SenProbe: Path Capacity Estimation in Wireless Sensor Networks[+]

Tony Sun, Ling-Jyh Chen, Guang Yang, M. Y. Sanadidi, Mario Gerla
Computer Science Department
University of California, Los Angeles
{tonysun, cclljj, yangg, medy, gerla}@cs.ucla.edu

Abstract

The infusion of mobile computing elements (e.g. data mules, mobile actuators/sinks and etc.) into Wireless Sensor Networks undoubtedly requires the knowledge of various wireless network properties to fasten their interactivity. Moreover, since the rate of a wireless link can vary dynamically and rapidly due to changes in interference, distance, or energy optimization policy, timely knowledge of the end-to-end path capacity in WSN can be of great assistance to more efficient routing and traffic management. Particularly, information on end-to-end path capacity can be used to assess the applicability of a given WSN deployment, and determine whether the deployment meets the requirements of its intended applications. In this paper, we propose SenProbe, an end-to-end path capacity estimation tool inspired by CapProbe concepts, yet specifically designed to work in the popular CSMA-CA based WSNs. With simulations and analysis, we evaluated the behavior and effectiveness of SenProbe, and showed that it is a simple and less intrusive technique that yields accurate results.

1. Introduction

The infusion of mobile computing platforms (e.g. data mules, mobile actuators/sinks and etc.) into wireless sensor network (WSN) will undoubtedly require knowledge of various wireless network properties to fasten their interactivity. This motivates and necessitates the development of a new end-to-end tool that can monitor and measure the properties of Wireless Sensor Networks (WSNs) well. To provide the optimal interaction between mobile computing elements and its ambient sensor fabrics, effective evaluation and measurement of effective channel capacities along wireless sensor paths are of realistic interest, especially for activities such as capacity planning, protocol design, performance analysis and system deployment.

Moreover, since the rate of a wireless link can vary dynamically and rapidly due to changes in interference, distance or energy optimization policy, timely knowledge of the end-to-end path capacity in WSN can be of great assistance to more efficient routing and traffic management. Additionally, information on end-to-end path capacity can be used to assess the applicability of a given sensor net deployment, and determine whether the deployment meets the requirements of its intended applications.

Although the path capacity estimation problem has been extensively studied in the literature, most tools work in the scenarios of wired and/or last-hop wireless networks and measure the bottleneck link capacity, e.g. [1][5][8][10]. The complexity and convergence time required for these schemes are not well suited for multi-hop adhoc WSN. Moreover, the assumption of bidirectional setup of some of the above techniques has proved to yield detrimental results for adhoc wireless networks.

In fact, end-to-end path capacity estimation in WSN is very challenging. Wireless capacity estimation depends not only on the rate of the "narrow" link along the path (as in a wired network), but also on the topology, path layout, interference between nodes along the path and several other environmental parameters. The end-to-end path capacity estimation tool must be simple and accurate. The estimation process must be fast so that it can reflect the path capacity in a timely manner, even when the actual capacity is varying. The estimation must be independent of cross traffic and non-intrusive so that it does not disturb the ongoing applications. A successful capacity estimation mechanism must understand and satisfy these factors respectively.

Previous technique proposed by Li et al in [12]

[+] This material is based upon work supported by the National Science Foundation under Grant No. ANI-0335302 and CNS-0435515.

addressed the adhoc end-to-end path capacity by sending a brute force UDP packet stream to measure the maximum achievable throughput. This scheme produces realistic results, but is not very practical since it heavily impacts existing traffic, and its result is affected by current on-going traffic conditions as well. Other work, such as a testbed measurement study of wireless available bandwidth in [11], which focused on the capacity in last hop wireless scenarios, addresses a different problem from the study of capacity in multi-hop WSN.

In this paper, we propose SenProbe, an end-to-end path capacity estimation tool inspired by CapProbe concepts, yet specifically designed to work for the popular CSMA enabled WSNs. We evaluated the behavior and effectiveness of SenProbe with simulations, and showed that SenProbe is a simple and less intrusive technique that yields accurate path capacity estimates for wireless sensor network based on CSMA.

The rest of the paper is organized as follows. In section 2, we discuss the design considerations for CSMA environment and what SenProbe intended to measure. An in-depth description of SenProbe is presented in section 3; follow by various Qualnet validation of SenProbe in section 4. In section 5, we summarize related work and recap the Internet CapProbe technique on which SenProbe is based upon. Section 6 concludes the paper.

2. Design Considerations

The Media Access Control (MAC) plays an important role in the performance of managing data impulses in a shared wireless medium. There is a growing effort in designing suitable MAC schemes to allow good performance while achieving energy conservation (i.e. S-MAC [19], TMAC [17], and etc). Yet for many solutions based on TDMA, many practical problems, including synchronization and scheduling overhead, need to be resolved before it is widely used.

