On Real-Time Capacity of Event-driven Data-gathering Sensor Networks

Bo Jiang, Binoy Ravindran
Department of Electrical and Computer Engineering
Virginia Polytechnic Institute & State University
Blacksburg, VA 24061
Email: {bjiang,binoy}@vt.edu

Hyeonjoong Cho
Department of Computer and Information Science
Korea University
Seoul, Korea 136-701
Email: raycho@korea.ac.kr

Abstract—**Network capacity is a critical feature of wireless ad hoc and sensor networks. It is particularly challenging to determine network capacity when combined with other performance objectives such as timeliness. This paper investigates real-time capacity for event-driven data-gathering sensor networks with the unbalanced many-to-one traffic pattern. First, we compute the average allowable throughputs of nodes for a given event distribution, based on which we then leverage results of queuing theory to estimate the per-hop delays. We develop a new slack time distribution scheme for the unbalanced many-to-one traffic pattern, and prove it optimal in terms of the per-hop success probability. Here the per-hop success probability is defined as the probability for a packet to meet its sub-deadlines at each hop. Finally, we define the network-wide real-time capacity, i.e., given a threshold for the per-hop success probability, how much data (in bit per second) can be delivered to the sink node meeting their deadlines. For these research results, we provide some application scenarios, including configuring packet deadlines or verifying a specific deadline configuration, setting a packet's priority for dynamic scheduling, and trading the reliability of real-time data delivery for capacity efficiency etc. We also study two special cases of WSNs, the chain model and the continuous model. Our slack distribution scheme yields consistent or similar results for these two special cases as that of past works, but is more adaptive by supporting more generic cases.**

I. Introduction

Wireless sensor networks (or WSNs), which are composed of a large number of multi-functional low-cost sensor nodes, are increasingly being used for collecting data from a geographical region of interest and reporting them back to sink nodes (or base stations). Typical applications of WSNs include surveillance, environment monitoring, medical diagnostics, transportation management, and industrial control [1]. Nodes in sensor networks are typically capable of sensing, processing, and communicating data. But at the same time, their power supply, and computing and communicating capabilities are strictly constrained. This leads to one of the most fundamental problems of WSNs, the capacity—i.e., how much data a WSN can collect and report, when given specific constraints on power, real-time, reliability etc. Knowledge of the network capacity can help application designers design a sensor network for the problems at hand.

Network capacity is important not just for sensor networks. In [2], Li *et. al.* present evaluation results for wireless ad hoc networks, showing that the actually available per-node throughput is dozens of times smaller than the apparent radio capacity. They also point out that, this is because the interference among nodes may exceed the range at which they can communicate successfully. Another cause is the overheads of MAC protocols such as headers, RTS/CTS exchange, and ACK packets. The capacity of wireless networks, in general, has been well studied since the landmark work of Gupta and Kumar [3]. Except for the interference and overheads introduced in [2], network size, traffic pattern, and network architecture are also among the most commonly studied factors that constrain the achievable capacity. (We discuss related work in details in Section II.)

In this paper, we study the real-time capacity for event-driven data-gathering sensor networks with the unbalanced many-to-one traffic pattern:

1) We define *real-time capacity* as the network's capability to transmit time-sensitive data within deadlines. Deadlines for data come from the specific configuration of the applications, e.g., for timely response of actors [4], or for timely target tracking [5] etc.

2) We discuss *event-driven* WSNs, a major category of sensor networks in terms of the data delivery mode, in which sensor nodes collect and report data only if an event of interest occurs. (Other categories include continuous, observer-initiated, and hybrid [6].) Event-driven sensor networks are typically used for intrusion detection or sensing/acting systems. Most of the previous research efforts on network capacity of WSNs have focused on the continuous mode, where sensors periodically report thereby produce balanced traffic loads. Compared with the continuous mode, the event-driven mode introduces more uncertainties, produces unbalanced traffic loads, thereby is more complex to analyze.

3) We focus on the *many-to-one* traffic pattern, which is caused by the data gathering structure of WSNs. Unlike ad hoc networks, sensor networks collect data from a large number of sensor nodes and report them to a much smaller number of sink nodes. Usually nodes propagate their measurements to the closest sink node. Without loss of generality, we consider a many-to-one data gathering network with only one sink node.

We study the real-time capacity in a per-hop manner. This is because with unbalanced traffic loads, describing network-wide conditions from a macro view may not guarantee that a

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6813
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6813

packet can meet all of its sub-deadlines at multiple hops.

First, we compute the average allowable outgoing/incoming throughputs of nodes based on a given event distribution, and then estimate the expected per-hop delays. We consider the waiting time that a packet may incur in the queues as the major source of per-hop delays, thereby the major consumer for the slack time. Thus we leverage results from queuing theory to estimate it. By simulating nodes that have the same hop distance from the sink node as a queue, we utilize $M/M/1$ and $D/D/1$ queuing models to estimate the expected per-hop delays based on the computed throughputs.

Secondly, we present a new slack time distribution, and estimate the probability of delivering a packet to the sink within its deadline. By subtracting the necessary time from the end-to-end deadline of a packet, we obtain the end-to-end slack time for which the packet can be delayed at the most. Previous work used either a uniform model or an exponential model [7] for partitioning the end-to-end slack time among multiple hops. However, both of them are completely based on mathematical models without considering much of the realistic network requirements. We design a new slack distribution scheme, which is closer to the realistic network than these two models. Also we introduce the concept of the per-hop success probability, which is computed by multiplying the probabilities of meeting the sub-deadlines at each hop. In terms of the per-hop success probability, we prove that the new slack time distribution scheme is optimal.

Finally, we define the real-time capacity of the network as the amount of real-time data that the network can deliver within the deadline with a lower-bounded, per-hop success probability. For both the slack time distribution scheme and the concept of real-time capacity, we provide some application scenarios, including configuring packet deadlines or verifying a given deadline configuration, setting a packet's priority for dynamic scheduling, and trading the reliability of real-time data delivery for capacity efficiency etc.

In addition, we study two special cases of network models, i.e., the chain model and the continuous model. In the chain model, there is only one source node as well as one sink node, with a single routing path connecting them. On the contrary, each node in the network periodically produces an equal amount of traffic load for the continuous model. We show that our slack distribution scheme yields consistent or similar results for these two special cases as the past works, but is more adaptive by supporting more generic cases.

The paper makes the following contributions:

1) We establish queuing models and leverage results from queuing theory to estimate the expected per-hop delays of event-driven sensor networks with the unbalanced many-to-one traffic pattern.

