On Real-time Capacity Limits of Multihop Wireless Sensor Networks *

Tarek F. Abdelzaher, Shashi Prabh, Raghu Kiran
Department of Computer Science, University of Virginia, Charlottesville, VA 22904
zaher@cs.virginia.edu

Abstract

Multihop wireless sensor networks have recently emerged as an important embedded computing platform. This paper defines a quantitative notion of real-time capacity *of a wireless network. Real-time capacity describes how much real-time data the network can transfer by their deadlines. A capacity bound is derived that can be used as a sufficient schedulability condition for a class of fixed-priority packet scheduling algorithms. Using this bound, a designer can perform capacity planning prior to network deployment to ensure satisfaction of applications' real-time requirements.*

1 Introduction

This paper establishes fundamental capacity limits on *real-time* information transfer in multihop wireless networks. Real-time information transfer is one where there are deadlines on data communication. Only those bits that are transferred prior to their deadlines contribute towards useful information. Deadlines could arise for various reasons, for example, the necessity to react to external events in a timely manner, and the need to deliver dynamically changing data prior to the expiration of their respective validity intervals.

Recently, information-theoretic bounds have been derived for wireless networks that quantify the ability of the network to transfer bits across distance [11]. These bounds (often expressed in bit-meters) provide a fundamental understanding of network throughput as a function of network parameters such as bandwidth, total size and average density. While current bounds quantify throughput, they do not consider other key performance metrics; in particular network delay. For time-sensitive applications, it is useful to understand both delay and throughput limitations of the network. Observe that network delay and throughput are interrelated. Intuitively, the network should be able to transfer more bits by their deadlines if the deadlines are more relaxed. The results reported in this paper can be interpreted as understanding the feasible trade-off space between achievable throughput and delay.

Schedulability (i.e., the ability to meet deadlines) in distributed systems is, in general, an NP-hard problem. Hence, there is no closed-form formula to quantify the exact real-time capacity. To overcome this difficulty, in this work, we derive closed-form *sufficient* (rather than both necessary and sufficient) conditions on schedulability for a class of fixed-priority packet scheduling policies. Sufficient schedulability conditions have the merit of erring on the safe side. By definition, they guarantee that systems satisfying these conditions will meet their timing requirements. This property is convenient for capacity planning.

Observe that sufficient conditions for NP-hard schedulability problems generally exhibit a trade-off between simplicity and exact characterization. More complex expressions are needed to identify larger fractions of the schedulable space. Similarly to the Liu and Layland bound, being an early result, the expression derived in this paper is aimed at simplicity. We hope this simplicity provides a first step towards understanding the limitations on achievable delay and aggregate throughput in real-time multihop wireless networks.

The rest of this paper is organized as follows. Section 2 formulates the real-time capacity problem. Section 3 presents the main results of the paper. Section 4 verifies the results using simulation. Section 5 highlights related work. The paper concludes with Section 6.

2 Model and Problem Formulation

In this section, we describe the notion of real-time capacity in more detail, define the problem statement, and highlight the general approach taken to derive capacity bounds.

2.1 Real-time Capacity

In a multihop wireless network, it is natural to expect that more bits can be delivered by a larger deadline and that (exploiting spatial concurrency) more bits can be delivered in time if they traverse a shorter distance. Said differently, message schedulability is expected to decrease with an increase in transmitted bits, an increase in traversed distance, or a decrease in the relative deadline (the difference between bit arrival times and their due dates). It is therefore informa-

*The work reported in this paper was supported in part by the National Science Foundation under grants CCR-0093144, CCR-0208769, and CCR-0325197, and by MURI grant N00014-01-1-0576

tive to consider the bit-distance product of messages, normalized by their relative deadline. Intuitively, an increase in this normalized product decreases schedulability. This paper shows that, indeed, all messages are schedulable as long as the sum of their normalized bit-meter products remains below a certain bound. We call this bound the *real-time capacity* of the network, denoted C_{RT}.

To illustrate the notion of real-time capacity, let us use a numeric example. Consider two messages, A and B, traversing a wireless network. Message A is 1000 bits long and must travel a distance of 50 meters (i.e., consume a total of $50,000$ bit-meters) within 200 seconds. It is said to have a real-time capacity requirement of $50,000/200 = 250$ bit-meters/second. Message B must transfer 300 bits a distance of 700 meters within 100 seconds. Its capacity requirement is thus $300*700/100 = 2100$ bit-meters/second. Hence, the total real-time capacity needed is $250 + 2100 = 2350$ bit-meters/second. The messages are guaranteed to meet their deadlines as long as their combined requirements do not exceed the real-time capacity of the network (i.e., as long as $2350 < C_{RT}$).

It is often useful, in large multihop wireless networks, to define message *velocity* as the ratio of the end-to-end distance traversed (between source and destination) to the end-to-end deadline. Real-time capacity, defined above, can be equivalently interpreted as a constraint on feasible message velocities. All messages are schedulable if the sum of their velocities weighted by their respective sizes is less than the real-time capacity of the network. In the numeric example presented above, message A must traverse 50 meters within 200 seconds. Its velocity is thus $50/200 = 0.25$ meters/second. Multiplying by size, its weighted velocity is 250 bit-meters/second. Similarly, message B has a weighted velocity of $7 * 300 = 2100$ bit-meters/second. As before, adding up, the messages are schedulable if the sum of their weighted velocities is less than C_{RT}.

Real-time capacity, C_{RT}, of a wireless network depends on the order in which packets access the communication medium. This order is defined by the medium access control (MAC) protocol, and is called a *packet scheduling policy*. Many examples of prioritized MAC protocols are discussed in the related work section. In this paper, we concern ourselves with fixed-priority packet scheduling only since it is easier to implement on network nodes. While we do not discuss the feasibility of fixed priority scheduling, we restrict ourselves to a category of fixed-priority scheduling policies in which packet priority does not depend on absolute time and does not depend on distance metrics (such as the distance or the number of hops from source to destination). We call policies that satisfy the above conditions *independent* fixed-priority scheduling. The rational for this decision is two-fold. First, it is generally expensive to maintain clocks perfectly synchronized in a large network. Hence, priority schemes that require a notion of absolute time may sometimes be impractical. Second, nodes in a wireless network may be unaware of locations of other nodes. Hence, scheduling policies where priority assignment requires knowledge of accurate distance between two points might not be adequately supported.

Given the above constraints on priority assignment, we derive two important results. First, we prove that the best-case sufficient capacity bound for independent fixed-priority scheduling is $C_{RT} = \frac{n\alpha}{mN}W$, where α depends on the scheduling policy ($\alpha = 1$ for deadline monotonic scheduling), n is the total number of nodes in the network, N is the maximum communication path length, m is the number of nodes within a single hop neighborhood, and W is the transmission rate. The bound is derived for the capacity-maximizing case of a perfectly load-balanced network. Second, we derive an approximate bound for the common case of data monitoring networks in which a large number of distributed sensor measurements are collected by a much smaller number of sinks. The approximate bound in this case is $C_{RT} = \frac{\alpha K N}{1+0.5\,lnN}W$, where α, N, and W are as defined above and K is the number of sinks. In all cases, we first assume a perfect (zero overhead) MAC-layer protocol then quantify the implications of MAC-layer arbitration delays on network capacity (which affect the value of α). Finally, we discuss an effect similar to priority inversion (which is shown to cut capacity in half in the worst case) and quantify the capacity reduction due to load imbalance.