A growing number of sensor networks use the basic CSMA-CA or variant for medium access. For example, the widely used Berkeley motes [3] use a simple CSMA-CA MAC as part of the TinyOS [16] platform. The emerging IEEE 802.15.4 [4] standard, which is popularly adapted by the sensor networks communities, also uses CSMA-CA when it is set in the beaconless mode.

In this paper, we focus on the basic CSMA-CA enabled sensor network. We now briefly discuss the design considerations for CSMA-CA based sensor networks, and the path capacity metric that our proposed scheme, namely SenProbe, is intended to measure.

2.1 Carrier Sense Multiple Access-Collision Avoidance (CSMA-CA) Overview

Carrier Sense Multiple Access-Collision Avoidance (CSMA-CA) is a protocol for carrier transmission for wireless networks. In the basic (CSMA-CA) scheme, each time a device wishes to transmit a packet, it checks to be sure the channel is clear (no other node is transmitting at the time). If the channel is clear, then the packet is sent. If the channel is not clear, the node waits for a randomly chosen period of time, and then checks again to see if the channel is clear.

The theoretical maximum capacity for CSMA-CA depends on the time needed for carrier sensing, signal propagation time, as well as the number of hops along the wireless path that the transmission need to traverse. If nodes can detect idle periods on the channel quickly, CSMA-CA offers very good channel utilization regardless of the offered load.

However, CSMA-CA suffers from the well-known hidden terminal problem in multi-hop environments [16]. IEEE 802.11 utilizes the virtual carrier sense (VC), namely RTS/CTS exchanges, to eliminate hidden terminals. In sensor networks, since packets are usually small and sparse in nature [2], the signaling cost of RTS/CTS and even ACK exchanges would be high in terms of both time and energy costs.

Therefore, in the context of sensor networks, the VC scheme is costly and mostly unnecessary during normal operations. As a result, CSMA-CA remains a popular MAC scheme of choice for sensor networks. This requires a technique that can work satisfactorily, with or without the VC scheme, to properly estimate the end-to-end path capacity in a responsive way, yet incurring low cost.

2.2 What Does SenProbe Actually Measure?

In a wired network, the capacity along an end-to-end path, or called the path rate, is equal to the bottleneck capacity on the path. However, the path rate in a multi-hop wireless network is related, but not necessarily equal, to the minimum of the link capacities on the path. We now review the factors that determine path capacities in a wireless network, and verify that SenProbe measurements can indeed reflect these factors.

We recall that the effective end-to-end path rate is defined as the maximum achievable data rate in the absence of any cross traffic. In the simplest CSMA-CA scheme, this is smaller than the raw data rate at the physical layer. The difference is due to channel access coordination to handle multiple, pipelined packets on the path, which incorporates the works of carrier sensing as well as random back-off mechanisms. If the optional ACK is deployed, the effective end-to-end rate will also depend on the overhead from transmitting the ACK packet, the upper-bound capacity can be calculated as

$$C = \frac{S}{S + ACK} \times C_P \qquad (1)$$

for a single-hop connection, where S is size of the network layer packet (including IP header), ACK is the size of the ACK packet, and C_P is the link capacity at physical layer.

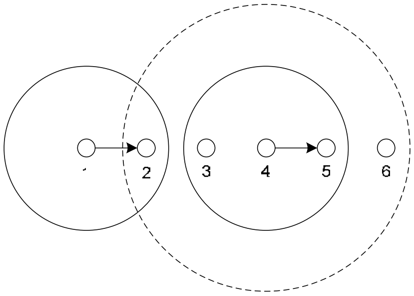

Figure 1: The chain topology, where the solid-line circle denotes a node's effective transmission range and the dotted-line circle denotes a node's interference range.

Moreover, due to the collision avoidance mechanism, the effective capacity of a wireless link decreases when there is more than one node within its collision domain. For example, when *N* active nodes, belonging to the same path, are within each other's transmission range, the maximum effective rate on that path is *C/N* since only one of the *N* nodes can transmit at any time. Naturally, it is unusual to have a sensor network path that hops several times within the same collision domain. However, this would clearly cause a reduction in the effective data rate. Such rate reduction must be also be captured by SenProbe.

Much more common is the reduction in capacity occurred when the path spans multiple hops. We consider a simple forwarding chain topology as shown in Figure 1. For simplicity, we assume the nodes are placed on a line with 300 meters between each pair of adjacent neighbors; the effective transmission range of each node is 350 meters. When the radio interfering range is the same as the transmission range, previous study by Li et al [12] has shown that the effective capacity of a forwarding chain topology becomes just 1/3 of the effective capacity of a single-hop connection.

