ELSEVIER

CrossMark

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 37 (2014) 181 – 188

Procedia
Computer Science

The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2014)

Interference-Aware Congestion Control Protocol for Wireless Sensor Networks

Mohamed Amine Kafi[a,b,*], Djamel Djenouri[b], Jalel Ben Othman[c], Abdelraouf Ouadjaout[a,b], Miloud Bagaa[a,b], Noureddine Lasla[a,b], Nadjib Badache[a,b]

[a]*CERIST Research Center, Ben Aknoun, Algiers 16000, Algeria*
[b]*University of Sciences and Technology Houari-Boumediene (USTHB), Algiers, Algeria*
[c]*Laboratoire L2TI. Universite de Paris 13, Paris, France*

Abstract

This paper deals with congestion and interference control in wireless sensor networks (WSN), which is essential for improving the throughput and saving the scarce energy in networks where nodes have different capacities and traffic patterns. A scheme called *IACC* (*Interference-Aware Congestion Control*) is proposed. It allows maximizing link capacity utilization for each node by controlling congestion and interference. This is achieved through fair maximum rate control of interfering nodes in inter and intra paths of hot spots. The proposed protocol has been evaluated by simulation, where the results rival the effectiveness of our scheme in terms of energy saving and throughput. In particular, the results demonstrate the protocol scalability and considerable reduction of packet loss that allow to achieve as high packet delivery ratio as 80% for large networks.

Peer-review under responsibility of the Program Chairs of EUSPN-2014 and ICTH 2014.
Keywords: Wireless sensor networks; transport protocols; congestion control; fairness; interference; link capacity.

1. Introduction

In a wireless sensor network, sensors gather information about the environment and notify the base-station. Applications may require the notification to be continuous, periodic, or on event occurrence [1]. In some event-based (resp. continuous) applications, nodes may transmit significant volumes of data towards the sink upon event occurrence (resp. permanently), e.g., video tracking, surveillance applications. As sensors share the same wireless channel, contention on the available bandwidth is inevitable. In a real environment, the packet collision caused by links interference or packets loss due to congestion on inter-paths and intra-path nodes dramatically affects the application throughput and causes high energy consumption. Congestion control consists of two parts: i) the congestion and interference detection, and ii) a rate control mechanism establishment, which adjusts the reporting rate. Different

* Corresponding author. Tel.: +213-553-195-197 ; fax:+213-21-91-21-26
E-mail address: kafiamine@gmail.com

Peer-review under responsibility of the Program Chairs of EUSPN-2014 and ICTH 2014.
doi:10.1016/j.procs.2014.08.028

metrics are used in literature for detecting congestion, such as buffer length, packet inter arrival time, packet service time, channel load, etc[1].

The existing approaches, in their mitigating or avoiding forms, underuse the throughput that the network capacity can offer to the application. The mechanisms based on rate regulation between nodes use AIMD (Additive increase multiplicative decrease) methods to balance the offered capacity, but ignore nodes interference, the principal cause of loss. On the other hand, scheduling based methods do not take into account physical-link-capacity differences between nodes. In this paper, we propose a scheduling scheme that takes into account dynamic link interference and capacity. It provides efficient and fair rate partitioning over congested links. The proposed protocol identifies the hot spot points by measuring locally, at every node, the interfering neighbor links that cannot be simultaneously active in the whole schedule. Finally, the rate of each link and the number of slots given to each node is determined by taking into account the depth of the node and its link capacity.

The paper is organized as follows. Section 2 presents the related works on congestion control and avoidance. In Section 3, the problem is formulated with a thorough network model. The proposed protocol *IACC* (Interference-Aware Congestion Control) is presented in Section 4. Section 5 shows the simulation results. Finally, Section 6 concludes the paper.

2. Related Work

ESRT[2] (Event-to-Sink Reliable Transport) is a centralized rate allocation control protocol where the sink controls nodes rate in event driven-applications. Nodes are assumed to be located at one hop from the sink. ESRT defines event reliability as the number of data packets required for reliable event detection from the whole set of nodes, while minimizing energy consumption. Each node sends at a fixed rate until receiving rate update instruction from the sink. The sink uses the perceived goodput and congestion level (based on the buffer length) to update rates. This method ensures fairness, but not efficiency of throughput.