2) We design a new slack time distribution scheme based on the requirements of realistic networks, and prove it optimal in terms of the per-hop success probability. Based on this per-hop success probability, we also define the real-time capacity of WSNs.

3) We provide some potential applications of the research results, including configuring packet deadlines or verifying a given network-wide deadline configuration, setting a packet's priority for dynamic scheduling, and trading the reliability of real-time data delivery for capacity efficiency etc.

The rest of this paper is organized as follows. In Section II, we introduce the related work. In Section III, we describe our assumptions, models and notations. In Section IV we compute the allowable throughputs based on a given event distribution, and leverage results of queuing theory to estimate the expected per-hop delays. In Section V, we present the slack time distribution scheme and the concept of per-hop success probability, for which we provide two application examples on deadline configuration and dynamic priority scheduling. In Section VI, we define the concept of real-time capacity, and discuss some application scenarios such as capacity efficiency etc. Section VII finally concludes the paper and discusses our future work.

II. Related Work

The capacity of wireless ad hoc networks has been well studied in the past several years since the work of Gupta and Kumar [3]. In [3], the authors estimate the achievable per-node throughput for a static ad hoc network as $\Theta(\frac{1}{\sqrt{n \log n}})$, and further show that the per-node throughput cannot exceed $\Theta(\frac{1}{\sqrt{n}})$ even if nodes are optimally deployed and the transmission range is optimally chosen. Their work is based on a one-to-one balanced traffic pattern, where each node generates equal amount of traffic loads and destinations are randomly chosen. Based on [3], many further results were developed. In [8], Grossglauser *et. al.* show that the capacity can be further increased if node mobility is introduced to reduce the path length between the source node and the destination, but with unbounded delay as a cost. In [9], Bansal *et. al.* improve the work of [8] by providing low delay guarantee. Gastpar *et. al.* study a one-to-one traffic pattern in [10], where there is only one source-destination pair and all the other nodes work as relays. Li *et. al.* present the capacity analysis for some typical topologies in [2]. Beyond these typical but simple network architectures, Liu *et. al.* describe the throughput capacity of hybrid wireless networks with sparse base stations connected via a high-bandwidth wired network in [11]. In [12], Jain *et. al.* model the interference using a conflict graph and study the maximum throughput with any given network and workload specified as inputs. Most of these previous research efforts on the capacity of wireless ad hoc networks only consider continuous, balanced traffic loads.

Based on the research for wireless ad hoc networks, network capacity has also been studied for wireless sensor networks. A major difference of WSNs from ad hoc networks is that WSNs emphasize the many-to-one traffic pattern [13], [14], as they are often used to collect and report data to a small number of sink nodes. In addition, due to the resource constraints, their network capacity is often discussed together with deployment [15], network architecture [16]–[18], data aggregation or in-network computation [19], [20].

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6813
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6813

One of the earliest efforts to study real-time capacity for sensor networks is [21]. Before that, delay is only considered as a constraint for the capacity [9], [22], and they do not address the question of how much real-time data a network can transmit. In [21], Abdelzaher *et. al.* present real-time capacity analysis for sensor networks with balanced loads and continuous convergecast traffic. They leverage their preceding work on real-time scheduling that specifies utilization bounds [23], and assume time-independent, fixed-priority scheduling policy for the analysis. However, [21] does not consider event-driven sensor networks with unbalanced traffic loads, and it implicitly assumes that a packet's status keeps unchanged during the propagation. The definition of their synthetic utilization depends only on the packet size and its end-to-end deadline, while these two factors remain unchanged for a specific packet during its entire lifetime. However, packets often experience variable message velocities (i.e., the speed of propagation) at multiple hops due to the congestion around sink nodes caused by the convergecast traffic. The utilization that packets impose on nodes therefore should be a function of their ever-changing status, e.g., the distance from the sink, the remaining time to the deadline etc.

In contrast with [21], our work focuses on the slack time distribution scheme and real-time capacity of event-driven sensor networks with the completely unbalanced traffic pattern. As far as the authors know, this has not been studied in the past. We discuss the throughput, slack times, and the capacity in a per-hop manner, which makes it possible to optimize the probability of meeting a packet's deadline based on its ever-changing status.

III. Preliminaries

In this section, we introduce the assumptions, models (for the network, the event, the throughput, the deadline and the slack time) and summarize all the notations used in the following discussion.

A. Assumptions

We make the following assumptions.

- *Network architecture.* We assume a flat network architecture consisting of a large number of homogeneous sensor nodes and one sink node. These nodes are equipped with omni-directional antennas, and are deployed randomly and uniformly in a disk on the plane. In addition, we assume that the only sink node is in the center of the disk field. Except that, there is no other cluster headers or any relay nodes that compose a hierarchical architecture.
- *Transmission range.* We assume that the transition power of sensor nodes is fixed, and thereby the transmission range, denoted as r, is also fixed.
- *Connectivity.* We do not make any assumptions on node location, network topology, or link quality. But we assume that the transmission range can provide connectivity.
- *Communication model.* We adopt the protocol model introduced in [3], where both transmission and interference depend only on the Euclidean distance between nodes. The protocol model does not consider information theoretic techniques such as network coding, interference cancelation, superposition coding, and coherent combining [19]. Therefore, we simplify the problem by avoiding the complexity of information theory and the physical model. And we assume that there is only one wireless channel. For wireless communication, we assume that nodes communicate with each other through packets, and concurrent transmissions around a receiver may collide.
- *Traffic pattern.* Our work mainly focuses on the unbalanced traffic pattern in a many-to-one convergecast sensor network. In an event-driven sensor network, sensor nodes report data towards the sink only when an event of interest occurs. Since events of interest are assumed to occur randomly, it is highly probable that some nodes generate data while others do not, resulting in unbalanced workloads. Also we assume that the event distribution is static, i.e., how many events may occur and how they occur keep unchanged.
- *MAC protocol.* We assume the underlying MAC protocol is perfect regarding to the transmission, i.e., the traffic transmitted by nodes $h+1$ hops away from the sink will be able to arrive at nodes h hops away from the sink successfully without being dropped, as long as the total transmission throughput of hop $h+1$ is less than or equal to the acceptable receiving throughput of hop h.