In fact, as identified in [19], the radio interference range is usually much larger than the transmission range. Therefore, the effective end-to-end capacity of a chain configuration will further decrease. For instance, in Figure 1, if the interference range (marked by a dotted-line circle) is 750 meters, transmission from node 4 will interfere with transmission from node 1 to 2. In other words, simultaneous data transmission is not possible among nodes 1, 2, 3, and 4. It turns out that the theoretical limit of an ideal adhoc multi-hop forwarding chain can achieve 1/4 of the throughput that a single-hop transmission can achieve. The theoretical limit as well as the maximum throughput that is measure by a rapid CBR traffic (used in [12]) is illustrated in Figure 2.

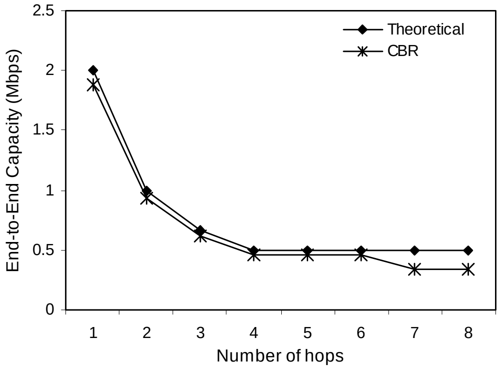

Figure 2: Theoretical capacity in an ideal adhoc multi-hop forwarding chain vs. simulated throughput as measured by rapid CBR traffic

3. SenProbe

SenProbe is inspired by the work of CapProbe, a well-proved bottleneck link capacity estimation tool for wired and last-hop wireless networks [8]. However, SenProbe differs from CapProbe in several significant ways. First of all, SenProbe, while utilizing the packet pair dispersion concept, is a packet train technique that is designed to overcome the hidden terminal effects present in a CSMA-CA environment. Secondly, SenProbe is a one-way (instead of round-trip) estimation scheme, which better reflects the true capacity of the wireless channel (by preventing round-trip packets from contending for the wireless channel). Thirdly, SenProbe is designed to work under conditions not present in a typical Internet path, such as when the WSN is mobile and interfered. Lastly, SenProbe measures the maximum achievable rate on an "unloaded" path in the shared wireless medium, the capacity estimation results actually incorporates the influence of channel access coordination in wireless channel as well as random back-off experienced by the packets.

3.1 **SenProbe Algorithm**

SenProbe intends to estimate the *end-to-end path rate* based on one-way measurements. The end-to-end path rate is the *maximum achievable rate* over the wireless path in the absence of any competing traffic. The maximum *achievable rate* is typically lower than the nominal channel transmission rate due to characteristics of wireless networks, e.g. multi-hopping, carrier sensing, random backoff mechanisms. SenProbe is able to accurately measure such achievable rate

Similar to CapProbe, SenProbe relies on the time dispersion between estimation packets to provide capacity estimation for WSN. However, instead of using back-to-back packet pairs, SenProbe relies on back-to-back packet train to overcome the effect of hidden terminal in CSMA-CA. The length of this back-to-back packet train depends on the interference range and the transmission range of the specific radio technology under question, and can be determined from the equation

$$N_{TRAIN} = \left\lfloor \frac{InterferenceRange}{TranmissionRange} \right\rfloor + 2 \tag{2}$$

Often the interference range in a wireless radio network is approximately twice the transmission range. In this case, only four probing packets of the fixed size will be sent back-to-back from the sender to the receiver. The sending time is stamped on every packet by the sender; the One Way Delay (OWD) of every packet is then calculated at the receiver. Finally, the path capacity (i.e. rate) estimation is calculated at the receiver and reported back to the sender.

The receiver measures the OWD of every packet in *kth* packet train received as the difference between time received (clocked at the receiver) and time sent (stamped in the packet header) as

$$OWD\,[k,i] = T_{rcv}[k,i] - T_{send}[k,i], \quad 1 \le i \le N_{TRAIN} \tag{3}$$

$$OWDSUM\,[k] = \min_{1 \le i < j \le N_{TRAIN}} OWD\,[k,i] + OWD[k,j] \tag{4}$$

Various OWD sums are then computed for each packet train, separately summing the OWD of leading packets and their trailing packets, and the minimum OWDSUM is kept for the *kth* packet train. The "good" dispersion sample *r* (i.e. the estimation packet samples encountering no cross traffic) is the sample with the minimum OWD sum, which satisfies the following conditions:

$$r = \arg\min_{k} OWDSUM\,[k] \tag{5}$$

$$(u,v) = \arg_{i,j} \min_{1 \le i < j \le N_{TRAIN}} OWD\,[k,i] + OWD[k,j], i < j \tag{6}$$