CODA[3] (Congestion Detection and Avoidance) is a mitigation rate control protocol, where each node detects congestion using both channel and buffer loads. The node controls its rate in an AIMD manner. CODA considers two strategies: i) open-loop back pressure for transient congestion, where the concerned node broadcasts the message to its neighbours that further propagate these messages to upstream nodes, depending on their buffer occupancy, and ii) an end-to-end acknowledgement-based approach (also named closed-loop for persistent congestion). However, CODA does not focus on per-source fairness.

In CCF[4], each node uses packet transmission duration to estimate the channel capacity. It then divides this capacity between its sub-tree nodes. A major drawback of CCF is that the remaining capacity from no-sending nodes goes unused.

In ARC[5], rate adjustment is done through an AIMD, which is proportional to the number of descendant nodes. The congestion is passively detected when a node finds that its parent does not forward its traffic. ARC tries avoiding interference by introducing a jitter before sending. This allows desynchronizing neighbours.
QCRA[6] attempts to determine optimal and fair sources transmission rate at the sink using information about topology, link loss rates, and communication pattern. Its heuristic results to a coarse-grained TDMA schedule between neighbours using CSMA. Loss rate is used to assign nodes' sending rates. QCRA decisions are periodic with epochs at the order of few tens of minutes.

MCCP[7] uses successive data and schedule intervals. During data interval, nodes send data using a schedule received from their next hop nodes. As one packet can be sent at a slot, slot length represents the reporting rate. With short length slots, the rate is increased. During the schedule intervals, nodes generate the schedule for the next data interval. Packet delivery time and buffer size are used to attribute time slots. But slot attribution in this scheme does not specify any contention avoidance.

TARA (Topology-Aware Resource Adaptation)[8] uses resource control to ensure application fidelity. It defines the link congestion sum as the sum of link's traffic and interfering links' traffic. TARA uses graph-coloring approach for capacity estimation. It defines the topology interference degree and constructs the spatial interference graph.

In Flush[9], large data is divided into packets and sent in a pipelined transmission. The sink schedules transfers to avoid inter-flows interference. It uses end-to-end ACKs to ensure reliability, and hop-by-hop rate control. Flush dynamically chooses the sending rate using bandwidth measurements and interference information to avoid intra-path interference, which depends on the interference range at each node. Flush is very restrictive as only one source at a time can transmit. It does not describe the scheduling heuristic but just the capacity sharing part. CADT[10] protocol is similar to FLUSH but with many flows. CADT applies AIMD rate control using immediate downstream nodes buffer and the concerned node link state.

In[11], authors proposes a TDMA schedule to ensure maximum throughput and fair rate allocation by taking into account the requirement imposed on network lifetime. The authors use lexicographic Max-Min to formulate the rate allocation along with fairness, maximum throughput, and slots reuse in the purpose to reach a minimum frame length.

3. Network Model and Problem Formulation

We assume that all nodes have synchronized clocks and each node has a single half-duplex radio transceiver. Communication between nodes is modeled by an undirected graph $G = (V, E)$, where V is the set of nodes, with a base-station $B \in V$, and E is the edges set of G. Based on the observation that the links between nodes do not have the same capacity, every edge $(u, v) \in E$ is supposed having a specified capacity $C_{u,v}$ to transmit packets from node u to node v.

The data are gathered over a tree structure $G_T = (V, E_T)$ rooted at B, such that $E_T \subseteq E$ represents the tree edges. The problem that we tackle in this paper is how to gather the data, without collisions and congestions, from a set of source nodes called V_S (i.e., $V_S \subseteq V$) through the tree structure, while maximizing the network throughput. All source nodes are supposed to continually generate data. Time is divided into contiguous frames, where each frame is constituted of a set of slots $\{\sigma_1, \sigma_2, \ldots \sigma_l\}$ having the same duration δ. During each slot σ_i, the data packets are generated or transferred from a set of children nodes to their parents. $\mathbb{S}(u)$ denotes the set of slots used by a node $u \in V$ to generate data, receive the data from its children, or transmit to its parent w. $\tau_\sigma(u)$, $\lambda_\sigma(u)$, $\mu_\sigma(u)$ denote the rates of data generation, reception, and transmission during a slot $\sigma \in \mathbb{S}(u)$, respectively.