B. Models

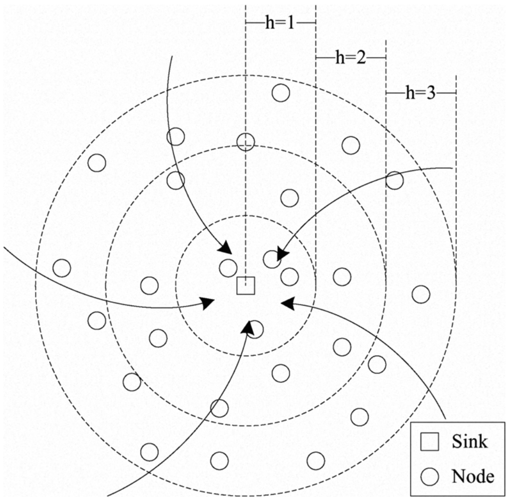

Fig. 1. Network Architecture

1) Network: Based on the assumptions that we discussed, Figure 1 shows the network architecture in a disk on the plane. The square at the center stands for the sink node, small circles represent sensor nodes, and the large dotted circles show the distance of nodes from the sink. We count the hop number starting from the sink node. In other words, we say that all

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6813
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6813

the nodes in the ring between $(h-1)r$ and hr belong to *hop layer h*. We let H_{max} denote the maximum hop number, and N_h denote the number of nodes that are located in hop layer h. In the figure, the curved arrows signify the transmission direction of the convergecast data flows.

In this layered architecture, nodes in hop layer $h > 1$ cannot communicate with the sink directly. Instead, the packets that they generate have to be relayed by nodes in the inner layers. Furthermore, we assume that nodes do not communicate with neighbors in the same hop layer. Instead they receive packets only from neighbors in the outer layer and relay them only to neighbors in the inner layer.

With the random and uniform deployment, such a layered architecture for multi-hop transmission establishes a quantitative relation among the node amounts of different layers. Many previous works such as [21], [24] leverage this advantage to simplify the analysis. Letting ρ denote the node density of the sensor network, the average number of nodes in hop layer h can be computed as $N_h = \rho(\pi(hr)^2 - \pi[(h-1)r]^2) = \rho\pi r^2(2h-1)$. In fact, the constant $\rho\pi r^2$ is just the number of nodes in an area within a node's transmission range, i.e., N_1. Thus $N_h = N_1(2h-1)$. When $h = 0$, there is only one sink node, thus $N_0 = 1$.

2) Event: For event-driven sensor networks, we establish an event distribution model, which describes how many events may occur and how they occur. Let E_h denote the average traffic (in bit per second) that each node in hop layer h may produce after observing events. E_h is a function of hop layer h, and represents a distribution of events in the network. For example, a constant E_h for $\forall\ h$ signifies the continuous sensor network model, where each node periodically produces an equal amount of traffic load. For a border surveillance model, where events only occur on the border of the network, E_h is given by:

$$E_h = \begin{cases} constant & , \quad (h = H_{max}) \\ 0 & , \quad (h < H_{max}) \end{cases}$$

In fact, E_h should be a random process $E_h(t)$, as the occurrence of events changes with time. However, we assume that the event distribution is static for simplification. We let $\hat{E} = \{E_1, E_2, \cdots, E_{H_{max}}\}$ denote an arbitrary event distribution model.

3) Throughput: For a node in hop layer h, we let C_h denote its average allowable *outgoing throughput*, and C'_h denote its average allowable *incoming throughput*. As the outgoing throughput is composed of the traffic that a node produces and that it relays, we have

$$C_h \leq E_h + C'_h \tag{1}$$

The inequality is strict when the total traffic of $E_h + C'_h$ exceeds the allowed outgoing capacity of nodes.

For all the nodes in hop layer h, the total outgoing throughput is N_hC_h, and the total incoming throughput is $N_hC'_h$. Based on the assumption of a perfect MAC protocol, we have

$$N_{h+1}C_{h+1} = N_hC'_h \tag{2}$$

i.e., the traffic transmitted by nodes in hop layer $h+1$ can arrive at nodes in hop layer h successfully without being dropped. And it is obvious that

$$N_iC_i \leq N_jC_j \ (for\ \forall\ i > j) \tag{3}$$

as the total schedulable traffic produced by nodes in the outer layers cannot exceed the traffic that can be consumed by nodes in the inner layers. Then combining Equation (2) and Equation (3), we have $N_{h+1}C_{h+1} = N_hC'_h \leq N_hC_h$, therefore $C'_h \leq C_h$. Many factors may constrain this inequality as a strict one, one typical example among which is that nodes in hop h may produce traffic themselves, i.e., $E_h > 0$.

Let W denote the fixed transmission capacity of the wireless channel, i.e., each node can transmit or receive data at $W\ bit/s$ at the most. We have $N_1C_1 \leq W$, i.e., the receiving capacity of the sink cannot exceed the receiving capacity of its wireless channel. Based on the assumption of a perfect MAC protocol, we tighten this inequality as $N_1C_1 = W$.

4) Deadline and Slack Time: The slack time of a time-constrained activity denotes the redundant time that is available for the completion of that activity before the expiration of the time constraint. For real-time data delivery in WSNs, the end-to-end *slack time* is the redundant time after subtracting the time necessary, e.g., for transmission and propagation, from the deadline. In fact, the necessary time here is just the optimal delay of the multi-hop propagation when everything is in a perfect condition. For example, if a packet has a deadline of 2000 ms, but the end-to-end necessary time is 1500 ms which is impossible to avoid or reduce, then the slack time is $2000 - 1500 = 500$ ms. Nodes along the propagation path may safely utilize this slack time for queuing, routing, and other operations on the packet without jeopardizing the packet deadline.

Let $D_{e2e,H}$ denote the end-to-end deadline of a packet generated in hop layer H, and τ denote the average per-hop necessary time. Then $L_{e2e,H} = D_{e2e,H} - H\tau$ is the end-to-end slack time, and $L_{h,H}$ is the slack time spared for a node in hop layer h to relay this packet.

C. Notations

For the convenience of discussion, we summarize all the notations in Table I.

Most of the notations in the table have been introduced. Only α, $T_{h,H}$, P_{e2e} and $RTC(\beta)$ will be defined in the following sections.