2.2 Problem Formulation and Approach

The derivation of the real-time capacity is made possible by our recent results in real-time scheduling that specify utilization bounds [5, 3] and feasible regions of multi-resource aperiodic task systems [4]. Feasible regions quantify the relation between load at various stages of a real-time system and the ability of the system to meet end-to-end deadlines.

Consider a sensor network with multiple data sources and data sinks. Packets traverse the network concurrently, each following a multihop path from some source to some destination. Each hop represents a packet transfer between two neighboring nodes on its path. A single-hop transfer occurs only if the receiver of this transfer is within the communication range of the sender. At this time, we do not make assumptions regarding channel symmetry or the shape of a node's communication range. We merely state that each node j can receive packets from a set of neighboring nodes we call $neighborhood(j)$.

Each packet T_i has an arrival time A_i defined as the time at which the sending application injects the packet into the outgoing communication queue of its source node. The packet must be delivered to its final destination no later than time $A_i + D_i$, where D_i is called the relative deadline of T_i. Different packets may generally have different

relative deadlines. We call packets that have arrived to the system but whose delivery deadlines have not expired *in-transit* packets. Each packet T_i has a transmission time C_i that is proportional to its length. This transmission time is incurred at each forwarding hop of its path.

Performing our analysis in terms of transmission times of packets (as opposed their sizes in bits) is an instance of separation of concerns, which allows us to focus on the real-time aspects. The mapping from bits to transmission time depends on physical and link-layer issues such as the channel bandwidth, the signal-to-noise ratio and the encoding technique used, which are concerns of information theory. We separate those concerns away by assuming a transmission speed, W, and deriving real-time capacity expressions in terms of that transmission speed. Note that, in practice, W may already be known and fixed for a particular network product, which makes our analysis very useful, as it can explicitly take this specification into account.

We define a per-node metric called *synthetic utilization* that captures the impact (on schedulability) of both the resource requirements and urgency associated with packets. We say that each packet contributes an amount C_i/D_i to the synthetic utilization of each hop along its path in the interval from its arrival time A_i to its absolute deadline $A_i + D_i$. More formally, at any time t, let $S(t)$ be the set of packets that are in-transit[1] in the entire sensor network. Hence, $S(t) = \{T_i | A_i \leq t < A_i + D_i\}$. We define $S_j(t) \in S(t)$ as the subset of $S(t)$ forwarded through node j. We define the synthetic utilization, $U_j(t)$, of node j as:

$$U_j(t) = \sum_{T_i \in S_j(t)} C_i/D_i \qquad (1)$$

which is the sum of the individual contributions to synthetic utilization (on this node) accrued over all in-transit packets passing through that node. Multiplying the packet transmission time, C_i, by the channel transmission speed, W, yields packet size. Hence, multiplying both sides of the above equation by W establishes the number of bits that can be transmitted by a node for each unit of time of the relative deadline. Summing up that quantity over the whole network is what defines the total real-time capacity requirements (in bit-hops per second) of all in-transit traffic. If we can compute an upper limit U_j on node synthetic utilization for which it is known that all deadlines are still met, then no deadline misses occur as long as capacity requirements are below $W \sum_j Uj$. In other words, the real-time capacity is given by:

$$C_{RT} = W \sum_j Uj \qquad (2)$$

[1]Remember that we consider a packet T_i to be *in transit* in the interval $[A_i, A_i + D_i)$.

Observe that capacity is first computed in bit-hops per second. To convert to bit-meters per second, it is enough to multiply the previous expression by the average distance per hop. The reader is also reminded that this paper derives sufficient but not necessary conditions only. It is possible for deadlines to remain satisfied when C_{RT} is exceeded.

3 Total Capacity

Consider a packet T_n traveling on an arbitrary path P through the wireless network. Without loss of generality, let us number the hops of that path $1, ..., N$ in the direction of the destination, such that node j is the destination of the j^{th} packet transfer. Figure 1 shows an example with $N = 4$.

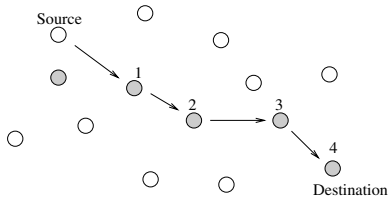

Figure 1. A path through the sensor network

To derive C_{RT}, we first find a *path-specific* condition on meeting end-to-end deadlines, which we call the *path feasible region*. The path feasible region is a function whose arguments are the synthetic utilization values of each hop along the path. It is guaranteed that the end-to-end deadlines of all packets transmitted along that path are met under an independent fixed-priority scheduling policy as long as this function does not exceed a pre-computed bound. The above condition is a generalization of utilization bounds for schedulability, such as [13, 12, 7]. It relates the synthetic utilizations of nodes along a path to the ability to meet end-to-end deadlines of aperiodic arrivals. This function is then used to infer the total capacity of the network.

Let the packet T_n on path P originate at its source node at time A_n and be delivered to the destination by $A_n + D_n$. The arrival time of the packet at hop j denotes the time it is fully inserted into the queue at that node. The departure time of T_n from hop j denotes the time the transmission of T_n is accomplished. The arrival time of the packet at hop $j + 1$ is equal to its departure of time from j plus the propagation delay. Let the time that packet T_n spends at hop j be denoted L_j, which is the interval between its arrival time and departure time at hop j. Thus, for the packet to be schedulable, it must be that $\sum_j L_j + p < D_n$, where p is the sum of propagation delays along the path. Since packet propagation occurs at the speed of light, it is much faster than transmission and queueing delays and is therefore neglected in the rest of the derivation.

In [4], we proved that the delay L_j of a task waiting for a resource accessed in fixed-priority order is related to

synthetic utilization as follows:

Theorem 1 (The Stage Delay Theorem) [4]: If task T spends time L_j at resource j, and u_j is a lower bound on the maximum synthetic utilization at that hop, then:

$$L_j = \frac{u_j(1 - u_j/2)}{1 - u_j} D_{max} \quad (3)$$

where D_{max} is the maximum end-to-end deadline of all tasks of higher priority than T.

The theorem was derived for abstract resources whose scheduler maintains a fixed-priority queue that determines the order of resource access. The resource is indivisible and is accessed by one task at a time in priority order. The theorem states that if the synthetic utilization of all tasks enqueued for resource j never exceeds u_j, then the delay of a task on that resource never exceeds L_j.

We now apply this theorem to wireless networks. In this context, task T is the act of sending one packet to the next hop, j, of its path. The resource under consideration is the channel bandwidth at the receiver of that transfer. It is either available (resource is idle) or occupied by other transmissions (resource is busy). The only transmissions that can contend on the channel are those originating in $neighborhood(j)$, defined as the set of nodes whose transmissions can be heard at node j. Note that due to the broadcast nature of the wireless channel, these transmissions will make the channel busy whether they are in fact destined to j or to some other node, as long as they originate in $neighborhood(j)$. Observe that j is a member of its own neighborhood, since its own transmissions contend on the same channel.

