End-to-End Channel Capacity Measurement for Congestion Control in Sensor Networks*

Jaewon Kang
DATAMAN Lab
Rutgers University
jwkang@cs.rutgers.edu

Yanyong Zhang
WINLAB
Rutgers University
yyzhang@winlab.rutgers.edu

Badri Nath
DATAMAN Lab
Rutgers University
badri@cs.rutgers.edu

ABSTRACT

Congestion control in sensor networks is important not only to improve the overall throughput but also to enlengthen the network lifetime by saving the scarce energy wasted during congestion. While throttling the incoming traffic during congestion can effectively alleviate congestion, it also lowers the throughput called *accuracy level* observed by the application deployed in the sensor network.

In this paper, we investigate how congestion can be alleviated by increasing the available resource amount rather than suppressing the incoming traffic. To optimally increase the available channel capacity during congestion, the end-to-end channel capacity of a flow with multiple paths and their various configurations is analyzed, which is then verified by simulations. Based on the analysis and simulations, we suggest several guidelines on how to adjust the end-to-end channel capacity under the various congestion scenarios.

1. INTRODUCTION

Congestion at a node happens when the incoming traffic volume exceeds the resource amount available to the node. To alleviate congestion, either the incoming traffic into the node should be throttled (referred to as *traffic control*) or the available resource amount should be increased (referred to as *resource control*). While the traffic control has been extensively studied especially in wired networks, the resource control has received little attention. The major reasons are: (1) additional resources may not be available during congestion; (2) even though the congested node has extra resources, sometimes it is difficult to increase the available resources in quick response to congestion. For example, the time to increase outgoing link capacity by establishing an additional routing path towards the ultimate destination may be an order of magnitude greater than the temporal granularity of a transient congestion; and (3) if the increased resources incurs too much overhead such as energy consumption, a *persistent* congestion may not be handled solely by the resource control.

However, the traffic control during congestion is not suitable for some congestion scenarios in sensor networks. Sensor networks are increasingly deployed to monitor the disastrous events such as fires in a forest, earthquake, cracking of a building, etc. These networks operate with a low reporting rate at the sensors most of time. However, once an abnormal phenomenon is detected, the sensors start to report the event to the sinks with a high reporting rate, so that the large volume of delivered data can make the monitoring application at the sinks accurately grasp the ongoing event. These high reporting rates from the sensors frequently incur congestion in the network. Since the data reported during the crisis state are of great importance, throttling the incoming traffic during this period greatly hurts the accuracy level seen by the application at the sinks. Therefore, the resource control can be a better option not only for alleviating congestion but also for enhancing the accuracy level if extra resources are quickly available during congestion.

In sensor networks, the availability of resources are elastic and redundant, thereby readily available, to achieve a reasonable network lifetime. For this reason, the resource control can be easily implemented in sensor networks. Adjusting the radio transmission power in conjunction with congestion level can be one choice.

In this paper, we address the issues involved in congestion control in sensor networks. In Section 2, we classify different types of congestion found in sensor networks and discuss the congestion detection which differs from the conventional congestion detection schemes. We also discuss two typical congestion control schemes, i.e. traffic control and resource control schemes and under what condition each scheme should be performed.

In Section 3, we investigate how resource, especially end-to-end channel capacity, can be increased rather than suppressing the incoming traffic during congestion. To optimally increase the demanded channel capacity, the end-to-end channel capacity a flow with multiple paths towards a sink and their various topology is first quantified by the technique called *time frame assignment*. This technique is later verified by simulations. Based on these analysis and simulations, we suggest several guidelines to effectively increase the end-to-end channel capacity during various congestion scenarios in sensor networks such as merging traffic or crossing traffic case.

These guidelines on the end-to-end channel adjustment

*This research work was supported in part by DARPA under contract number N-666001-00-1-8953, NSF grant ANI-0240383 and DIMACS.

.

can serve as a basis not only for congestion control but also for the sensor network design such as calibrating the radio transmission power, deciding the number of deployed sensors, controlling the duty cycles of nodes. The accurately quantified end-to-end channel capacity can also be used by routing protocols to find a routing path that has a higher channel capacity.

Related work is introduced in Section 4 and Section 5 concludes this paper.

2. CONGESTION CONTROL IN SENSOR NETWORKS

2.1 Types of Congestion

Depending on the location of congestion, congestion can be classified into source congestion, sink congestion, and forwarder congestion.

- *Source Congestion*: As soon as an event occurs, it will be detected by all the sensors whose sensing ranges (with radius r) cover the event spot. These nodes will act as *sources*. Since a node's radio range (R) is usually greater its sensing range (for example, $R > 2r$ [12]), these sources will be within each other's radio range as well. If all these source nodes start sending packets at high rates to the sink at the same time, then a hot spot will quickly form around the sources, and a large number of packets will be dropped within this hot spot.