The corresponding capacity is given by $C=P/T$, where *P* is the packet size and *T* is the dispersion of the packet pair calculated by

$$T = OWD[r,v] - OWD[r,u] \tag{7}$$

The N_{TRAIN} back-to-back probing packets are designed to overcome the hidden terminal effect of the CSMA environment. As illustrate by the time sequence snapshot in Figure 3, when N_{TRAIN} =4 (interference range doubles transmission range), interference from packet 1 eventually collides with both packets 2 and 3 at node 2 due to the hidden terminal effect. Packet 1 always survives the ordeal, but packets 2 and 3 are usually dropped. However, in this scenario, since packet 4 is distant enough to avoid packet collision with packet 1, it becomes feasible to use the dispersion between packet 1 and 4 to estimate the correct *effective end-to-end path capacity*. It turns out that this scheme only depends on the minimum OWD sum to yield the correct capacity estimation, therefore, if packets 2 and 3 are not dropped and yield the minimum OWD, SenProbe will still produces the correct result (which can be the case when the network path is less than 4 hops).

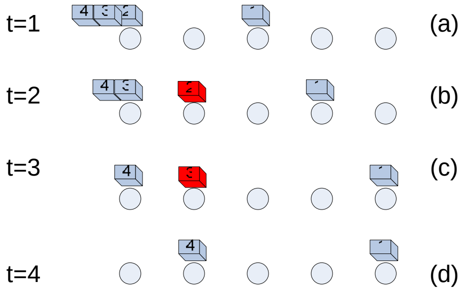

Figure 3: For N_{TRAIN} =4, snapshot of SenProbe packets progressing in an ideal adhoc forwarding chain of 5 nodes (CSMA environment)

SenProbe does not implement the "convergence test" (as CapProbe does), in order to make the algorithm simple, fast, and timely to the highly varying characteristics of wireless networks. Instead, SenProbe simply reports the capacity estimation after receiving a certain number of samples defined by *S*. Similar to CapProbe, the accuracy of capacity estimation increases as *N* grows. However, a large *S* value is not suitable for mobile wireless networks as it will lead to high latency in estimation and may not allow us to capture the dynamic properties of the wireless network.

Apart from the number of samples, *S*, the latency of the estimate also depends on the sending rate of probing packets. For simplicity, SenProbe sends probing packets of size *P* bytes at a constant rate of *R* trains/second, or equivalently N_{TRAIN} $*P*R$ bytes/second. The expected duration of a single estimation is then approximately *S/R* seconds. Clearly, the larger *R* is, the less time a capacity estimation process takes. However, *R* should be upper-bounded since a large *R* may disturb the ongoing foreground traffic in the network or even congest the network. As a result, the capacity estimate may become inaccurate (hard to get one good sample) or extremely slow (packets are lost due to congestion).

The probing parameters *S* and *R* need to be carefully tuned in accordance with the underlying network properties and by trading off precision for speed. This tradeoff clearly depends on the application. In this paper, we simply set S = 200, P= 1500, and R = 4 sample trains/second for all simulations.

3.2 SenProbe Analysis

The transmission delay for each estimation packet can be calculated as

$$T_{tx} = \frac{P}{C} + T_{backoff} + T_{overhead} \tag{8}$$

P is the packet size, *C* is the capacity of the link, $T_{backoff}$ is the minimum amount of time radio has to backoff (or wait) before sending another packet, $T_{overhad}$ is the amount of time radio has to spend in sending excess overhead (including possibly acknowledgement packets) .If *L* represent the end-to-end hop count between the sender and the receiver. For N_{TRAIN} =4, a correct estimation can be obtained as long as there exist minimum idle time of

$$T_{idle}(s) = \begin{cases} L*2*T_{tx} \ , \ L \le 4 \\ (L+4)T_{tx} \ , L > 4 \end{cases} \tag{9}$$

in the wireless path, as illustrated in Figure 4.

For example, in the popular IEEE 802.15.4 beaconless mode, T_{tx} translates to 6.368 ms. If the interference range is twice of the effective transmission range, a successful capacity estimate between two nodes 5 hops away would only require less than 58ms. For WSNs that don't experience heavy traffic. SenProbe can usually produce a good estimate given the network idle time period requirement.

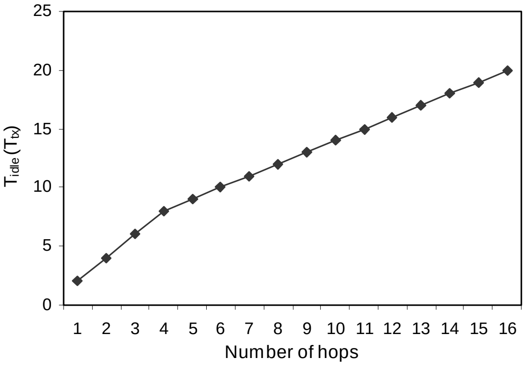

Figure 4: T_{idle} requirement in T_{tx} unit time as a function of wireless hop length between the sender and the receiver

4. Simulation Results

This section presents simulation results of SenProbe in estimating the end-to-end path capacities in a number of CSMA-CA wireless network configurations. The *QualNet* [15] network simulator is used along with 802.11 radio configurations, but with RTS/CTS/ACK messages disabled for all the simulations to mimic CSMA-CA environment. The wireless channel rate is fixed at 2Mbps. The effective transmission range is 350 meters while the interference range is 750 meters. Nodes remain stationary during all of the simulations.