The objective of the medium access control protocol in a wireless network is to ensure that for each node j to which a packet is ready for transmission, only one packet is transmitted at a time from all nodes in $neighborhood(j)$. Moreover, packets are transmitted in priority order. In the following, we first consider the case of an ideal MAC layer, which implements medium arbitration with zero overhead. We then consider the effects of channel arbitration delay on network capacity. Observe that the set of all packets ready for transmission in $neighborhood(j)$ represents a virtual queue from which packets are dequeued in priority order to access receiver bandwidth. Hence, the stage delay theorem applies. Let us define the *neighborhood synthetic utilization* of node j, denoted H_j, as:

$$H_j = \sum_{i \in neighborhood(j)} U_i \quad (4)$$

For every hop j along the path P shown in Figure 1 (observe that j refers to the destination of the packet transfer at that hop), the stage delay theorem states that:

$$L_j = \frac{H_j(1 - H_j/2)}{1 - H_j} D_{max} \quad (5)$$

If the synthetic utilization in the neighborhood is always kept below H_j, the packet delay on hop j will never exceed L_j (assuming a zero-delay MAC layer). For packet T_n to be delivered to the destination by its end-to-end deadline, it must be that $\sum_j L_j < D_n$ (propagation delay is neglected). Substituting from Equation (5) for L_j in this summation, we get the equivalent condition:

$$\sum_{j=1}^{N} \frac{H_j(1 - H_j/2)}{1 - H_j} < \frac{D_n}{D_{max}} \quad (6)$$

Observe that D_n/D_{max} is the ratio of the deadline of packet D_n to that of a higher priority packet that delays its transmission. To obtain a conservative bound, this ratio must be minimized across all possible packet pairs. The minimum possible ratio depends on the scheduling policy. If deadline monotonic scheduling is used, by definition $D_n/D_{max} \geq 1$. In other words, for all packet pairs T_{hi} and T_{lo}, where T_{hi} has higher priority than T_{lo}, $\min_{T_{hi} \geq T_{lo}} D_{lo}/D_{hi} \geq 1$. (We assume equality in the worst case.) In general, for an arbitrary independent fixed-priority scheduling policy, we define $\alpha = \min_{T_{hi} \geq T_{lo}} D_{lo}/D_{hi}$ to be the minimum possible relative deadline ratio across all priority-sorted packet pairs. Intuitively, it represents the degree of urgency inversion (not to be confused with priority inversion). Urgency inversion occurs when a packet with a shorter relative deadline receives a lower priority by the scheduling policy than a packet with a larger relative deadline. For example, if priorities are assigned randomly, $\alpha = D_{least}/D_{most}$, where D_{least} and D_{most} are the minimum and maximum relative deadlines in the packet set respectively. The feasible region for such a scheduling policy is thus:

$$\sum_{j=1}^{N} \frac{H_j(1 - H_j/2)}{1 - H_j} < \alpha \quad (7)$$

In particular, for deadline monotonic scheduling, α is maximized:

$$\sum_{j=1}^{N} \frac{H_j(1 - H_j/2)}{1 - H_j} < 1 \quad (8)$$

Deadline monotonic scheduling is therefore optimal among independent fixed priority scheduling in the sense of maximizing the schedulability bound. This policy means that for each node j to which a packet awaits transmission, the MAC layer chooses for transmission the packet with the shortest relative deadline in $neighborhood(j)$. Later, we shall explore the case where the MAC layer is not ideal.

The main contribution of Equation (7) lies in relating end-to-end delay to a bound on the sum of throughput-like metrics (synthetic utilizations). These metrics can now

be related to real-time capacity. To relate H_j to the total real-time capacity C_{RT} of the network, let m be the average number of neighbors a node can send packets to. We call this parameter *node density*. Hence, on average, each node is a member of m sets $neighborhood(j)$ from which, $\sum_j H_j = m \sum_j U_j$ over all nodes in the network. Equation (2) can therefore be re-written as:

$$C_{RT} = \frac{W}{m} \sum_j Hj \tag{9}$$

where the summation is over all network nodes. This result will be used to compute real-time capacity. Two important cases are considered. First, we determine real-time capacity under the assumption of a perfectly load-balanced network (which happens to be the maximum capacity condition). Second, we determine real-time capacity for the case where all traffic congregates at a small number of sinks. This pattern is more common in sensor networks where all data is routed to a small set of observers.

3.1 The Maximum Bound

It is desired to maximize the total real-time capacity given by Equation (9). From the symmetry of this capacity expression with respect to H_j, as well as the symmetry of the schedulability condition given by Equation (7), the solution that maximizes capacity must be symmetric with respect to H_j. In other words, H_j must be equal for all j. Let this constant value of neighborhood synthetic utilization be H. This is called a *load-balanced* network. Let N be the length of longest communication path (in hops) that a node can be a part of. We call it the *communication diameter*. Intuitively, the communication diameter (not to be confused with the radio range) represents the degree of locality of communication. For example, if the communication pattern is such that nodes communicate with other nodes that are at most 5 hops away, then $N = 5$. It is desired to ensure that all packet deadlines are met on all paths up to length N. From Equation (7), the maximum neighborhood synthetic utilization H must therefore satisfy:

$$\frac{H(1 - H/2)}{1 - H} = \alpha/N \tag{10}$$

Solving for H and taking the lower value we get:

$$H = 1 + \frac{\alpha}{N} - \sqrt{1 + (\frac{\alpha}{N})^2} \tag{11}$$

Let the network contain n nodes. From Equation (9), the real-time capacity of the network is bounded by:

$$C_{RT} = \frac{n}{m}(1 + \frac{\alpha}{N} - \sqrt{1 + (\frac{\alpha}{N})^2})W \tag{12}$$

For a large network, the path length N is large. Hence, $(\frac{\alpha}{N})^2 << 1$. Consequently, the above equation can be simplified as follows:

$$C_{RT} = \frac{n\alpha}{mN} W \tag{13}$$

Observe that the more localized network communication is (i.e., the smaller N is), the larger the real-time capacity. The above capacity expression can be stated as the following theorem:

Theorem 2. The Maximum Capacity Theorem: In a large load-balanced connected multihop wireless network of n nodes, a radio transmission speed W, communication diameter N, node density m (nodes per communication radius), and a zero-delay medium access control implementing independent fixed-priority scheduling, a sufficient bound on real-time capacity is $\frac{n\alpha}{mN} W$, where α is the urgency inversion of the scheduling policy ($\alpha/N << 1$).

This theorem presents the first known bound that establishes real-time capacity limits as a function of network size, density, and radio transmission speed. It is the first step towards a comprehensive theory that addresses the relations between time, space, and information transfer capabilities of embedded wireless networks.