  Source congestion can be eliminated by careful scheduling between these sources which allows only a small subset of nodes (out of all the nodes within the event range) to report to the sink. This has two main advantages. First, the traffic volume within the event range will be significantly reduced while not affecting the accuracy level seen by the application at the sinks significantly because these nodes will be reporting the similar data [10, 11]. Second, if we only allow a subset of nodes to be active (i.e., both sensing and communication) at any time period, then we can save a tremendous amount of energy, and further extend the network lifetime [13, 12].

- *Sink Congestion*: When the sensors report an event at a high data rate, sink nodes (and the nodes around them) will experience a high traffic volume, as also mentioned in [8]. If a hot spot occurs around a sink, the packets will be dropped within the congested area near the sink; dropping a packet around the sink has a much worse impact on the entire network lifetime because a considerable amount of energy has already been consumed by the nodes along the routing path to ship the packet from the source. Another serious side effect is that the battery power from all the nodes that are around the sink will be drained quickly, making the sink unreachable from the rest of the network.

  An effective way of alleviating sink congestion is to deploy multiple sinks that are uniformly scattered across the sensor field, and then balance the traffic between these sinks [8].

- *Forwarder Congestion*: A flow refers to a pair of source and sink and all the corresponding intermediate forwarding nodes. A sensor network will have more than one flow, and these flows will intersect with one another. The area around the intersection will likely become a hot spot. Please note that intersecting flows do not necessarily have distinct sources and sinks; they can share the same source or sink and thus share segment(s) of the routing path. In a tree-like communication paradigm, every intermediate node in the tree can suffer from forwarder congestion. Compared to the other two scenarios, Forwarder congestions are far more challenging because it is very difficult to predict the intersection points due to the network dynamics.

2.2 Congestion Detection

To detect congestion, the level of congestion should be quantified to provide a fine-grained congestion control. We define two types of congestion levels in sensor networks as follows.

- per-node congestion level
- per-flow congestion level

The per-node congestion depicts the local congestion level each individual node perceives. To measure the per-node congestion level, each node investigates the statistics on several metrics such as queue length, packet drop rate, channel loading, etc. For example, ESRT [10] uses buffer utilization alone. However, if the wireless channel is unreliable either by the unreliable MAC protocol (e.g. when the maximum number of retransmissions is small) or by the unstable channel condition, the queue length or packet drop rate cannot be a metric for congestion detection. This is because even though a node's incoming traffic volume exceeds the outgoing channel capacity, the queue length remains small since packets quickly leave the queue due to collision and no subsequent retransmission after the collision. Therefore, in CODA [11], each node measures its congestion level based on the perceived channel loading as well as its queue length.

In general, the per-node congestion level measurement function f returns a positive real number indicating the congestion level based on the metrics of $m_1, m_2, ..., m_n$ as follows.

$$L_i = f(m_1, m_2, ..., m_n) \quad (1)$$

where L_i indicates the node i's congestion level.

A flow includes a source, a sink, and all the intermediate nodes between them. The data generated at a source traverse several intermediate nodes before they arrive at the sink. The delivered data volume to the sink is bottlenecked by the node whose per-node congestion level is the highest along the flow path. Therefore, the per-flow congestion level, denoted as L^{flow}, is defined to be the highest per-node congestion level of all the nodes in the flow and represented as follows.

$$L^{flow} = max(L_1, L_2, ..., L_n) \quad (2)$$

where L_i indicates node i's per-node congestion level. The per-flow congestion level can be calculated in a distributed manner by each node of a flow. The source records its current per-node congestion level in the header of a data packet destined for the sink. Each subsequent node compares its per-node congestion level with the per-flow congestion level

recorded in the header of the data packet forwarded from the previous hop. If its per-node congestion level is greater than the per-flow congestion level, then the node updates the per-flow congestion level in the header of the data packet with its per-node congestion level and forwards it to the next hop. If its per-node congestion is less than the per-flow congestion level, the node simply forwards the packet without modifying the per-flow congestion level. Therefore, the highest per-node congestion level is passed downstream towards the sink. The sink may enforce an end-to-end congestion control based on the per-flow congestion level.

2.3 Congestion Control

When congestion is detected by a node, the node should first decide whether the traffic control or the resource control is performed. We suggest three criteria shown below for choosing between the traffic control and the resource control to alleviate congestion.

- *availability of resource*: If there is no *extensible* resource during congestion, the incoming traffic should be throttled.
- *accuracy level requirement*: The required accuracy level is imposed by the deployed application that extracts the delivered packets from the sinks and makes a decision based on the extracted data. The accuracy level is frequently correlated with the data volume received at the sinks. Therefore, each intermediate node should provide a certain level of throughput during the event period.

  If the achieved throughput is lower than the required accuracy level, the node should perform the resource control to increase the available resource amount, so that it can enhance its throughput, thereby satisfying the accuracy level requirement imposed by the application.
- *energy consumption*: The remaining energy in each node is the most scarce resource in sensor networks. The energy depletion in the node shortens the lifetime of a network by breaking a routing path or reducing the accuracy level. The resource control generally consumes more energy than the traffic control due to its maintenance overhead of the increased resource provisioning and the subsequent higher data transmission rate. Therefore, when the additional energy consumption incurred by the resource control exceeds some threshold, the node should switch to the traffic control to conserve its remaining energy.