4.1 Path capacity of adhoc multi-hop forwarding chain in CSMA-CA wireless environment

This subsection studies the capacity on a single adhoc multi-hop forwarding chain, where packets originate from the first node and are forwarded to the last node on the chain, such as when a sink node request information from sensors multiple hops away. Forwarding nodes are expected to contend and interfere with their neighbors, meaning that the effective path capacity will be adversely affected.

Here, we use the same topological scenario as shown in Figure 1. The transmission range (marked by a solid-line circle) of the CSMA-CA node is 350 meters, the interference range (marked by a dotted-line circle) is 750 meters, and the nodes are placed on a straight line with 300 meters in between. We run SenProbe on the multi-hop forwarding chain topology using a packet size of 1500 bytes, and the results are depicted in Figure 5.

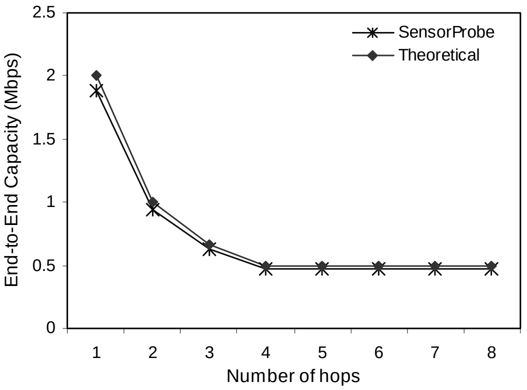

Figure 5: Capacity estimates along a multi-hop forwarding chain in CSMA-CA enabled wireless network

Based on the discussion in section 2, the effective end-to-end capacity decrease as the length of the chain grows longer, demonstrating an inverse relationship between the two variables. When the chain length exceeds four, at the packet size of 1500 bytes, the estimated end-to-end capacity converges to ~500 kbps, which is approximately ¼ of the single hop transmission, extremely close to the analytical results discussed earlier.

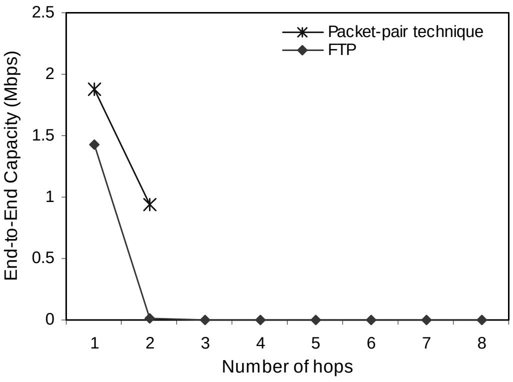

Figure 6: End-to-End Path Capacity measured by a FTP connection, and by a pure packet-pair technique (one way CapProbe)

In Figure 6, we illustrate the inadequacy of capacity estimation result that would be obtained using a normal FTP connection or a pure packet pair technique (e.g. one-way CapProbe).

4.2 Path capacity within the same interference region

Next, we evaluate the capacity of a highly interfered wireless path. More precisely, we wish to validate the *C/N* relationship mentioned in section 2. To this end, we have designed a simulation experiment where the hops of the multi-hop path are all in the same collision domain. The topology and configurations used here are the same as in section 4.1, except that the distance between a node and its next-hop neighbor is reduced to only 5 meters here. SenProbe is run using the packet size of 1500 bytes with various numbers of hops. Figure 7 shows the path capacity estimation at each number of hops. As predicted by the model, the end-to-end capacity estimate decreases as the inverse of the number of interfering nodes (or equivalently, the number of hops in the same collision domain).

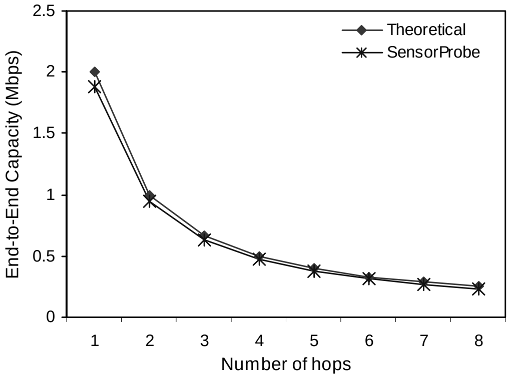

Figure 7: End-to-end capacity estimation of wireless multi-hop connections within the same collision domain.