Observe that if the application ensures that the communication diameter, N, is bounded independent of network size, n (i.e., localized distributed algorithms are used), then:

$$C_{RT} = O(n)\alpha W/m \tag{14}$$

On the other hand, if paths are randomly chosen through the wireless network, the network diameter N is of the order of the square root of the area of the network, which in turn is of the order of the number of nodes. Consequently, $C_{RT} = O(n/\sqrt{n})\alpha W/m$, or:

$$C_{RT} = O(\sqrt{n})\alpha W/m \tag{15}$$

Comparing Equation (14) and Equation (15) emphasizes the importance of localized communication.

3.1.1 Frequency-Division Multiplexing

The capacity expression given above was derived based on the assumption that all nodes in a neighborhood transmit at the same frequency inside that neighborhood. It is interesting to compare that expression to the case where a dedicated channel is set up at the MAC layer for transmissions of each individual node. For example, frequency division multiplexing could be used, where each node in a neighborhood is assigned a unique frequency. Hence, transmissions from different nodes do not collide but are rather multiplexed in the frequency domain. The receiver employs a demultiplexor. Assuming a uniform node density, m, the

available spectrum must be divided into m unique channels. The transmission speed of an individual channel is thus $W_c = W/m$. Only one packet is transmitted from a node at a time.

Because channels are dedicated, packets at node j compete for transmission only with other packets on the same node (regardless of their destinations). Hence, the stage delay theorem states that packet delay at node j is given by:

$$L_j = \frac{U_j(1 - U_j/2)}{1 - U_j} D_{max} \tag{16}$$

The difference between Equation (16) above and Equation (5) introduced earlier lies in that the synthetic utilization used in the former refers only to that of node j, whereas in the latter it is that of the entire $neighborhood(j)$. This difference is due to the fact that in the latter case, packet transmission incurs contention from the entire neighborhood and not only from the transmitting node. Following the same steps as in the proof above, we get the path schedulability condition below (compare to Equation (7)):

$$\sum_{j=1}^{N} \frac{U_j(1 - U_j/2)}{1 - U_j} < \alpha \tag{17}$$

From Equation (2), the real-time capacity is given by $W_c \sum_j U_j$ over all nodes in the network (where W is now replaced by W_c). As before, from symmetry of the capacity expression and path schedulability expression with respect to U_j, the capacity is maximized when U_j is the same for all nodes. Let us denote it by U. This leads to:

$$U = 1 + \frac{\alpha}{N} - \sqrt{1 + (\frac{\alpha}{N})^2} \tag{18}$$

and since $(\alpha/N)^2 << 1$, we eventually get:

$$C_{RT} = \frac{n\alpha}{N} W_c = \frac{n\alpha}{mN} W \tag{19}$$

If $N = O(\sqrt{n})$, the above equation leads to:

$$C_{RT} = O(\sqrt{n})\, \alpha W/m \tag{20}$$

Comparing Equation (13) to Equation (19) and comparing Equation (15) to Equation (20), it can be seen that the maximum capacity expression is independent of how channel bandwidth is divided at the radio layer. Partitioning the bandwidth decreases both the transmission speed and contention by the order of the size of the neighborhood, leading to the same total capacity expression.

3.1.2 Time-Division Multiplexing

A disadvantage of frequency-division multiplexing is the increased cost of the radio hardware. The abstraction of dedicated channels can alternatively be implemented in software by multiplexing the channel in time. This can be achieved using clock-based or token-based MAC protocols. While it is not our intent to discuss the specifics of MAC-layer mechanisms that implement this abstraction, it is interesting to quantify their impact on real-time capacity. From a real-time perspective, one essential difference between multiplexing in time and true bandwidth partitioning, is an additional multiplexing delay term quantified in the seminal work on generalized processor sharing [17]. Intuitively, this term is due to the fact that exact fairness cannot be achieved at all times when packet transmissions are quantized and serialized. Virtual clock schemes have been proposed to guarantee bounded fairness, and hence bound the additional multiplexing delay. Let that delay be denoted d. Assuming channel arbitration schemes can be implemented in a distributed manner with zero overhead, the total delay of packet T_n on stage, j, is that predicted by the stage delay theorem plus d, or:

$$Delay = \frac{u_j(1 - u_j/2)}{1 - u_j} D_{max} + d \tag{21}$$

Summing up over all hops, we must ensure that the total delay is less than the relative deadline D_n. Hence, for a network of diameter N:

$$\sum_{j=1}^{N} (\frac{u_j(1 - u_j/2)}{1 - u_j} D_{max} + d) < D_n \tag{22}$$

Rearranging, we get:

$$\sum_{j=1}^{N} \frac{u_j(1 - u_j/2)}{1 - u_j} < \frac{D_n}{D_{max}} (1 - \frac{Nd}{D_n}) \tag{23}$$

Minimizing the right-hand side, we rewrite the above equation as:

$$\sum_{j=1}^{N} \frac{u_j(1 - u_j/2)}{1 - u_j} < \alpha' \tag{24}$$

where $\alpha' = \alpha(1 - Nd/D_{min})$, and D_{min} is the minimum end-to-end deadline. Comparing the above equation to Equation (7), it is easy to show that this equation leads to the same real-time capacity expressions, with the exception that α is replaced by $\alpha' < \alpha$. The difference between α and α' quantifies the capacity degradation we suffer because of delay introduced by having to wait for the node's turn to transmit in the presence of time-division multiplexing. For example, the real-time capacity of a load-balanced network becomes:

$$C_{RT} = \frac{n\alpha'}{mN} W \tag{25}$$

3.1.3 Realistic Medium Access Control

The above derivations assumed that nodes in the neighborhood of a receiver can immediately tell which outgoing packet has the highest priority among all those that can interfere with this receiver. The medium access control protocol then sends that packet first. In reality, some arbitration may be needed before the packet is transmitted. Several arbitration protocols have been proposed in previous sensor network and local area network literature that differ in the worst case amount of time it takes to determine who gets the medium.

Let B be the total additional delay experienced by a packet due to arbitration. The schedulability condition becomes:

$$\sum_j (\frac{U_j(1-U_j/2)}{1-U_j} D_{max} + B) < D_n \qquad (26)$$

Dividing by D_{max} and rearranging (similarly to the steps Section 3.1.2), we get:

$$\sum_{j=1}^{N} \frac{U_j(1-U_j/2)}{1-U_j} < \alpha(1-N\gamma) \qquad (27)$$

where $\gamma = B/D_{min}$ is the normalized blocking due to arbitration experienced by a packet at hop j. The above effect will propagate to capacity expressions, such that α is replaced by $\alpha' = \alpha(1-N\gamma)$. Observe, trivially, that unless arbitration delay is bounded, no deterministic guarantees are possible. However, choosing a value for B that is exceeded with a low probability, we get a capacity expression that upper-bounds the miss ratio by the same probability that B is exceeded. It is easy to prove that in a network where time-division multiplexing introduces per-hop delay d and MAC-layer arbitration introduces an additional delay B, the α in capacity expressions is replaced by $\alpha' = \alpha(1-N\gamma - Nd/D_{min})$

3.2 The Common Case Bound

The bounds described above were computed for a load-balanced network. While realistic load patterns in multihop wireless networks are generally difficult to characterize, a very common case that occurs in sensor networks is one where data is collected from all sensor nodes by a small number of sinks. These sinks (called *relays*) are usually more powerful data processing devices or transmitters that relay the data to a remote location. The routing protocol ensures that data from a given sensor node is sent to the nearest relay. Consequently, nodes closest to relays are the most congested. In the following, we derive an approximate expression for real-time capacity for the aforementioned common data communication pattern in sensor networks.