The *Maximum Throughput Congestion and Collision Prevention* (TCCP) problem is how to find the appropriate slot distribution $\mathbb{S}$ along with the appropriate functions $\tau_\sigma(u)$, $\lambda_\sigma(u)$, $\mu_\sigma(u)$ for every node $u \in V$ and every slot $\sigma \in \mathbb{S}(u)$, such that the throughput is maximized, and collisions and congestions are prevented. An instance of the *TCCP* problem must satisfy the following conditions:

1. The data are generated only by the source nodes, i.e., $\forall u \notin V_S, \forall \sigma \in \mathbb{S}(u) : \tau_\sigma(u) = 0$.
2. The rate of data transmission from a node u to its parent w (*i.e.*, $(u, w) \in E_T$) should not exceed the capacity $C_{u,w}$. Formally, $\forall \sigma \in \mathbb{S}(u) : \mu_\sigma(u) \leq C_{u,w}$.
3. Each node has a half-duplex radio transceiver and then it cannot transmit and receive simultaneously: $\forall \sigma \in \mathbb{S}(u) : \lambda_\sigma(u)\mu_\sigma(u) = 0$.
4. *Collision-free constraint*: Two nodes u and v cannot transmit at the same time slot σ in one of the following cases:

   - Node u is parent of v or vice versa;
   - Nodes u and v have different parents, say w and z, respectively. A collision occurs *iff* w or z is within the transmission range of v or u, respectively.

   In these two cases, to prevent the collision between nodes, u and v, the following *collision-free constraint* should be verified:

$$\forall \sigma \in \mathbb{S}(u) \cap \mathbb{S}(v) : \mu_\sigma(u)\mu_\sigma(v) = 0$$

5. *Congestion-free constraint*: To prevent the congestion, we should verify that the amount of the transmitted data should be greater than the amount of the received and generated data. That is, the *congestion-free constraint* can be expressed as follows:

$$\sum_{\sigma \in \mathbb{S}(u)} \mu_\sigma(u) \geq \sum_{\sigma \in \mathbb{S}(u)} (\lambda_\sigma(u) + \tau_\sigma(u))$$

4. Protocol Description

In order to facilitate the presentation of *IACC*, Fig.1 will be referenced throughout the discussion, as an illustrative example. In the figure, a solid arrow between two nodes, a and b, indicates that b is the parent of a in the communication tree. The dotted lines represent the graph connectivity, i.e., the presence of a communication link between a pair of nodes. The number besides the solid arrow (a, b) in Fig.1(b) represents a's physical link capacity on the communication tree. The number besides the solid arrow (a, b) in Fig.1(c) represents the number of packets required to be forwarded by node, a, to ensure the fairness. Whereas, the number besides the solid arrow (a, b) in Fig.1(d) represents the number of slots required for node, a, to ensure the fairness. The number besides the solid arrow (a, b) in Fig.1(f) represents the assigned slots for each node in the network. All the network nodes in Fig.1 are considered as sources. In what follows, the communication tree is supposed to be constructed using some routing protocol, such as [12], which serves as input to the proposed solution.

IACC consists of two steps aiming to establish an appropriate schedule that takes into account the real network capabilities. Firstly, the link capacity between each node and its parent is estimated. The estimated capacities are forwarded from the network nodes to the base-station. Secondly, the scheduling is established to ensure the fairness while preventing congestions and collisions in the network. The estimated capacities help *IACC* to establish a schedule that allocate for every node the appropriate time to transmit, and the appropriate rate.

4.1. Link Quality Estimation

The congestion control in *IACC* is based on real measurements of radio links capacity. To obtain an estimation of these capacities, every node, u, assesses radio link relating it to its parent, w, by sending a burst of packets during a determined amount of time, T. Every packet is sent after the reception of the previous packet's acknowledgement, or after a time-out from the former submission. The link capacity, $C_{u,w}$, is then approximated by the total number of acknowledged packets divided by the period T. Upon completion of this phase, node u sends to the base station the estimated link capacity, as well as the ID of its parent. This message is forwarded via the tree structure. This phase and the previous one are depicted in Fig.1(b).

