packets have an equivalent packet rate of 0.07 packets/s. Under Pritrans, the high priority nodes can send packets reliably faster, allowing the data sink to retrieve more information from the area of interest.

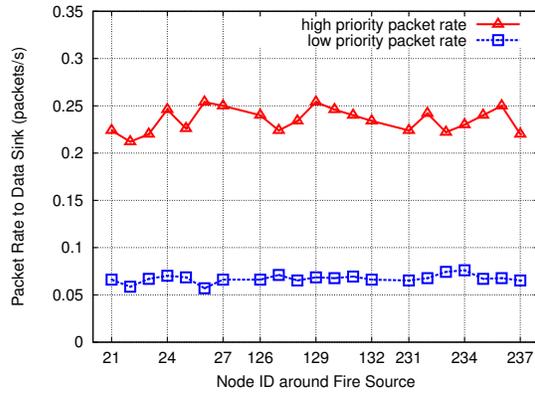
In order to demonstrate the effects of Pritrans on individual nodes, we plot the packet rates of a leaf node 129 and its corresponding cluster head node 9 in Fig. 3(a) and Fig. 3(b). The fire source arrives at node 129 at the time of 635 seconds and pauses for 500 seconds. During that time, the temperature of nodes 126 ~ 132 will become higher than other leaf nodes and their sending rate will be increased by the adaptive packet rate scheme in Pritrans. In Fig. 3(a), the outgoing packet rate of node 129 increases from 0.26 packets/s to 0.85 packets/s. The cluster head of node 129 is node 9. In Fig. 3(b), the aggregated incoming packet rate at node 9 is 2.77 packets/s when none of its child nodes is affected by the fire source. When the fire source approaches node 129, the aggregated packet rate grows to 5.49 packets/s, equal to the combination of the packet rates from node 126 to 132. This implies that even though the outgoing traffic of multiple nodes are increased simultaneously within a cluster, it does not adversely increase the network congestion level, as the data rates are adjusted by the cluster head according to its own capacity.

The results in Fig. 2 and Fig. 3 reveal that Pritrans can effectively increase the transmission rate of high priority packets delivered to the data sink by using adaptive node forwarding intervals and the preemptive queuing scheme.

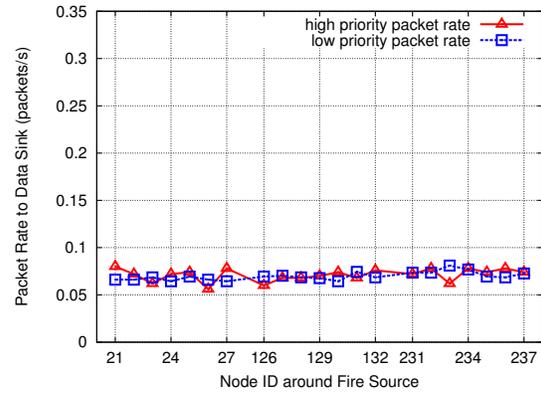
V. CONCLUSION

In this paper, we introduced a prioritized transmission protocol called *Pritrans* that differentiates the priority of packets by computing the consensus relation from the information they carry. In a data collection tree, each child node is assigned a relative rank based on the packets transmitted. The parent nodes periodically broadcast feedback messages to adaptively adjust the packet rate of the subtree nodes. A preemptive queuing mechanism is applied at each node such that an incoming packet with more important information arriving at a full packet queue can evict the low priority ones. This allows more data to be retrieved from the critical locations with higher packet delivery ratio.

The current consensus relation function allocates equal weight to each attribute in the collected data. For certain applications, some attributes may retain greater significance than the rest. The importance of the data packets should be computed from both the values and the associated weighting