4.3 Path capacity of adhoc multi-hop forwarding chain in CSMA-CA wireless network w/ ACK enabled

If Acknowledgement (ACK) packets are used in conjunction with the original CSMA-CA scheme, it is imperative to make sure that 1) functionality of SenProbe remains undisturbed when acknowledgement packets are enabled, and 2) SenProbe measurements remain accurate and correctly reflect the effective capacity of the wireless channel with the added overhead of ACK packets.

From section 2, we know that the effective end-to-end capacity decreases when ACK packets are enabled in CSMA-CA networks. Therefore, we do expect a slight drop in the effective capacity of the wireless channel. If the probing packets are 1500 bytes in size and the ACKs are 40 bytes each, the maximum effective capacity at hop length greater than four is calculated to be just around ~487kbps according to what we know from section 2, which is close to the capacity estimates achieved by SenProbe as illustrated in Figure 8.

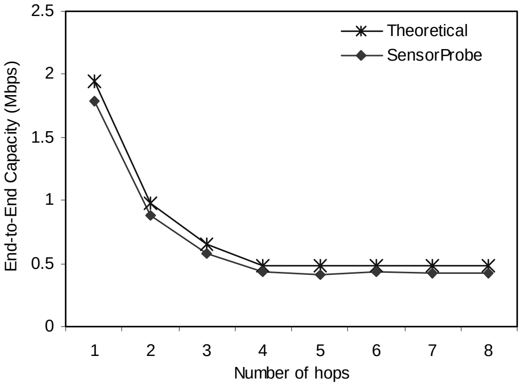

Figure 8: Capacity estimates along a multi-hop forwarding chain for CSMA-CA with ACK enabled wireless sensor network

4.4 Path capacity with interfering traffic in CSMA-CA environment

Since grid topologies are more representative of WSN configurations than chain topologies, we now consider the *n* x *n* regular grid shown in Figure 9. Nodes are placed 300 meters away from both their horizontal and vertical neighbors. Radio transmission range is set to 350 meters, and the radio interfering range is set to 750 meters.

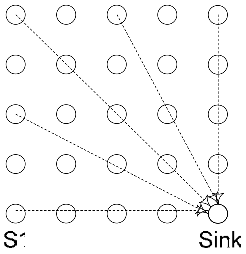

Figure 9: A regular 5x5 Grid topology, with five cross traffic originating from the edges of the network.

We consider traffic patterns scenario where 5 different source nodes are delivering different data to the subscribing sink, data aggregation is not assumed as the data types are vastly different and cannot be piggybacked to the source as packet sizes are big. SenProbe is run from node S1 to the sink node, as all the dash paths carry traffic flows with Poisson distribution at an average rate that varies up to 300Kbps.

As shown in Figure 10, SenProbe is able to give the correct capacity estimate despite the cross traffic. For example, in a *4x4* regular grid topology, where the path length is 3, and SenProbe reached an capacity estimate of ~630Kbps, which is pretty consistent with our results from the multi-hop forwarding chain of length 3. For a *5x5* regular grid, the path length is 4; SenProbe was able to measure ~470kbps, which is also consistent with our findings earlier. Although many estimation packet trains do not make through to the destination (due to hidden terminal effects), every once in a while an estimation packet train is able to sneak through the cross traffic and provide an estimate. However, the packet trains tend to be separated by an extra amount of time due to the intervening traffic (i.e. delay caused by random backoffs), which leads to the slight underestimation of the capacity.

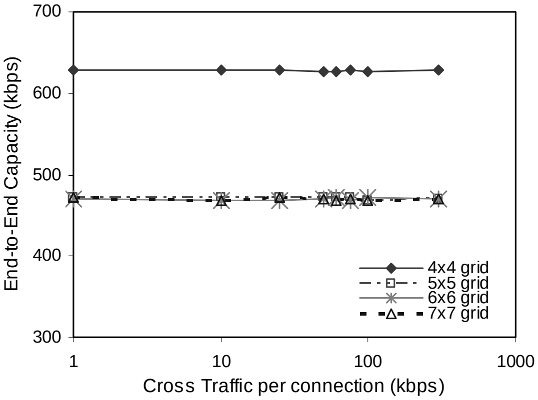

Figure 10: Capacity estimation in a grid wireless sensor network with 5 separate flows of cross traffic in CSMA-CA environment

We conduct the same experiment for CSMA-CA environment that utilizes ACK packets, and illustrate the results in Figure 11. For *5x5* regular grid, SenProbe is able to measure ~420kbps at light traffic. But for heavier cross traffic (i.e., more than 100kbps) and bigger grid sizes (*7x7 grid),* the estimates become incorrect since the grid becomes totally saturated with cross traffic. Figure 11 shows that below the saturation point, the capacity estimate is accurate, but around the saturation point (~70kbps) the estimate is not correct anymore. It becomes more difficult for SenProbe to find an “idle window” to sneak through and provide an estimate, since the usage of acknowledge packets increases the amount of traffic that is transmitted through the air.