IV. Per-hop Delay Estimation

First we compute the average allowable outgoing/incoming throughputs of nodes (C_h and C'_h) based on a given event distribution ($\hat{E}$). Then we simulate nodes in hop layer h as a queue. Based on results of queuing theory and the average allowable throughputs of nodes, we estimate the per-hop delay by which a packet consumes its end-to-end slack time.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6813
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6813

TABLE I
NOTATIONS

Parameter	Definition
r	Transmission range of nodes
H_{max}	Maximum hop number
h	Index of hop layers ($h \in [1, H_{max}]$)
N_h	Number of nodes in hop layer h
ρ	Node density
E_h	Average traffic in bit/s that each node in hop layer h may produce after observing events
$\hat{E}$	An arbitrary event distribution
C_h	Average allowable outgoing throughput of a node in hop layer h
C'_h	Average allowable incoming throughput of a node in hop layer h
W	Fixed transmission capacity of the wireless channel
α	Rate control factor
$T_{h,H}$	Expected waiting time of a packet in the queue of hop layer h
$D_{e2e,H}$	End-to-end deadline for a packet generated in hop layer H
τ	Average per-hop time that is necessary for propagation
$L_{e2e,H}$	End-to-end slack time for a packet generated in hop layer H
$L_{h,H}$	Slack time spared for a node in hop layer h
P_{e2e}	Per-hop success probability
$RTC(\beta)$	Real-time capacity

Lemma 1 (Throughputs): The average allowable outgoing throughput of a node in hop layer h is:

$$C_h = \frac{1}{N_h}(W - \alpha \sum_{i=1}^{h-1} N_i E_i) \tag{4}$$

The average allowable incoming throughput of a node in hop layer h is:

$$C'_h = \frac{1}{N_h}(W - \alpha \sum_{i=1}^{h} N_i E_i) \tag{5}$$

where $\alpha = \frac{W}{\sum_{i=1}^{H_{max}} N_i E_i}$.

Proof:

The total traffic that nodes produce after observing events, i.e., $\sum_{i=1}^{H_{max}} N_i E_i$, may exceed the receiving capacity W of the sink node. Therefore, we need to control the sending rate of source nodes. For the fairness of rate control, a good approach is to cut by their weights. Then the upper bound of the traffic load that a node in hop layer h may generate is:

$$\frac{W}{\sum_{i=1}^{H_{max}} N_i E_i} \cdot E_h$$

where $\frac{W}{\sum_{i=1}^{H_{max}} N_i E_i}$ is a *rate control factor*, which we define as α.

After controlling the sending rate of nodes, the total traffic load in the network is less than or equal to the capacity of the wireless channel. Nodes will not drop the packets due to lacking of enough outgoing capacity. Thus Equation (1) can be rewritten as $C_h = \alpha E_h + C'_h = \alpha E_h + \frac{N_{h+1}}{N_h} C_{h+1}$. Through iteration, we have

$$C_1 = \frac{\alpha}{N_1} \sum_{i=1}^{h-1} N_i E_i + \frac{N_h}{N_1} C_h$$

Since $N_1 C_1 = W$, C_h can be computed as:

$$C_h = \frac{1}{N_h}(W - \alpha \sum_{i=1}^{h-1} N_i E_i)$$

Substituting this result into Equation (2), we have

$$C'_h = \frac{1}{N_h}(W - \alpha \sum_{i=1}^{h} N_i E_i)$$

■

Based on the results of Lemma 1, we then analyze the expected per-hop delay by establishing queue models. We suppose that in the network layer of a node, packets that come up from the MAC layer (to be relayed) and those that are passed down from the application (to be sent) are served on a first-come, first-served basis.

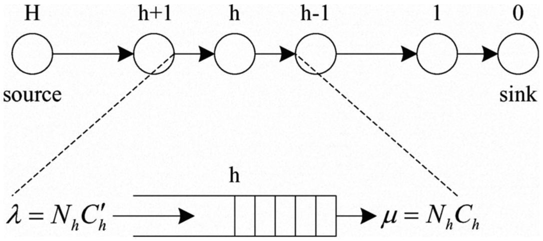

Fig. 2. Propagation Path and Virtual Queue

Figure 2 shows the propagation path of a packet from the source that is H hops away from the sink. Numbers above the node circles represent the hop numbers. We consider all the nodes in hop layer h as a queue, so that results of queuing theory could be leveraged for the analysis. Then the whole network can be considered as a chain of queues, each for a hop layer. For the queue representing all the nodes in hop layer h, the mean arrival rate λ is the total incoming throughput $N_h C'_h$, and the mean service rate μ is the total outgoing throughput $N_h C_h$, as shown in Figure 2.

We utilize two queue models for delay estimation, regarding to whether or not nodes produce their own traffic: 1) when nodes in hop layer h produce their own traffic (i.e., $C_h > C'_h$), we approximate them with a $M/M/1$ queue; and 2) when nodes in hop layer h do not produce their own traffic (i.e., $C_h = C'_h$), we approximate them with a $D/D/1$ queue[1].

Both $M/M/1$ queue and $D/D/1$ queue are simple queue models. Usually $M/M/1$ queue is closer to the actual network environment than $D/D/1$ queue [26]. However, a constraint of $M/M/1$ queue is that it is stable only if the service rate

[1] Based on the Kendall notation, $M/M/1$ is a queue with Poisson arrival process, exponential service time, one server, and infinite buffer. $D/D/1$ is a queue with deterministic arrival rate and service time, one server, and infinite buffer [25].

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6813
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6813

μ is greater than the arrival rate λ (i.e., $C_h > C'_h$). Queuing theory shows that only with this constraint, the queue will not keep growing, and the expected waiting time will not approach infinity. For the $C_h = C'_h$ case, the expected waiting time of $M/M/1$ queue will approach infinity and therefore is useless for delay analysis. Thus, we adopt the feasible $D/D/1$ queue model for this case, which is not as accurate as $M/M/1$ queue though.

From queuing theory, the waiting time of a packet in a queue is deterministic for $D/D/1$ queue and exponentially distributed for $M/M/1$ queue [25]. Then the expected waiting time of a packet in hop layer h (i.e., the deterministic delay of $D/D/1$ queue or the mean of the exponentially distributed delay of $M/M/1$ queue) is:

$$T_{h,H} = \begin{cases} \frac{1}{\mu-\lambda} = \frac{1}{N_h C_h - N_h C'_h} & , \quad (C_h > C'_h) \\ \frac{1}{\mu} = \frac{1}{N_h C_h} & , \quad (C_h = C'_h) \end{cases}$$

where H is the hop layer in which the packet is generated.

Substituting Equations (4) and (5) into it, the expected per-hop waiting time can be expressed as:

$$T_{h,H} = \begin{cases} \frac{1}{\alpha N_h E_h} & , \quad (C_h > C'_h) \\ \frac{1}{\alpha \sum_{i=h}^{H_{max}} N_i E_i} & , \quad (C_h = C'_h) \end{cases} \tag{6}$$

Then the expected end-to-end waiting time is:

$$T_{e2e,H} = \sum_{h=1}^{H} T_{h,H} \tag{7}$$

Equations (6) and (7) show the distribution of the expected delays that a packet is likely to experience, which serves as the foundation for distributing the end-to-end slack time to multi-hops as needed.