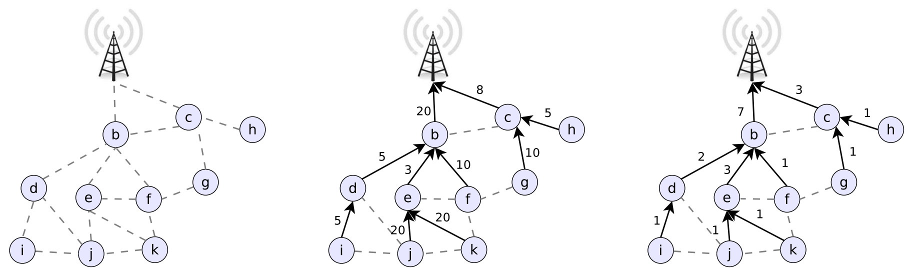
4.2. Schedule Construction

This step is executed by the base station after collecting link capacity messages on the communication tree. The process of rate scheduling is then divided into two sub-phases: *Rate Distribution* and *Slots Assignments*.

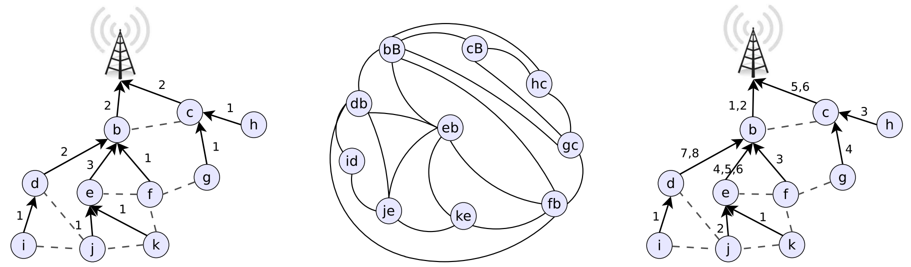
4.2.1. Rate Distribution

For each node u, the aim is to distribute its available bandwidth such that:

- all source nodes under the responsibility of node u are enabled to fairly send their data.
- congestion is avoided by computing an appropriate sending rate for u. This rate does not exceed the capacity of the router nodes present in the path towards the base station.



(a) Network topology (b) Physical capacities (c) Required packets during one frame



(d) Number of required slots per frame (e) Interference graph (f) slots assignment

Fig. 1. Illustrative example to show how *IACC* is applied

To guarantee the fairness condition, the base station computes the number of source nodes in the underneath sub-tree of u, which is denoted by η_u. This is shown in Fig.1(c). Therefore, node u should be allowed to send at least η_u packets during a single frame. Since the frame is divided into slots of duration δ, node u will be assigned at least $s_u = \left\lceil \frac{\eta_u}{\delta C_{u,w}} \right\rceil$ slots, where w is the parent of u. This is shown in Fig.1(d).

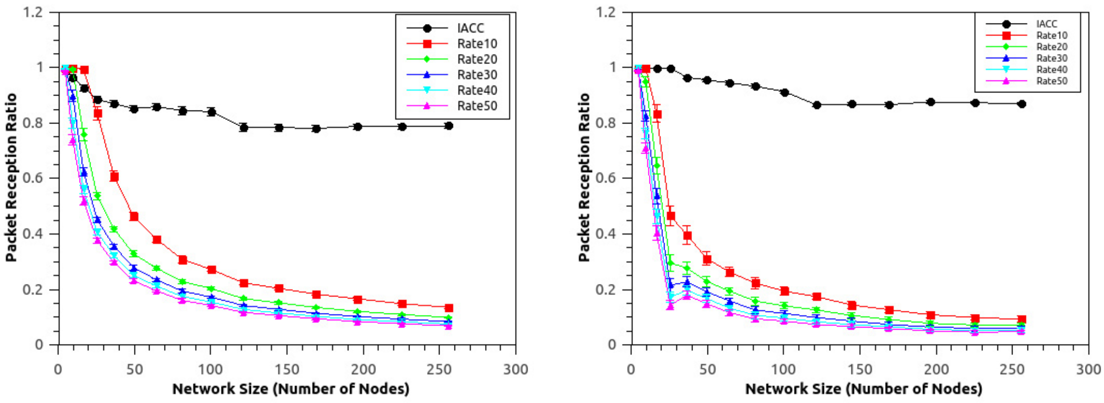
When the remainder of the euclidean division $\frac{\eta_u}{\delta C_{u,w}}$ is not null, the last slot will not be fully used. To enhance efficiency, the remaining capacity of all these slots is divided fairly between the nodes in the sub-tree that have these remaining capacities.