This experiment essentially reconfirms properties that we already discovered in wired networks. Namely, CapProbe can estimate the capacity correctly up to the point where the path becomes saturated! This is not very intuitive in multi-hop adhoc networks where the estimation packets are separated by 4 hops. Any cross traffic transmitted by 2-hop neighbors during this 4 hop window will interfere with the probing packets and invalidate the minimum sum requirement. Thus, for the same network loading, the risk of interference with the probing packet appears to be much higher in adhoc networks than in wired ones. However, since traffic is sparse in most WSNs, SenProbe can usually estimate the correct capacity.

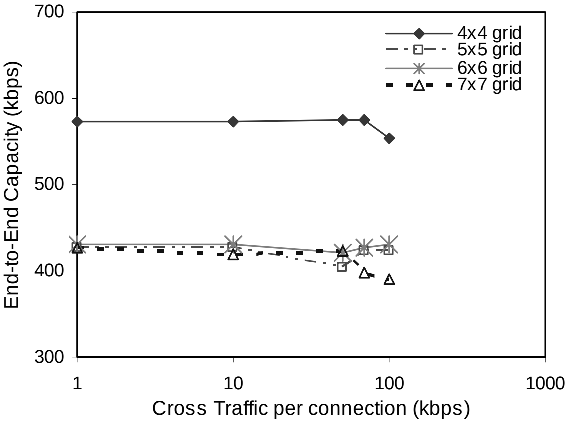

Figure 11: Capacity estimation in a grid wireless sensor network with 5 separate flows of cross traffic in CSMA-CA /ACK

5. Related Work and Background

5.1 **Related Work**

Link capacity estimation has been extensively studied in wired networks, e.g. [1] [5] [8][10]. Early research on capacity estimation relied on either delay variations among probe packets [5] or dispersion among probe packets [1][10]. Dovrolis revealed in [1] that the dispersion distributions can indeed be multi-modal without multi-channels, and that the strongest mode in the multimodal distribution of the dispersion may correspond to either (1) the capacity of the path, or (2) a “compressed” dispersion, resulting in capacity over-estimation, or (3) the Average Dispersion Rate (ADR), which is lower than the capacity. Kapoor et al proposed a packet-pair based approach called CapProbe in [8]. CapProbe combines delay and dispersion measurements to estimate the bottleneck link capacity fast and accurately.

However, all of the above schemes have been and evaluated in wired and last-hop wireless scenarios. They have never been tested in ad hoc wireless networks. Capacity estimation in ad hoc wireless networks remains challenging due to the rapidly varying channel conditions, presence of node mobility, and multiple hops of wireless links.

In [12], Li et al examined the interaction of 802.11 MAC and ad hoc forwarding and the ability to infer path capacity for several simple configurations and traffic patterns. A brutal force approach (i.e. using UDP packets to probe the maximum throughputs of the network) was used in simulation and experiments to validate the hypothetical limits of utilization in a chain network (1/4 of effective capacity also illustrated in Figure 2). However, this approach was only able to obtain a lower utilization.

In addition, as we mentioned earlier, Li’s approach is intrusive and the result approximates path capacity only if the network is idle and static.

5.2 **CapProbe**

CapProbe [8] is a recently proposed bottleneck capacity and path rate estimation technique shown to be both fast and accurate over a large range of scenarios. When two back-to-back packets are launched into a network, they are always separated at the bottleneck link by an interval related to bottleneck capacity. If such interval is preserved until destination unperturbed, it will allow us to compute the bottleneck link capacity (as shown in Figure 12). The bottleneck capacity is equal to the UDP stream rate that can be sustained by the path – hence the name of “path rate” estimator often given to these tools. The interval between “interfered” packet-pair samples might be either expanded or compressed, where “expansion” leads to under-estimation and “compression” leads to over-estimation of the capacity.

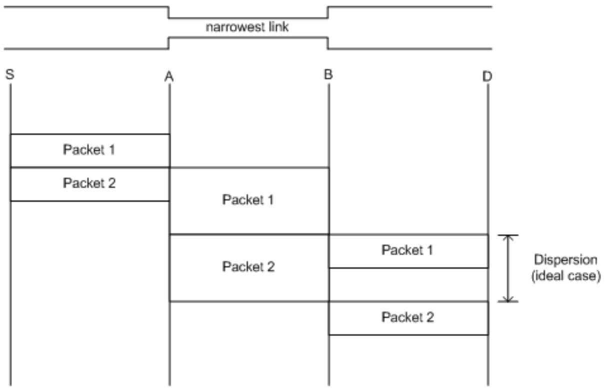

Figure 12: Packet-pair technique: the ideal case.