V. Slack Time Distribution

How the end-to-end slack time is distributed to multi-hops embodies how a system considers network congestion and message velocity. The slack distribution scheme may be different in different network environments.

The chain model and the continuous model are two typical special cases of event-driven sensor networks with the many-to-one traffic pattern. First, a many-to-one sensor network will degenerate to a chain architecture when only one source node generates traffic and propagates it via a single route. Along the chain path, there is no congestion, thus the packet velocity does not need to change. Secondly, recall that each node in a continuous WSN produces an equal amount of original traffic. Thus a continuous sensor network is a special case of event-driven sensor networks with extremely average traffic loads.

As previously mentioned, most existing works adopt either a uniform distribution or an exponential distribution for partitioning the end-to-end slack time, which works well for the chain model and the continuous model respectively. Uniform distribution allocates the total end-to-end slack time evenly to all the hops from the source to the sink, implicitly assuming that a packet suffers the same delay at each intermediate node along the path. In [7], Liu *et. al.* present an exponential model, where the per-hop slack time is computed as $L_h = \frac{L_{e2e}}{2^h}$, where L is the slack time and h is the number of hops from the sink node. Obviously, uniform distribution is exactly suitable for the chain model, while exponential distribution signifies the strictly monotonous increase of traffic loads in the continuous network model, as the hop number decreases. However, both of them are completely based on mathematical models without considering much of the realistic requirements of generic event-driven sensor networks with the unbalanced traffic pattern.

In a generic sensor network with the unbalanced many-to-one traffic pattern, the traffic load of nodes gets more and more as the hop distance from the sink decreases [27]. Consequently, the message velocity will decrease as a packet approaches the sink. Such an ever-changing message velocity requires variable slack times at each hop, i.e., nodes that are closer to the sink deserve more slack time as packets suffer more delay. At the same time, the increase of traffic loads may not be strictly monotonous, as it is possible that some relay nodes do not produce traffic with a given event distribution. Therefore, a generic slack distribution should hold a similar trend of monotonous increase as the exponential model, but the increase may not be strict.

A. Distribution Model

We design a new model to distribute the end-to-end slack time to multi-hops as needed, based on a specific event distribution and the corresponding delays estimated in Section IV. It can meet the requirements of realistic networks instead of simply establishing a mathematical model.

In details, we distribute the end-to-end slack time $L_{e2e,H}$ in a proportional manner to the expected delays determined by Equation (6). Such a distributed per-hop slack time can be actually considered as a sub-deadline for that hop.

Definition 1 (Per-hop Slack Time): If the throughputs of all the hop layers along a packet's routing path are known, the per-hop slack time for hop layer h is given as:

$$L_{h,H} = \frac{T_{h,H}}{T_{e2e,H}} \cdot L_{e2e,H}$$

Next we define a probability of meeting each of the sub-deadlines to verify the advantage of this slack time distribution scheme. As the waiting time of $D/D/1$ queue is less than that of $M/M/1$ queue [25], the probability that a packet experiences less time than a given sub-deadline in $D/D/1$ queue will be higher than that in $M/M/1$ queue. Therefore, we utilize the probability in $M/M/1$ queue as the unified lower bound. As the waiting time of a packet in a $M/M/1$ queue is exponentially distributed, the probability that a packet experiences less time than the distributed per-hop slack time $L_{h,H}$ can be expressed as:

$$P[t \leq L_{h,H}] = 1 - e^{-\frac{L_{h,H}}{T_{h,H}}} = 1 - e^{-\frac{L_{e2e,H}}{T_{e2e,H}}} \tag{8}$$

From Equation (8), we observe that by the definition of this slack distribution scheme, the probability that a packet meets the per-hop sub-deadline is independent of the hop number h.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6813
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6813

Definition 2 (Per-hop Success Probability): Given the deadline $D_{e2e,H}$ and a slack time distribution $\{L_{h,H} | 1 \leq h \leq H\}$ for a packet generated in hop layer H, the per-hop success probability is defined as the probability that all the sub-deadlines at H hops are satisfied:

$$\begin{aligned} & P_{e2e}(D_{e2e,H}, H) \\ & = P[t_1 \leq L_{1,H},\ \cdots,\ t_h \leq L_{h,H},\ \cdots,\ t_H \leq L_{H,H}] \\ & = P[t_1 \leq L_{1,H}] \cdots P[t_h \leq L_{h,H}] \cdots P[t_H \leq L_{H,H}] \\ & = (1 - e^{-\frac{L_{e2e,H}}{T_{e2e,H}}})^H = (1 - e^{-\frac{L_{e2e,H}}{\sum_{h=1}^{H} T_{h,H}}})^H \end{aligned} \tag{9}$$

where t_h is the time actually spent on a node in hop layer h.

$P_{e2e}(D_{e2e,H}, H)$ is a function of a packet's route length and its deadline in a given network environment with a specific event distribution.

Theorem 1 (Optimal Slack Distribution Theorem): Based on the relation between the event distribution and the expected per-hop delays, the slack distribution scheme determined by Definition 1 is optimal in terms of achieving the highest per-hop success probability.

Proof:

Without loss of generality, we consider the case of two hops. Thus

$$P[t_1 \leq L_{1,H},\ t_2 \leq L_{2,H}] = (1 - e^{-\frac{L_{e2e,H}}{T_{e2e,H}}})^2$$

Now, suppose that a different slack distribution scheme distributes the end-to-end slack time to these two hops with $L_{1,H} + \delta$ and $L_{2,H} - \delta$ ($\forall\ \delta > 0$), instead of $L_{1,H}$ and $L_{2,H}$. Then the per-hop success probability will become:

$$\begin{aligned} & P[t_1 \leq L_{1,H} + \delta,\ t_2 \leq L_{2,H} - \delta] \\ & = (1 - e^{-\frac{L_{e2e,H}}{T_{e2e,H}}} \cdot e^{-\frac{\delta}{T_{1,H}}}) \cdot (1 - e^{-\frac{L_{e2e,H}}{T_{e2e,H}}} \cdot e^{\frac{\delta}{T_{2,H}}}) \end{aligned} \tag{10}$$

We substitute the complicated expressions in Equation (10) with constants for simplification. Let $Y = e^{-\frac{L_{e2e,H}}{T_{e2e,H}}}$, $A = e^{-\frac{\delta}{T_{1,H}}}$, and $B = e^{\frac{\delta}{T_{2,H}}}$. Then we can tell $0 < Y < 1$ and $0 < A < 1 < B$. Now, what we need to prove is simply $(1 - AY)(1 - BY) < (1 - Y)^2 \Leftrightarrow (AB - 1)Y < A + B - 2$. This inequality can be proved as:

$$\begin{aligned} Left &= (AB - 1)Y < AB - 1 \\ &= (A - 1)(B - 1) + A + B - 2 \\ &< A + B - 2 = Right \end{aligned}$$

This proof process can be easily extended to the case of multiple hops. As δ is arbitrary, the slack distribution scheme in Definition 1 is guaranteed to be optimal in terms of the per-hop success probability.