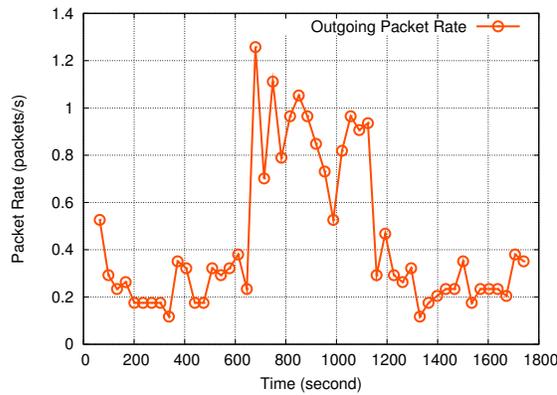


(a) Packet Rate vs. Node ID with Pritrans

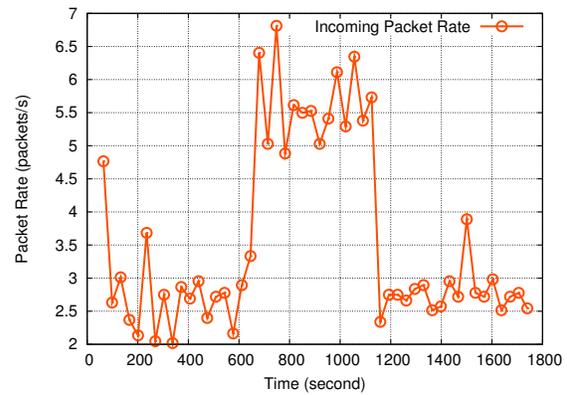


(b) Packet Rate vs. Node ID without Pritrans

Fig. 2. Packet Rate from Hot Nodes at Data Sink with and without Pritrans



(a) Outgoing Packet Rate of Leaf Node 129 over Time



(b) Incoming Packet Rate of Cluster Head 9 over Time

Fig. 3. Packet Delivery Performance of Pritrans at Node 129 and Node 9

factors. In the future work, we will explore the possibility of using alternate methods to estimate the global order of nodes in the network.

ACKNOWLEDGEMENTS

The work has been supported in part by the grant of the Eastern Europe Research Scientists & Students (EERSS) Exchange & Collaboration Program of the National University of Singapore.

REFERENCES

- [1] R. Szweczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *Proc. of SenSys*, Baltimore, MD, USA, 2004, pp. 214–226.
- [2] H. Song, S. Zhu, and G. Cao, "Svats: A sensor-network-based vehicle anti-theft system," in *Proc. of Infocom*, Phoenix, AZ, USA, 2008, pp. 171–175.
- [3] D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. of MobiCom*, San Diego, CA, USA, 2003, pp. 134–146.
- [4] S. Muravyov, M. C. Chan, and M. Khomyakova, "Prioritizing sensed data transmission by consensus relation in wireless sensor network," in *Proc. of the 12th IMEKO TC1 & TC7 Joint Symposium on Man Science & Measurement*, Annecy, France, 2008, pp. 277–282.
- [5] G. Hackmann, O. Chipara, and C. Lu, "Robust topology control for indoor wireless sensor networks," in *Proc. of SenSys*, Raleigh, NC, USA, 2008, pp. 57–70.
- [6] J. Newsome and D. Song, "GEM: Graph embedding for routing and data-centric storage in sensor networks without geographic information," in *Proc. of SenSys*, Los Angeles, CA, USA, 2003, pp. 76–88.
- [7] A. Woo and D. Culler, "A transmission control scheme for media access in sensor networks," in *Proc. of MobiCom*, Rome, Italy, 2001, pp. 221–235.
- [8] J. Paek and R. Govindan, "Rcrt: Rate-controlled reliable transport for wireless sensor networks," in *Proc. of SenSys*, Sydney, Australia, 2007, pp. 305–319.
- [9] S. Kim, R. Fonseca, P. Dutta, A. T. D. Culler, P. Levis, S. Shenker, and I. Stoica, "Flush: A reliable bulk transport protocol for multihop wireless networks," in *Proc. of SenSys*, Sydney, Australia, 2007, pp. 351–365.
- [10] H. Zhang, A. Arora, Y. Choi, and M. Gouda, "Reliable bursty convergecast in wireless sensor networks," in *Proc. of MobiHoc*, Urbana-Champaign, IL, USA, 2005, pp. 266–276.
- [11] H. P. Young, "Optimal voting rules," *Journal of Economic Perspectives*, vol. 9, no. 1, pp. 51–64, 1995.
- [12] A. Sen, "The possibility of social choice," *American Economic Review*, vol. 89, no. 3, pp. 349–378, 1999.
- [13] S. Muravyov, "Rankings as ordinal scale measurement results," *Metrolology and Measurement Systems*, vol. 13, no. 1, pp. 9–24, 2007.
- [14] J. P. Barthélemy, A. Guenoche, and O. Hudry, "Median linear orders: Heuristics and a branch and bound algorithm," *European Journal of Operational Research*, vol. 42, no. 3, pp. 313–325, 1989.
- [15] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szweczyk, A. Woo, E. Brewer, and D. Culler, "The emergence of networking abstractions and techniques in tinyos," in *Proc. of NSDI*, San Francisco, CA, USA, 2004, pp. 1–14.