4.2.2. Slots Assignments

After computing the appropriate needs of each node in terms of slots per frame, *IACC* starts the slot assignment process. This process aims at orchestrating the starting time of each slot so that to avoid collisions and let the non interfering links to transmit simultaneously. An interference graph $G_I = (V_I, E_I)$ is first built. For the previous example, the corresponding interference graph is shown in Fig.1(e). The set of vertices $V_I = E_T$ corresponds to the set of radio links between nodes and their parents in the tree structure. Two links, say (i, j) and (k, l) are connected in E_I if they are in interference with each other.

The algorithm of slot assignment is based on an iterative process. In every iteration n, *IACC* searches in, G_I, for the maximal independent set, say I_n, which contains the vertices with the higher number of edges. This is in the purpose to cut the interference graph into many sub-graphs that allow fast reuse of slots. Since the radio links in I_n are not interfering, the set of nodes $\{u \mid (u, v) \in I_n\}$ can transmit in slot n. When a slot is assigned to a node u, its required number of slots per frame is decreased. When this number becomes null, the vertices relating node, u, with its corresponding edges in, G_I, are removed. The iterations stop when the graph G_I becomes empty.

The presented algorithm ensures fairness while it avoids congestion and collisions. The resulted slots assignment is shown through the example of Fig.1(f). However, in some situations, it does not achieve an optimal efficiency and links can be underused. Indeed, the nodes that have consumed their required number of slots (removed from G_I) can take extra slots. This is provided if they do not create interference with links in I_n for the next steps. So in every



(a) Network Density=5

(b) Network Density=25

Fig. 2. Packet Reception Ratio

step, all the removed links from G_I are compared to the links in I_n. These new added slots for these links are marked separately. After emptying the G_I, all the added slots are kept if the direct upstream link has an added slot, starting from the root's children. To ensure congestion control, the capacity of the added slots is divided fairly between the concerned links.

5. Protocol Evaluation

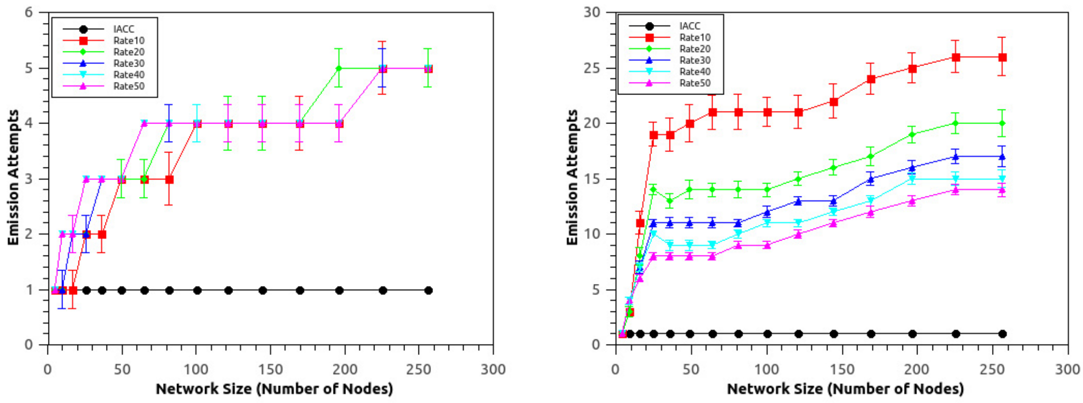
In this section *IACC* is evaluated by simulation using TOSSIM[13]. The number of nodes was varied up to 256. Before running the network simulator, a random connected graph with an expected node degree $d \in \{5, 25\}$ has been generated for every number of nodes. Similarly to[14], an edge between two nodes, u and v, is stochastically selected with a probability: $P = \frac{d}{N}$. This is called edge probability. Further, a random capacity is assigned to each edge in the communication graph. In TOSSIM, this is done by defining a "gain" value between two communicating nodes, which corresponds to the reception power measured in dBm. After extensive experiments, we have observed that for every "gain" value of the link between two nodes, it corresponds a reception probability. Therefore the link probability is obtained by sending packets burst between two neighboring nodes and evaluating the average packet reception ratio, as shown in Section 4.1.