■

Finally we provide the computational complexity for distributing the slack time and computing the per-hop success probability. Given an event distribution $\hat{E}$ and a network size H_{max}, the complexity for computing $T_{e2e,H}$ with Equation (7) is $O(H_{max}^2)$. Thus, the complexity for computing the per-hop slack time $L_{h,H}$ with Definition 1, and the complexity for computing the per-hop success probability $P_{e2e}(D_{e2e,H}, H)$ with Definition 2 are also $O(H_{max}^2)$.

B. Application Examples

As long as the event distribution is given, the new slack time distribution scheme can be generally applied to all the cases of event-driven sensor networks with the many-to-one traffic pattern, including those special cases for which uniform distribution and exponential distribution used to work well. Now we first study the chain model and the continuous model, and show how the new slack time distribution can be applied. From the study, we will find that the new slack distribution scheme yields consistent or similar results as that of the past works. In addition, our results for event-driven sensor networks are more generic by taking the event distribution into account. Then we present two examples on applying the slack time distribution to configuring the end-to-end deadline of a packet, and adjusting the priority of packets for dynamic priority scheduling.

1) Chain Model: When only one node in the network produces traffic, the many-to-one convergecast architecture will degenerate to a chain model, in which a single path connects and relays traffic loads from the source node to the sink node. Assume that the source node is located in hop layer H. For each hop layer $h \in [1, H-1] \cup [H+1, H_{max}]$, nodes do not produce their own traffic, i.e., $E_h = 0$ and $C_h = C'_h$. The only traffic source is shown as $E_H > 0$. Thus, the expected delay for $\forall\ h \in [1, H-1]$ is:

$$T_{h,H} = \frac{1}{\alpha \sum_{i=h}^{H_{max}} N_i E_i} = \frac{\sum_{i=1}^{H_{max}} N_i E_i}{W \sum_{i=h}^{H_{max}} N_i E_i} = \frac{N_H E_H}{W N_H E_H} = \frac{1}{W}$$

For the hop layer H, the source node produces its own traffic. Thus,

$$T_{H,H} = \frac{1}{\alpha N_H E_H} = \frac{\sum_{i=1}^{H_{max}} N_i E_i}{W N_H E_H} = \frac{N_H E_H}{W N_H E_H} = \frac{1}{W}$$

Therefore,

$$T_{e2e,H} = \sum_{h=1}^{H} T_{h,H} = \sum_{h=1}^{H} \frac{1}{W} = \frac{H}{W}$$

So based on Definition 1, the per-hop slack time can be distributed as $L_{h,H} = \frac{L_{e2e,H}}{H}$, which is independent of the hop number. This is consistent with the result that a uniform distribution yields.

The per-hop success probability then is:

$$P_{e2e}(D_{e2e,H}, H) = (1 - e^{-\frac{L_{e2e,H}}{T_{e2e,H}}})^H = (1 - e^{-\frac{W L_{e2e,H}}{H}})^H$$

2) Continuous Model: In a continuous sensor network, all the nodes produce the same traffic load. Therefore for $\forall\ h$, we have $C_h > C'_h$ and $E_h = E$ (a constant). Assume that packets produced by a node located in hop layer H are configured with an end-to-end deadline $D_{e2e,H}$. Then, the expected delay for $\forall\ h \in [1, H]$ is:

$$T_{h,H} = \frac{1}{\alpha N_h E_h} = \frac{\sum_{i=1}^{H_{max}} N_i E_i}{W N_h E_h} = \frac{\sum_{i=1}^{H_{max}} N_i}{W N_h}$$

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6813
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6813

The end-to-end expected delay is:

$$T_{e2e,H} = \sum_{h=1}^{H} T_{h,H} = \frac{\sum_{i=1}^{H_{max}} N_i}{W} \sum_{h=1}^{H} \frac{1}{N_h}$$

Thus, the per-hop slack time is:

$$L_{h,H} = \frac{\frac{1}{N_h}}{\sum_{h=1}^{H} \frac{1}{N_h}} L_{e2e,H}$$

We may observe that the closer a hop layer is to the sink, the smaller N_h is, and the larger their $L_{h,H}$ will be. This is consistent with the actual requirement that nodes closer to the sink deserve more slack time in many-to-one sensor networks. Furthermore, when nodes are deployed randomly and uniformly, this result yields strictly monotonously increasing slack times as the hop number decreases, just as what the exponential distribution looks like. But compared with the exponential distribution scheme, the new slack distribution scheme is optimal in terms of the per-hop success probability, thereby more accurate for actual network requirements.

3) Deadline Configuration: An important application of the new slack time distribution and the per-hop success probability concept is to configure the end-to-end deadline for a packet. In the past, most of research efforts simply assume a given deadline for packets without discussing how this deadline is computed or configured. In contrast, here, we provide an approach for configuring the packet deadline based on the requirement for the delivery probability.

For a packet generated in hop layer H, we can compute the lower bound of its deadline, so that the per-hop success probability can be achieved to be greater than a given requirement threshold β. Formally, this is computed as:

$$\begin{aligned}
&P_{e2e}(D_{e2e,H}, H) = (1 - e^{-\frac{L_{e2e,H}}{T_{e2e,H}}})^{H} \geq \beta \\
\Longleftrightarrow\ & e^{-\frac{L_{e2e,H}}{T_{e2e,H}}} \leq 1 - \beta^{\frac{1}{H}} \\
\Longleftrightarrow\ & L_{e2e,H} = D_{e2e,H} - H\tau \geq -T_{e2e,H} \ln(1 - \beta^{\frac{1}{H}}) \\
\Longleftrightarrow\ & D_{e2e,H} \geq H\tau - T_{e2e,H} \ln(1 - \beta^{\frac{1}{H}})
\end{aligned}$$

4) Dynamic Priority Scheduling: This slack distribution scheme can also be used for setting a packet's priority based on the computed per-hop sub-deadline.