Let the number of relays in the network be K. Consider an arbitrary relay k, $1 \leq k \leq K$, and the set of sensors reporting to that relay. Observe that since traffic from all these sensors congregates at one sink, the total schedulable traffic generated by all sources is exactly the traffic that can be consumed by that sink. Moreover, at steady state, the sum of synthetic utilizations on all hops some fixed distance j from the sink is no larger than the total synthetic utilization at the sink. This is because the total flow of packets crossing a given perimeter cannot exceed what the destination sees, as shown in Figure 2. Assuming an average node density of m nodes per radio range R, the number of nodes j hops from the relay is approximated by the product of the ring area, $\pi(jR)^2 - \pi((j-1)R)^2$, and the number of nodes per unit area, m/R^2, which yields $(2j-1)m$ nodes. Hence, the average per-node synthetic utilization decreases linearly with distance from the destination as it gets divided among $(2j-1)m$ nodes. The same applies to the neighborhood synthetic utilization. Assuming the neighborhood synthetic utilization at the destination is H, and renumbering the hops in ascending order from destination to sources, $H_j = H/(2j-1)m$. From Equation (7), the path-specific schedulability condition is:

$$\sum_{j=1}^{N_k} \frac{\frac{H}{(2j-1)m}(1-\frac{H}{2(2j-1)m})}{1-\frac{H}{(2j-1)m}} < \alpha \qquad (28)$$

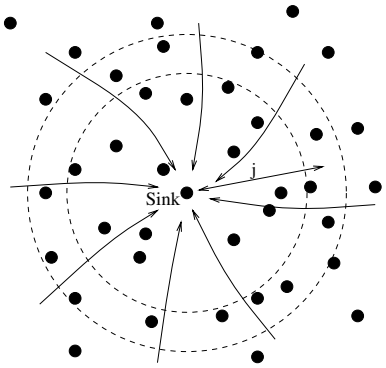

Figure 2. The common sink case

The above equation can be solved for H as a function of the number of hops N_k on the longest path to relay k. The equation can be rewritten as:

$$\frac{H}{2m}\sum_{j=1}^{N_k}\frac{1}{2j-1-H/m} + \frac{H}{2m}\sum_{j=1}^{N_k}\frac{1}{2j-1} < \alpha \qquad (29)$$

Since, $H < 1$ (which can be derived from Equation (7) given that $\alpha \leq 1$), for large j, $\frac{1}{2j-1-H/m}$ is approximately equal to $\frac{1}{2j-1}$. Using results of the harmonic series summation and some algebraic manipulation, we can show that

the sum $\sum_{j=1}^{N_k} \frac{1}{2j-1}$ is well approximated by $1 + 0.5ln\ N_k$. Thus, Equation (29) leads to:

$$H = \frac{\alpha m}{1 + 0.5\ lnN_k} \tag{30}$$

As observed above, note that the aggregate synthetic utilization over all nodes distance j from the destination is upper bounded by that at the destination (or else some packets will be dropped). Let us cut the area reporting to relay k into N_k concentric zones, each encompassing all nodes that are the same hop distance from the relay. As argued above, the sum of neighborhood synthetic utilizations within each of the N_k zones is upper bounded by H. From Equation (9), the total real-time capacity of all nodes reporting to the relay becomes $\frac{W}{m}(N_k H)$. Substituting for H from Equation (30) and multiplying by the number of relays K, we get:

$$C_{RT} = \frac{\alpha K N_k}{1 + 0.5\ lnN_k} W \tag{31}$$

This approximate expression is very useful for capacity planning as will be demonstrated by example below. Evaluation shows that the approximation is very accurate in that (i) no deadlines misses are observed in our experiments when the approximate bound is met, and (ii) misses occur very shortly after the bound is exceeded. This is true even in very large networks (in excess of 1000 nodes). The parameter α in the capacity expression can be modified as previously described in Section 3.1.3 to account for MAC layer delays.

3.3 Example: Sizing the Network

The main advantage of the capacity expressions derived in the previous subsections lie in the ability of an application developer to choose network and application parameters that result in schedulability guarantees. We illustrate this claim by an example. Consider a network of 1000 nodes and 8 relays to be deployed such that the path length from any node to the nearest relay is 7 hops on average but no more than 10 hops. Let the transmission speed be $50KBps$. Furthermore, let each node generate 24 Bytes of sensor data (including headers) periodically at period T. Data must reach a relay within 1.5 seconds. FIFO scheduling is used (observe that $\alpha = 1$ because all deadlines are the same). It is desired to find the minimum T that does not violate schedulability.

The capacity bound derived in this paper can be used to solve this problem. Remember (from Section 2) that real-time capacity can be interpreted as a bound on the weighted sum of message velocities, where velocity is weighted by the message size. In a schedulable system, there can be at most $\lceil 1.5/T \rceil$ messages in transit from the same source, accounting for a total of $24\lceil 1.5/T \rceil$ bytes. The capacity requirements of each source i are thus $24\lceil 1.5/T \rceil v_i$, where v_i is its message velocity given by the ratio of the hop distance N_i (between the source and the sink) to the end-to-end deadline. The total requirements of all 1000 sources are thus $1000 * 24\lceil 1.5/T \rceil * 7/1.5 = 112,000\lceil 1.5/T \rceil$ byte-hops/second.

The available real-time capacity of the network is computed from Equation (31) to be $1 * 8 * 10 * 50,000/(1 + 0.5\ ln10) = 1859.3$ Kilobyte-hops/second. For all traffic to be schedulable, we thus have $112,000\lceil 1.5/T \rceil \leq 1859.3K$, or $\lceil 1.5/T \rceil \leq 16.6$. For a minimum T, and since the ceiling function has integer values, we have $\lceil 1.5/T \rceil = 16$. Hence, $T = 93.75ms$. Observe that if the flow deadline is changed, the schedulable sampling period may change too. For example, if the deadline is reduced to $150ms$, following the same steps, it can be seen that the minimum period becomes $150ms$.

Observe that the throughput limits of the system can be inferred by setting the traffic deadline $D \rightarrow \infty$. In this case, the total capacity requirements of flows are given by $1000 * 24\lceil D/T \rceil * 7/D$, where $\lceil D/T \rceil \rightarrow D/T$ as $D \rightarrow \infty$. Hence, traffic capacity requirements become $168000/T$, independent of the deadline. (More generally, the weighted velocity of a periodic flow becomes independent of its deadline.) Comparing the aggregate capacity requirements to the available real-time capacity, we get $168000/T \leq 1859.3K$, from which the minimum period is $T = 90ms$. In other words, it is (conservatively) estimated that smaller periods may create unbounded delays that cannot satisfy any finite deadline. Hence, while our bound is derived primarily for the benefit of real-time applications, it can also be used to reason about network throughput limits in bandwidth-constrained systems of deterministic non-real-time periodic flows.