We have considered an application that periodically generate 500 packets, with a 400 seconds pause time. Each packet contains 29 bytes in its payload. This is a typical scenario for monitoring applications that use periodic traffic sampling (snapshot of the monitored zone in a video surveillance application). All nodes in the network act as sources for traffic generation, i.e. $V_S = V \setminus \{B\}$. The simulation stops when every node successfully transmits 2000 packets. Each generated packet is put in a waiting queue, from which the transport protocol picks up a packet from the tail.

We have compared the performance of *IACC* with a baseline approach that uses a fixed rate algorithm in which packets are consumed periodically from the queue with a predefined constant rate. They are transmitted with best effort towards the base station. In the experiments, these rates have been varied from $10 pkt/s$ to $50 pkt/s$. Each experiment is repeated 33 times and the results are presented with 95% confidence interval. The following metrics were evaluated: the packet reception ratio, the number of retransmissions and the average throughput.

5.1. Packet Reception Ratio

The average Packet Reception Ratio is the ratio between packets successfully received at sink, to the total number of packets generated by the sources. This metric reflects the impact of interference and congestion on the packets



(a) Network Density=5

(b) Network Density=25

Fig. 3. Emission attempts

goodput. In Fig.2(*a*) and 2(*b*), the packet reception ratio in the proposed protocol (IACC) outperforms the base line approach for almost all the defined rates. This is because IACC assigns the appropriate rate for every node while avoiding interferences, whereas the base line approach applies the same rate for every node in the network without considering interferences.

5.2. Number of Retransmissions

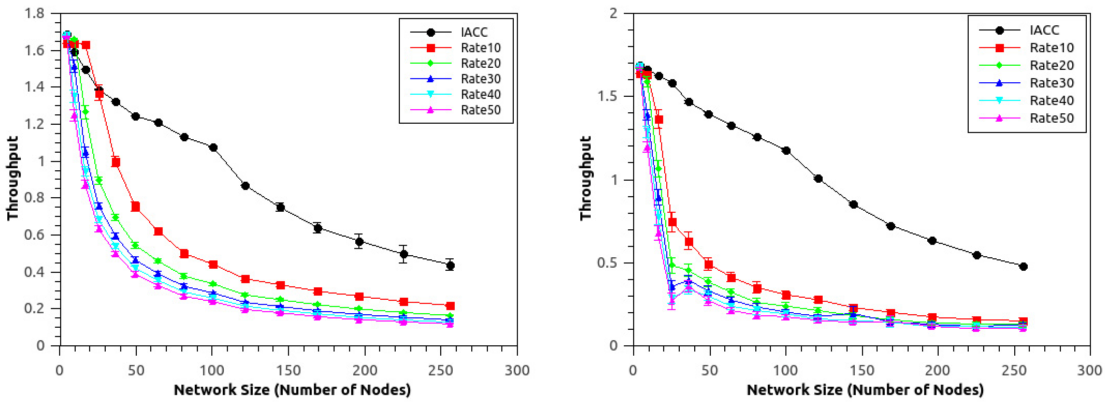
It is notable that the more the number of retransmissions per packet increases, the more energy will be consumed. This dramatically affects the network lifetime. In Fig.3(*a*) and 3(*b*), the average energy consumed for packet transmission is presented by the average sending attempts. The average number of retransmission in *IACC* is too close from the optimal number (one transmission). This can be justified by the control applied by this protocol, which ensures packets delivery with minimum loss even in a dense network. However, in the fixed rate protocols the average number of retransmission increases with the number of nodes. In low network density, the number of retransmission exceeds 5, where in denser network, this number exceeds 25. This is due to the number of interferences that can occur, especially in dense networks.

5.3. Throughput

The average throughput at the sink is the number of packets received during a time unit. The fixed rate protocols try to send their packets as soon as they are queued. We remarked that the first transmitted packets are received within a short delay, then the remaining are just lost. Therefore, using uniform traffic generation with the traditional definition of the throughput (that does not capture packet loss) will lead to biased performance results in favour of this lossy protocols. This justifies the interwining transmission and idle periods in our scenarios. IACC shows a smooth degradation when the number of nodes grows, as depicted in figures 4(*a*) and 4(*b*).