Each time a packet produced in hop layer H proceeds for a hop, e.g. from hop layer H to hop layer $H-1$, it may be considered as a different packet, which is initially produced in hop layer $H-1$ instead of hop layer H. And the end-to-end deadline $D_{e2e,H-1}$ is just the remaining time to the original deadline $D_{e2e,H}$. Note that here we are using the second subscript to differentiate parameters for the original packet from those for the virtual one, e.g., $D_{e2e,H-1}$ means the end-to-end deadline of the virtual packet that we suppose is produced in hop layer $H-1$. If at hop H it consumes more or less slack time than the expected $L_{H,H}$, its remaining end-to-end slack time $L_{e2e,H-1}$ as well as the distribution has to be recalculated, according to the original end-to-end deadline and the time that actually elapses.

For example, we suppose that at hop H the packet consumes a fixed necessary time τ and δ more slack time than the distributed $L_{H,H}$, then the remaining end-to-end slack time $L_{e2e,H-1} = L_{e2e,H} - L_{H,H} - \delta$. We recalculate the slack time distributed to hop layer $H-1$ as:

$$\begin{aligned}
L_{H-1,H-1} &= \frac{T_{H-1,H-1}}{T_{e2e,H-1}} \cdot L_{e2e,H-1} \\
&= \frac{T_{H-1,H}}{T_{e2e,H} - T_{H,H}} \cdot \left(L_{e2e,H} - L_{H,H} - \delta\right) \\
&= \frac{L_{H-1,H}}{L_{e2e,H} - L_{H,H}} \cdot \left(L_{e2e,H} - L_{H,H} - \delta\right) \\
&= L_{H-1,H} \cdot \left(1 - \frac{\delta}{L_{e2e,H} - L_{H,H}}\right)
\end{aligned} \tag{11}$$

Based on the actual slack time that elapses in the previous hop, Equation (11) shows by how much percentage the distributed slack time should change for the current hop. Accordingly, the priority of the packet may need to be changed to guarantee to meet this new sub-deadline. This builds a foundation for dynamic priority scheduling.

VI. Real-time Capacity

We now define the concept of real-time capacity. Also we study some examples for potential applications of this real-time capacity.

A. Definition

In the previous research efforts, the capacity of wireless ad hoc and sensor networks is usually defined as a function of the network size. The upper or lower bounds of the network capacity are often the research focus [3], [13]. Delay has been studied usually as a constraint for the capacity [9]. In [21], the real-time capacity is defined as the upper bound of the sum of all the in-transit packets' ratios of the packet size divided by their respective deadline, i.e., $C_{RT} = W \sum \frac{C_i}{D_i}$, where C_i and D_i are the transmission time and the deadline of packet i, respectively. However, [21] has used this definition as a network-wide term without considering per-hop details.

Sometimes network-wide conditions cannot provide sufficient guarantee especially for event-driven sensor networks with the unbalanced traffic pattern. For example, two nodes each with 100 bit/s capacity cannot simultaneously guarantee the timely delivery of a data flow requiring 20 bit/s capacity and another requiring 150 bit/s capacity.

In this paper, we define real-time capacity based on the per-hop success probability given by Definition 2, i.e., the probability that a packet meets the per-hop sub-deadlines at all the hops. Let β denote a threshold for the per-hop success probability that must be satisfied. We define real-time capacity as follows.

Definition 3 (Real-Time Capacity): The real-time capacity of an event-driven data-gathering sensor network is defined as the sum of traffic loads, the per-hop success probability of

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6813
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6813

which is higher than a given requirement threshold β, i.e.,

$$RTC(\beta) = \alpha \cdot \sum_{P_{e2e}(D_{e2e,i},i) \geq \beta}^{1 \leq i \leq H_{max}} N_i E_i \tag{12}$$

As in Lemma 1, $\alpha = \frac{W}{\sum_{i=1}^{H_{max}} N_i E_i}$ is the rate control factor.

This real-time capacity defines how much real-time data (in bit per second) the network can deliver before their deadlines with a per-hop success probability that is greater than the given threshold β, for an arbitrary event distribution $\hat{E}$. Based on the computational complexity for calculating the per-hop success probability, the complexity for computing the real-time capacity $RTC(\beta)$ is $O(H_{max}^3)$.

B. Application Examples

As in Section V, we also provide some examples for the potential application of the real-time capacity concept. In fact, the computation of real-time capacity itself is an important application, which can be utilized as the foundation for network configuration and rate control. Besides this, we introduce two more applications.

1) Capacity Efficiency: The efficiency of the network capacity is an important metric for estimating the performance of network optimization [28]. It is defined as the percentage of the utilized capacity in the total network capacity. Based on Definition 3, the efficiency of real-time capacity, denoted as $CE(\beta)$, can be expressed as:

$$CE(\beta) = \frac{RTC(\beta)}{W} = \frac{\sum_{P_{e2e}(D_{e2e,i},i) \geq \beta}^{1 \leq i \leq H_{max}} N_i E_i}{\sum_{i=1}^{H_{max}} N_i E_i}$$

$CE(\beta)$ is a function of the requirement threshold β. When β increases but the event distribution and the deadline configuration keep unchanged, the traffic load that can achieve the required probability of timely delivery will decrease, accordingly $CE(\beta)$ will decrease. With the distribution of $CE(\beta)$ over β, we would be able to either find out the tradeoff point, where both the capacity efficiency and the reliability of real-time data delivery can satisfy their respective constraints, or otherwise draw a negative conclusion.

2) Deadline Configuration Verification: In Section V, we provided an application example of the slack time distribution to configure the deadline for a specific packet stream generated in a specific hop layer H. Now we could utilize the concept of real-time capacity to verify whether or not a network-wide deadline configuration, denoted as $\hat{D} = \{D_{e2e,1}, D_{e2e,2}, \cdots, D_{e2e,H_{max}}\}$, can meet the requirement of the timely delivery probability.

In Definition 3, real-time capacity is a function of both β and a network-wide deadline configuration $\hat{D}$ (implicitly). Consequently, β is a function of both real-time capacity RTC and $\hat{D}$, i.e.,

$$\beta = F(RTC, \hat{D})$$

For a given network-wide deadline configuration $\hat{D}$ as well as a real-time capacity distribution over β, we can find out whether or not the required threshold β can be satisfied, i.e., whether or not real-time data delivery can be probabilistically guaranteed.

This provides another approach for checking the configuration of the packet deadlines by a specific application, other than the one discussed in Section V.