In general, the bound can be used prior to deployment to determine network and workload parameters for which all deadlines are met. For example, the capacity expression could be used to find the number of relays required given a particular sampling period, the maximum distance between relays for deadlines to be met, the guaranteed end-to-end data delivery delay for particular traffic and network parameters, the required radio radius that keeps capacity above traffic requirements (observe as the radio radius determines the number of hops, which affects both the traffic requirements and the capacity bound), or simply as a feasibility check on a particular workload and network configuration to determine if it meets timing specifications. Hence, the capacity expression is a very versatile tool for real-time network sizing.

3.4 Pseudo Priority Inversion

The discussion presented so far assumes that packet transmission in the neighborhood of a receiver contends only with other packets transmitted in the same neighborhood.

In the following, we show that this assumption is not always satisfied. A packet due for transmission might be blocked by packets *outside* the neighborhood of its receiver. This blocking imposes additional delays, hence reducing real-time capacity. In this section, the resulting reduction in capacity is quantified.

To illustrate this point, let us consider the situation in the wireless network depicted in Figure 3. In this figure, sender S_1 has a packet to send to receiver R_1. Senders S_2 and S_3 have a packet each for receiver R_2. In the neighborhood of R_2, sender S_2 has the highest priority and should send its packet first. However, in the neighborhood of R_1, sender S_1 has a higher priority. Consequently, sender S_1 transmits first, thereby blocking S_2. Since S_2 is blocked, the MAC layer in the neighborhood of R_2 lets S_3 send its packet. A condition similar to priority inversion occurs in the neighborhood of R_2 since S_3 transmits before S_2. Unlike true priority inversion, this condition is not brought about by blocking on a lower priority task. In this case, S_2 is blocked because of a higher priority sender, S_1, that can transmit concurrently with S_3 (a situation that has no equivalent in single processor scheduling where a task that preempts S_2 should also preempt S_3). We call this condition *pseudo priority inversion*.

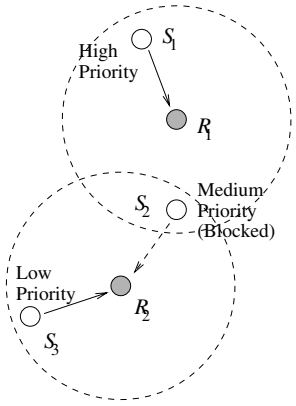

Figure 3. Priority inversion in multihop wireless networks

It can be shown that pseudo priority inversion can cascade. For example, in Figure 3, S_1 might be blocked by a higher priority sender S_0 in which case S_2 can send and S_3 must wait. This chain can be arbitrarily long. In general, whether or not S_3 will transmit before S_2 may depend on transmissions that are arbitrarily far away.

To quantify the effect of pseudo priority inversion on capacity, we include the effect of transmissions outside the current neighborhood when we use the stage delay theorem. We observe that the transmission of a packet to node j competes not only with packets transmitted in $neighborhood(j)$ (as it would in an ideal world) but also with packets transmitted outside $neighborhood(j)$ to nodes in $neighborhood(j)$. We now quantify the synthetic utilization of the above two components. First, the synthetic utilization due to packets transmitted in $neighborhood(j)$ is by definition the neighborhood synthetic utilization, H_j. Second, since nodes mostly forward received packets, the total traffic received by nodes in $neighborhood(j)$ is generally equal to traffic transmitted by those nodes. Hence the synthetic utilization of received traffic in $neighborhood(j)$ is equal to the neighborhood synthetic utilization H_j. Consequently, the synthetic utilization due to the fraction of that traffic originating outside $neighborhood(j)$ is less than or equal to H_j. Adding the two components together, the total synthetic utilization of traffic that contends with a transmission to node j is no less than H_j and no greater than $2H_j$. Hence, by the stage delay theorem, to account for pseudo priority inversion, we replace H_j by βH_j in the derivations, where $1 \leq \beta \leq 2$. Making that substitution, we eventually get a modified capacity expression, $C_{RT}(actual) = C_{RT}/\beta$. In other words, the capacity is at most reduced in half. Hence, for a load balanced network, a conservative bound on capacity in the presence of pseudo priority inversion is:

$$C_{RT}(actual) = \frac{n\alpha'}{2mN}W \tag{32}$$

Similarly, for the common case of traffic collected by a small number of sinks, K, with a maximum source-to-sink hop count N_k, Equation (31) for real-time capacity becomes:

$$C_{RT}(actual) = \frac{\alpha' K N_k}{2 + lnN_k}W \tag{33}$$

where $\alpha' \leq 1$, as defined in Section 3.1.3.

3.5 Cost of Load Imbalance

Finally, it is interesting to compare the bound derived for a load-balanced network, given by Equation (32), to that derived for the data collection scenario, given by Equation (33). For brevity, let us denote them by C^{lb}_{RT}, and C^{dc}_{RT} respectively. The difference between the two is the cost of load imbalance. Specifically, it is interesting to compare these capacities for the same communication diameter (i.e., for $N = N_k$). In other words, in both cases, we assume the communication pattern is such that the maximum hop distance to a destination is N. It is expected under these conditions that Equation (32) should produce the higher bound since it is derived for the optimal (balanced) load distribution. Dividing Equation (32) by Equation (33), with $N = N_k$, we get:

$$\frac{C^{lb}_{RT}}{C^{dc}_{RT}} = \frac{n(2 + lnN)}{2KmN^2} \tag{34}$$

Note that, in the above expression, n/K is numerically equal to the number of nodes reporting to a single relay, which is given by the number of nodes within a radius of N hops around the relay. Assuming uniform density, this number is proportional to N^2 (the area). We also know that when $N = 1$, the number of nodes within a single hop radius is m. Hence, the number of nodes within N hops from the relay is mN^2, from which $n/K = mN^2$. Substituting in Equation (34), it is simplified to yield:

$$\frac{C_{RT}^{lb}}{C_{RT}^{dc}} = \frac{2 + lnN}{2} \tag{35}$$

Two points are interesting to observe about the above expression. First, as expected, $C_{RT}^{lb} \geq C_{RT}^{dc}$. The difference grows when N grows, because increasing the hop-count from which a sink is collecting data only increases the load imbalance compared to a load-balanced case of the same communication diameter. Second, when $N = 1$, the two bounds are identical. This is because at this point, in the data collection scenario, all senders communicate directly with a sink, originating $1/m$ of the neighborhood traffic. Hence, the load is balanced. Equation (35 quantifies the capacity reduction due to load imbalance.