6. Conclusion and Future Work

In dense high rate wireless sensor networks using CSMA based communications, nodes contend for a shared medium. In addition, links suffer from capacity variation during network lifetime due to interference. This causes higher buffer occupancy and hence packet losses on intermediate nodes, in the presence of network congestion. A novel data transport mechanism that addresses the link variation and network congestion has been proposed in this



(a) Network Density=5 (b) Network Density=25

Fig. 4. Throughput

paper. The proposed protocol integrates an efficient capacity based, intra-path and inter-paths congestion and interference aware scheduling scheme. Extensive simulation has shown its efficiency in terms of throughput, packet delivery ratio and energy (packet emission attempts) compared to different fixed rates sending for different network densities.

As future work, we plan to extend the proposed congestion control scheduling scheme with a scheme that ensures reliability and packet loss recovery. As another enhancement, we plan to consider the congestion and interference mitigating in order to overcome transient congestion situation caused by buffer loss or link bad quality. Reaction to persistent change in interference and capacity parameters, that may happen during application lifetime, is also in our perspectives. Proper selection of a threshold that reflects significant change of the network state may be useful. This will enable on-demand scheduling invocation rather than periodic invocation[6]. A deeper comparison with state of the art protocols such[8,9], and a real testbed experiment are also in our agenda.

References

1. Kafi, M.A., Djenouri, D., Ben-Othman, J., Badache, N.. Congestion Control Protocols in Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials, accepted for publication* 2014;.
2. Zgr, Y.S., Sankarasubramaniam, Y., Akan, Ö.B., Akyildiz, I.F.. ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks. In: *Proc. 4th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc*. 2003, p. 177–188.
3. Wan, C.Y., Eisenman, S.B., Campbell, A.T.. Energy-efficient congestion detection and avoidance in sensor networks. *ACM Trans Sen Netw* 2011;**7**(4):1–31.
4. Ee, C.T., Bajcsy, R.. Congestion control and fairness for many-to-one routing in sensor networks. In: *ACM, SenSys*. 2004, p. 148–161.
5. Woo, A., Culler, D.E.. A Transmission Control Scheme for Media Access in Sensor Networks. In: *Proc. of ACM Mobicom*. 2001, p. 221–235.
6. Bian, F., Rangwala, S., Govindan, R.. Quasi-static Centralized Rate Allocation for Sensor Networks. In: *Proc. of 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON*. 2007, p. 361–370.
7. Hussain, F.B., Cebi, Y., Shah, G.A.. A multievent congestion control protocol for wireless sensor networks. *EURASIP J Wirel Commun Netw* 2008;:44:1–44:12.
8. Kang, J., Zhang, Y., Nath, B.. TARA: Topology-Aware Resource Adaptation to Alleviate Congestion in Sensor Networks. *IEEE Trans Parallel Distrib Syst* 2007;**18**(7):919–931.
9. Kim, S., Fonseca, R., Dutta, P., Tavakoli, A., Culler, D., Levis, P., et al. Flush: a reliable bulk transport protocol for multihop wireless networks. In: *Proc. of the 5th international conference on Embedded networked sensor systems, SenSys*. 2007, p. 351–365.
10. Rahman, M., Monowar, M., Hong, C.. A Capacity Aware Data Transport Protocol for Wireless Sensor Network. In: *Proc. of the International Conference on Computational Science and Its Applications, ICCSA*; Lecture Notes in Computer Science. 2009, p. 491–502.
11. Yao, M., Lin, C., Zhang, P., Tian, Y., Xu, S.. TDMA Scheduling with Maximum Throughput and Fair Rate Allocation in Wireless Sensor Networks. In: *IEEE ICC 2013 - Ad-hoc and Sensor Networking Symposium*. 2013, .
12. Gnawali, O., Fonseca, R., Jamieson, K., Moss, D., Levis, P.. Collection Tree Protocol. In: *Sensys*. 2009, .
13. , H.L., , A.C., Levis, P.. Improving Wireless Simulation Through Noise Modeling . In: *IPSN*. 2007, .
14. Chung, F., Lu, L.. Connected components in random graphs with given expected degree sequences. *Ann Combinatorics* 2002;**6**:125–145.