VII. Conclusions & Future Work

This paper investigates the real-time capacity of event-driven data-gathering sensor networks with the unbalanced many-to-one traffic pattern. We establish the relation between the event distribution and the average allowable throughputs of nodes. Then we leverage $M/M/1$ and $D/D/1$ models of queuing theory to estimate the expected per-hop delays, so as to distribute the end-to-end slack time proportionally to these expected delays. We introduce the concept of per-hop success probability to estimate the probability that a packet can reach the sink node meeting all of its sub-deadlines. In terms of the per-hop success probability, we verify that the new slack time distribution scheme is optimal. Finally, we define the real-time capacity of the network. Some examples are given for the potential application of the research results of this paper. In addition, we study two special cases of network models, the chain model and the continuous model. Compared with past works, our slack distribution scheme yields consistent or similar results for these two special cases as that of past works, but is more adaptive by supporting more generic cases.

The future work includes but not limited to:

- *Dynamic event distribution.* Instead of considering static event distribution only, the slack time distribution would be more accurate and more adaptive if dynamic event distribution is considered with random processes. A typical example that requires the support for dynamic event distribution is mobile target tracking and predicting.
- *Further applications.* In this paper, we discuss several application examples of the slack time distribution scheme and real-time capacity. Many more potential applications can be explored, e.g., for rate control. Based on real-time capacity, the transmission rate of source nodes can be monitored to make the best use of network resources, but avoid any unnecessary congestions and delays at the same time.
- *Multiple sink nodes.* In realistic sensor networks, designers usually deploy multiple sinks to improve the network performance such as the end-to-end delay. The real-time capacity of sensor networks with multiple sink nodes needs to be revised based on the results of this paper.

Acknowledgment

This work was supported by research project grants from MKE/MND, South Korea.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks (Amsterdam, Netherlands: 1999)*, vol. 38, no. 4, pp. 393–422, 2002.

[2] J. Li, C. Blake, D. S. J. De, C. Hu, I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *MobiCom*, 2001, pp. 61–69.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6813
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6813

[3] P. Gupta and P. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, 2000.

[4] V. C. Gungor, O. B. Akan, and I. F. Akyildiz, "A real-time and reliable transport (rt)2 protocol for wireless sensor and actor networks," in *IEEE/ACM Transactions on Networking*, 2008.

[5] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*, 2004, pp. 129–143.

[6] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 2, pp. 28–36, 2002.

[7] K. Liu, N. Abu-Ghazaleh, and K.-D. Kang, "Jits: just-in-time scheduling for real-time sensor data dissemination," *Fourth Annual IEEE International Conference on Pervasive Computing and Communications*, pp. 5–46, 2006.

[8] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 477–486, 2002.

[9] N. Bansal and Z. Liu, "Capacity, delay and mobility in wireless ad-hoc networks," *INFOCOM*, vol. 2, pp. 1553–1563, 2003.

[10] M. Gastpar and M. Vetterli, "On the capacity of wireless networks: the relay case," *INFOCOM*, vol. 3, pp. 1577–1586, 2002.

[11] B. Liu, Z. Liu, and D. Towsley, "On the capacity of hybrid wireless networks," *INFOCOM*, vol. 2, pp. 1543–1552, 2003.

[12] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, 2003, pp. 66–80.

[13] D. Marco, E. Duarte-Melo, M. Liu, and D. Neuhoff, "On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data," *Proc. International Workshop on Information Processing in Sensor Networks*, 2003.

[14] H. Gamal, "On the scaling laws of dense wireless sensor networks: the data gathering channel," *Information Theory, IEEE Transactions on*, vol. 51, no. 3, pp. 1229–1234, 2005.

[15] J. L. Bredin, E. D. Demaine, M. Hajiaghayi, and D. Rus, "Deploying sensor networks with guaranteed capacity and fault tolerance," in *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 2005, pp. 309–319.

[16] E. J. Duarte-melo and M. Liu, "Data-gathering wireless sensor networks: Organization and capacity," *Computer Networks*, vol. 43, pp. 519–537, 2003.

[17] L. Sankaranarayanan, G. Kramer, and N. Mandayam, "Hierarchical sensor networks: Capacity bounds and cooperative strategies using the multiple-access relay channel model," in *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, 2004, pp. 191–199.

[18] G. Barrenechea, B. Beferull-Lozano, and M. Vetterli, "Lattice sensor networks: capacity limits, optimal routing and robustness to failures," in *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*, 2004, pp. 186–195.

[19] A. Giridhar and P. Kumar, "Computing and communicating functions over sensor networks," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 4, pp. 755–764, 2005.

[20] A. Giridhar and P. R. Kumar, "Toward a theory of in-network computation in wireless sensor networks," *Communications Magazine, IEEE*, vol. 44, pp. 98–107, 2006.

[21] T. F. Abdelzaher, S. Prabh, and R. Kiran, "On real-time capacity limits of multihop wireless sensor networks," in *RTSS '04: Proceedings of the 25th IEEE International Real-Time Systems Symposium*, 2004, pp. 359–370.

[22] C. Comaniciu and H. Poor, "On the capacity of mobile ad hoc networks with delay constraints," *Wireless Communications, IEEE Transactions on*, vol. 5, no. 8, pp. 2061–2071, 2006.

[23] T. Abdelzaher, G. Thaker, and P. Lardieri, "A feasible region for meeting aperiodic end-to-end deadlines in resource pipelines," *Proceedings of 24th International Conference on Distributed Computing Systems*, pp. 436–445, 2004.

[24] H. Kang, H. Hong, S. Sung, and K. Kim, "Interference and sink capacity of wireless cdma sensor networks with layer architecture," in *ETRI Journal*, vol. 30, no. 1, 2008, pp. 13–20.

[25] M. Zukerman, *Introduction to Queueing Theory and Stochastic Teletraffic Models*, 2008. [Online]. Available: http://www.ee.unimelb.edu.au/staff/mzu/classnotes.pdf

[26] M. Harchol-Balter and D. Wolfe, "In network of queues, m/m/1 can outperform m/d/1," Tech. Rep., 1994.

[27] M. Ahmad and D. Turgut, "Congestion avoidance and fairness in wireless sensor networks," *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–6, 2008.

[28] J. Suhonen, M. Kohvakka, M. Kuorilehto, M. Hannikainen, and T. Hamalainen, "Cost-aware capacity optimization in dynamic multi-hop wsns," *Design, Automation & Test in Europe Conference & Exhibition*, pp. 1–6, April 2007.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6813
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6813