4 Evaluation

We implemented a simulator to study the capacity of wireless sensor networks. The simulator constructs a network of sensor nodes of a user-specified size in a perturbed grid structure. The radio layer is implemented as a simplified disk model of a specified radius (range). The sinks are distributed uniformly across the network. We generated traffic at each non-sink node such that each packet was assigned a deadline at random from a preselected set. All packets were sent to their nearest sink. Packet contention was resolved in priority order. Only those nodes were allowed to transmit who were not within the radio range of another node that was already scheduled to receive a transmission. Ties between simultaneously arriving same priority packets were broken at random. We implemented a *shortest path* routing scheme in which the neighboring node nearest to the sink was chosen as the next hop. If this node was blocked due to another transmission, the packet was not scheduled until that transmission was over. The MAC layer implements deadline monotonic scheduling for medium arbitration.

All packets were checked for deadline misses. If there was a miss, the actual capacity consumption of all in-transit traffic was computed by multiplying each in-transit packet by the traversed hop count and normalizing by the end-to-end deadline. Each run was repeated 50 times with different randomized workloads. The minimum capacity consumption at which a deadline miss occurred was recorded.

Figure 4 and Figure 5 show the effect of increasing the radio radius, shown on the top horizontal axis, on real-time capacity in a network of 800 nodes and 1600 nodes respectively. Observe that increasing the radio radius also increases the neighborhood size (i.e., the number of nodes within the radio range), shown on the bottom horizontal axis. The number of sinks was kept at 12. The lower curve in both figures is the analytic capacity bound computed from Equation (33). This equation accounts for priority inversion. Parameters α' and W are set to 1. The top curve shows the minimum consumed capacity at which deadline misses were observed in simulations. Note the very close match between simulation and analytic prediction even at very large network sizes. As expected, capacity decreases with increasing radio radius because fewer concurrent transmissions become possible.

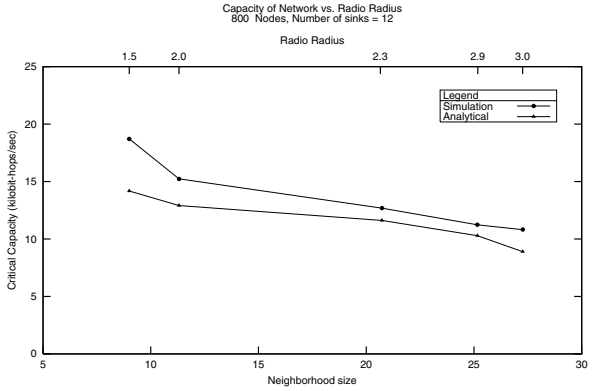

Figure 4. Effect of radio radius, 800 nodes

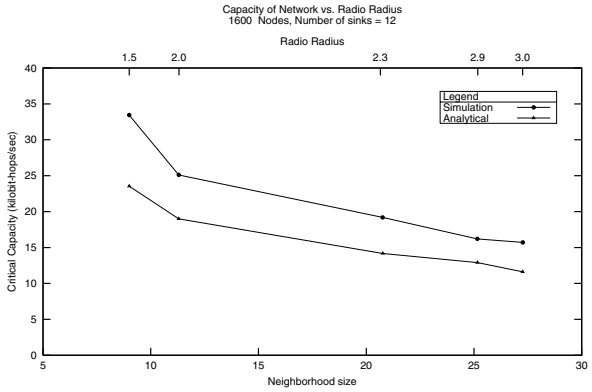

Figure 5. Effect of radio radius, 1600 nodes

Figure 6 and Figure 7 repeat the experiments for networks of 800 and 1600 nodes respectively, this time varying the number of sinks. The radio range is kept constant at a neighborhood size of 12 nodes. As before, a very close match is observed between simulation and analysis. Capacity grows with the number of sinks because data collection bottlenecks are alleviated.

Finally, Figure 8 shows the sharp increase in the miss ratio in a network of 800 nodes that occurs when capacity is exceeded. In this curve, the network workload is increased past the capacity bound. The miss ratio is then

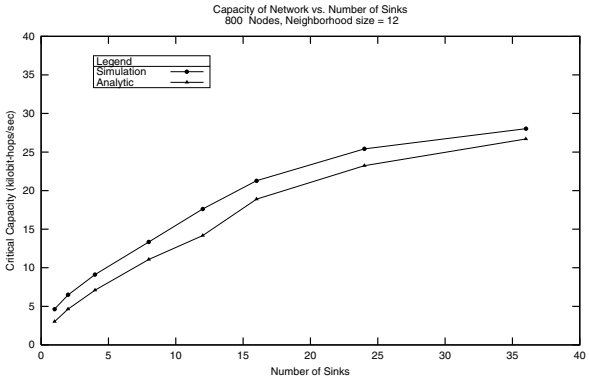

Figure 6. Effect of the number of sinks, 800 nodes

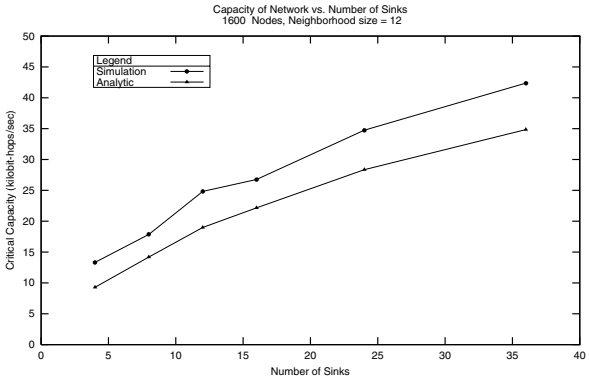

Figure 7. Effect of the number of sinks, 1600 nodes

plotted against the capacity requirements of the workload shown on the horizontal axis. Each point in the figure corresponds to a single experiment. Two sets of data points are shown for two different radio ranges that correspond to neighborhoods of 12 nodes and 24 nodes respectively. From Figure 4, we can see that the capacity bounds for these two cases are around 13 and 12 respectively. The miss ratio becomes non-zero shortly after these bounds are exceeded and increases sharply soon thereafter.

5 Related Work

The work described in this paper leverages previous results in aperiodic schedulability bounds. The first synthetic utilization bound for fixed priority scheduling of aperidic tasks was derived by the authors in [5]. This result was later extended it to multiprocessor scheduling [2], tasks with resource requirements [3], and real-time data pipelines [4]. This paper is the first extension of these results to the realm of sensor networks.

While several other utilization bounds were reported in previous literature, such as [13, 12, 18, 9, 21, 7, 16, 15, 14], they were confined to variations of the periodic task model

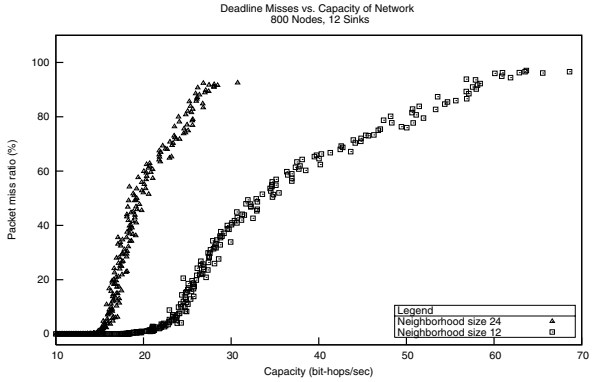

Figure 8. Miss ratio

and hence are inapplicable in our case. The priority inversion problem due to blocking was addressed in [19], proposing modifications to schedulability tests using the priority ceiling protocols. It would be interesting to derive MAC layer protocols that minimize the impact of psuedo priority inversion on real-time capacity.