CapProbe combines the use of time interval measurements and end-to-end delay measurements to filter out packet pair samples interfered by cross traffic. By construction, an incorrect value of capacity estimate can occur only if cross traffic has interfered with the packet pair. In this case, queueing ensues and the delay of one or both packets will be larger than the minimum observed in absence of cross traffic. The sum of the delays of the packets in the packet pair is defined as *delay sum*. A delay sum, which does not include any cross-traffic induced queuing delay, is referred to as the *minimum delay sum*. Any sample with delay sum larger than the observed minimum is thus discarded as it must have been interfered with. The capacity is easily derived from the equation:

$$C = P / T \tag{10}$$

where P is the sampling packet size, and T is the interval between packets with minimum delay sum.

CapProbe has been validated in wired and last-hop wireless networks with a variety of configurations. However, it has not been tested in ad hoc wireless networks yet. In wired networks, CapProbe is generally used in the round-trip mode, to evaluate the minimum capacity over the two directions. In ad hoc wireless networks, the round-trip mode is inadequate, since the first packet once relayed by the receiver may collide with the incoming second packet. Based on the CapProbe concept, we thus design a one-way technique, called SenProbe, to estimate the unidirectional path capacity in WSNs.

6. Conclusion

In this paper we have proposed an end-to-end path capacity estimation tool, SenProbe, specially designed for multi-hop CSMA-Ca based WSNs. To solve the problems of hidden terminals and interference between neighboring nodes, commonly seen in CSMA-based multi-hop wireless networks, SenProbe uses back-to-back packet trains, and use packet dispersion between the packet trains to measure the path capacities in a one-way fashion. We have shown via analysis and simulations that SenProbe is fast and accurate in capturing the dynamic path capacities in CSMA-CA based wireless environments. Results have confirmed that SenProbe is a simple and effective tool to measure the end-to-end path capacity in CSMA-CA based networks such as WSNs.

References

[1] Dovrolis, C., Ramanathan, P., and Moore, D., "*What do packet dispersion techniques measure?"* in Proceedings of IEEE Infocom'01, 2001.

[2] G. Pttie and W. Kaiser., "Wireless Integrated Network Sensors", Communications of the ACM, 43(5): 51-58, 2000.

[3] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler,D., and Pister, K., "System Architecture directions for network sensors," in proceedings of the 9th International Conference on Arch. Supoort for Programming Languages, and Operating Systems, pages 93-104, Novenber 2000.

[4] IEEE 802.15.4 Standard

[5] Jacobson, V., *"Pathchar: A tool to infer characteristics of Internet paths"*, ftp://ftp.ee.lbl.gov/pathchar

[6] Ji, Z., Yang, Y., Zhou, J., Takai, M., and Bagrodia, R., "Exploiting Medium Access Diversity in Rate Adaptive Wireless LANs," in proceedings of ACM MobiCom 2004.

[7] Kamerman, A. and Monteban, L., "WaveLAN II: A high-performance wireless LAN for the unlicensed band," Bell Labs Technical Journal, pp. 118-133, Summer 1997.

[8] Kapoor, R., Chen, L.-J., Lao, L., Gerla, M., Sanadidi, M. Y., "CapProbe: A Simple and Accurate Capacity Estimation Technique," in proceedings of ACM SIGCOMM 2004.

[9] Lacage, M., Hossein, M., and Turletti, T. IEEE 802.11 Rate Adaptation: A Practical Approach. In Proceedings of ACM MSWiM 2004.

[10] Lai, K., Baker, M., "*Measuring Bandwidth*," In Proceedings of IEEE INFOCOM '99, p. 235-245.

[11] Lakshminarayanan, K., Padmanabhan, V., Padhye. J., "Bandwidth Estimation in Broadband Access Networks," in proceeding of ACM IMC 2004.

[12] Li, J., Blake, C., Couto, D., Lee, H. I., and Morris, R., "Capacity of Ad Hoc Wireless Networks," in procedings of ACM MobiCom 2001.

[13] Paxson, V., "On Calibrating Measurements of Packet Transit Times," in proceeding of ACM SIGMETRICS 1998.

[14] Qiao, D., Choi, S. Jain, A., and S, K. G., "MiSer: An Optimal Low-Energy Transmission Strategy for IEEE 802.11a/h", in proceedings of MobiCom 2003.

[15] Qualnet Simulator, available from http://www.scalable-networks.com/

[16] TinyOS Homepage, available from http://www.tinyos.net

[17] T. van Dam, K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," ACM Sensys 2003

[18] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," ACM SIGCOMM, 1994

[19] Ye, W., Heidemann, J., Estrin, D., "An energy efficient MAC protocol for wireless sensor networks," In Proceedings Infocom 2002, New York, June 2002.

[20] Xu, K., Gerla, M., and Bae, S., "How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks?" in proceedings of Globecom 2002.