The results derived in this paper assume a priority-aware MAC protocol. There have been several MAC protocols which provide differentiated services in wireless networks. In [8], an implicit prioritized MAC protocol for wireless sensor networks has been presented with seven frequencies for transmission to avoid channel interference. In [22], a black-burst scheme is proposed that provides real-time access to CSMA wireless networks. This scheme is used in [20] to provide differentiated services at the MAC layer. In [24], a MAC protocol for supporting deterministic QoS in wireless local area networks is presented. In [23], narrow band busy tone (BT) signals are used to do priority scheduling at the MAC layer. A protocol which uses different values of contention window for different classes is presented in [6]. Similar contention-window-based schemes are presented in [1]. A dynamic time-division duplexed scheme is presented in [10]. While most prioritization schemes rain probabilistic, some degree of service differentiation is generally possible.

In our future work, we shall study the effects of MAC layers such as the above on capacity bounds for sensor networks. Observe that this is analogous to studying the effects of scheduling policies on task schedulability conditions. The result should be a body of knowledge for reasoning about real-time constraints in mission-critical wireless network applications.

6 Conclusions

This paper presented the first expressions for real-time capacity of a sensor network. We derive a sufficient condition for schedulability under fixed-priority scheduling which allows capacity planning to be employed prior to deployment such that real-time requirements are met at run-time. The

bound is derived for load balanced networks, as well as networks where all traffic congregates at a number of sinks. The effects of various MAC-layer multiplexing schemes such as time-division multiplexing and frequency-division multiplexing are discussed. A problem similar to priority inversion is presented and its effect on capacity is approximately quantified. The capacity expressions are evaluated in simulation. It is shown that deadlines are never missed when the network capacity bound is not exceeded. When the traffic requirements exceed the capacity bound by some margin, deadline misses were observed. This simulation validates the results and shows that capacity planning can be performed safely using the derived bounds. We hope this paper will serve as an initial step towards developing a more complete body of literature on schedulability in ad hoc wireless environments. Extensions may include investigating variable density networks, realistic MAC-layers, and effects of energy constraints to name a few.

References

[1] I. Aad and C. Castelluccia. Differentiation mechanisms for ieee 802.11. In *INFOCOM '01, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, pages 209–218, Anchorage, Alaska USA, April 2001. IEEE.

[2] T. Abdelzaher, B. Andersson, J. Jonsson, V. Sharma, and M. Nguyen. The aperiodic multiprocessor utilization bound for liquid tasks. In *Real-time and Embedded Technology and Applications Symposium*, San Jose, California, September 2002.

[3] T. Abdelzaher and V. Sharma. A synthetic utilization bound for aperiodic tasks with resource requirements. In *15th Euromicro Conference on Real-Time Systems*, Porto, Portugal, July 2003.

[4] T. Abdelzaher, G. Thaker, and P. Lardieri. A feasible region for meeting aperiodic end-to-end deadlines in resource pipelines. In *IEEE International Conference on Distributed Computing System*, Tokyo, Japan, March 2004.

[5] T. F. Abdelzaher and C. Lu. Schedulability analysis and utilization bounds for highly scalable real-time services. In *IEEE Real-Time Technology and Applications Symposium*, TaiPei, Taiwan, June 2001.

[6] M. Barry, A. T. Campbell, and A. Veres. Distributed control algorithms for service differentiation in wireless packet networks. In *INFOCOM '01, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, pages 582–590, Anchorage, Alaska USA, April 2001. IEEE.

[7] E. Bini, G. Buttazzo, and G. Buttazzo. A hyperbolic bound for the rate monotonic algorithm. In *13th Euromicro Conference on Real-Time Systems*, Delft, Netherlands, June 2001.

[8] M. Caccamo, L. Y. Zhang, L. Sha, and G. Buttazo. An implicit prioritized access protocol for wireless sensor networks. In *Proceedings of the 23rd IEEE International Real-Time Systems Symposium*, pages 39–48, Austin, TX, December 2002. IEEE.

[9] X. Chen and P. Mohapatra. Lifetime behavior and its impact on web caching. In *IEEE Workshop on Internet Applications*, 1999.

[10] S. Choi and K. G. Shin. A cellular wireless local area network with qos guarantees for heterogeneous traffic. In *INFOCOM 1997, Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings. IEEE*, pages 1030–1037, Kobe, Japan, April 1997. IEEE.

[11] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2), March 2000.

[12] T. W. Kuo and A. K. Mok. Load adjustment in adaptive real-time systems. In *IEEE Real-Time Systems Symposium*, December 1991.

[13] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *J. of ACM*, 20(1):46–61, 1973.

[14] J. M. Lopez, J. L. Diaz, and D. F. Garcia. Minimum and maximum utilization bounds for multiprocessor rate monotonic scheduling. In *13th Euromicro Conference on Real-Time Systems*, Delft, Netherlands, June 2001.

[15] J. M. Lopez, M. Garcia, J. L. Diaz, and D. F. Garcia. Worst-case utilization bound for edf scheduling on real-time m ultiprocessor systems. In *12th Euromicro Conference on Real-Time Systems*, pages 25–33, Stockholm, Sweden, June 2000.

[16] D.-I. Oh and T. P. B. TP. Utilization bounds for n-processor rate monotone scheduling wi th static processor assignment. *Real-Time Systems*, 15(2), 1998.

[17] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Transactions on Networking*, 1(3), June 1993.

[18] D.-W. Park, S. Natarajan, and A. Kanevsky. Fixed-priority scheduling of real-time systems using utilization bounds. *Journal of Systems and Software*, 33(1):57–63, April 1996.

[19] L. Sha, R. Rajkumar, and J. P. Lehoczky. Priority inheritance protocols: An approach to real-time synchronization. *IEEE Transactions on Computers*, Septemeber 1990.

[20] J. P. Sheu, C. H. Liu, S. L. Wu, and Y. C. Tseng. A priority mac protocol to support real-time traffic in ad hoc networks. *ACM Wireless Networks*, 10:61–69, 2004.

[21] W. K. Shih, J. Liu, and C. L. Liu. Modified rate-monotonic algorithm for scheduling periodic jobs with deferred deadlines. *IEEE Transactions on Software Engineering*, 19(12):1171–1179, December 1993.

[22] J. L. Sobrinho and A. S. Krishnakumar. Quality-of-service in ad hoc carrier sense multiple access wireless networks. *IEEE Journal on Selected Areas in Communications*, 17:1353–1368, 1999.

[23] X. Yang and N. H. Vaidya. Priority scheduling in wireless ad hoc networks. In *Proceedings of 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing, MOBIHOC*, pages 71–79, Lausanne, Switzerland, June 2002. ACM.

[24] Y. Ye, C. J. Hou, and C. C. Han. Qgma: A new mac protocol for supporting qos in wireless local area networks. In *Proceedings of the sixth International Conference on Network Protocols*, pages 339–348, Austin, TX, October 1998. IEEE.