Most of previous studies on congestion control in sensor networks focus on the traffic control by throttling the traffic from the previous node (hop-by-hop traffic control) or the source node (end-to-end traffic control) [10, 11, 14].

Next section, in an effort to enforce the resource control during congestion, we investigate how much end-to-end channel capacity can be increased by dynamically creating multiple paths or changing their topology under various congestion scenarios.

3. RESOURCE CONTROL IN SENSOR NETWORKS

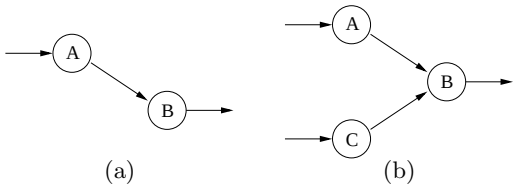


Figure 1: Congestion by merging traffic: (a) before congestion and (b) after congestion

3.1 Resources in Sensor Networks

In sensor networks, various types of resources exist such as:

- (end-to-end) channel capacity
- remaining energy
- active nodes (i.e. whose radio is on.)
- radio transmission power
- packet buffers in the queue (i.e. memory)

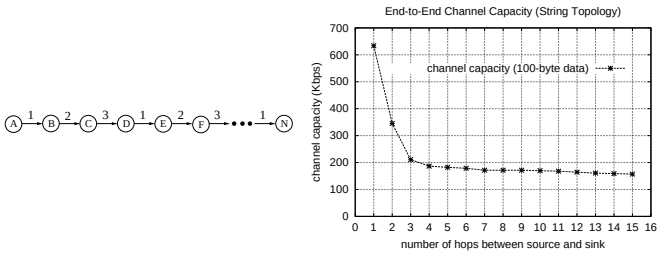
Resources in sensor networks are correlated in the sense that changing the availability of one resource affects the availability of other resources. For example, if the transmission power of a node increases, the remaining energy of a node is drained quickly and the channel capacity is affected. If the channel capacity increases, the packet buffer occupancy is also lowered.

3.2 Adjusting End-to-End Channel Capacity

Due to the elastic availability of resources in sensor networks, congestion can happen even when the incoming traffic remains constant. Figure 1 (a) shows an intermediate node A forwards its incoming traffic to the neighbor node B. Now let's assume that the nearby node C starts forwarding its incoming traffic to node B while node A's outgoing traffic remains unchanged as shown in Figure 1 (b), and node A and C's aggregate outgoing traffic exceeds node B's outgoing channel capacity. If this is a wired network, the incoming packets are dropped at node B due to the queue overflow. In the sensor network, however, that is configured in an ad-hoc fashion, node A and C experience congestion and most incoming traffic is discarded at node A and C not because the incoming traffic is increased at node A and C, but because the channel capacity available to node A or C has been reduced due to the sudden contention from its neighbor node. In this example, congestion can be alleviated by re-routing the incoming traffic on the non-interfered path, which effectively increases the channel capacity available to the congested node.

Since the effective channel capacity of a node is decided by many factors such as the deployed MAC protocol, raw bit rate, transmission power, contention level, etc, the end-to-end channel capacity adjustment can be done in many ways, for example by power control [7, 3], in the real sensor network. In this section, we investigate how much end-to-end channel capacity can be adjusted by re-routing the existing path or creating additional paths under the two typical congestion scenarios: merging traffic and crossing traffic.

Related with this issue, following questions can be asked.



(a) (b)

Figure 2: End-to-end channel capacity of a string topology

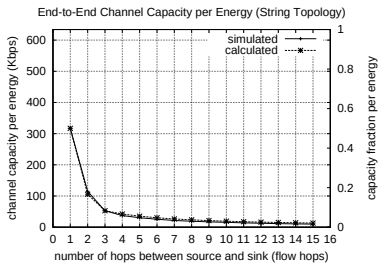


Figure 3: End-to-end channel capacity per energy of a string topology

- What is the maximum attainable end-to-end channel capacity of a flow in a non-interfered environment ?
- To accommodate a certain amount of incoming traffic during congestion, should a node create an additional path or re-route the existing path ?
- How should multiple flows' paths be configured to increase each flow's effective channel capacity ?

To answer these questions, we first investigate the end-to-end channel capacity of a string topology. In [6], they already measure the end-to-end channel capacity of a string topology, but not in the context of congestion control.

3.2.1 End-to-end Channel Capacity of Single Flow

Gupta and Kumar [5] show that when n fixed nodes are randomly deployed and pick a random destination, the attainable end-to-end throughput per flow is $\Theta(\frac{1}{\sqrt{nlogn}})$. This is a somewhat pessimistic result since the achievable per-flow throughput goes to zero as the network scales. In [4], they try to increase the end-to-end channel capacity by introducing mobility into the network model. However, as already indicated by [6], if local communication dominates, the path length between the source and the sink could remain nearly constant even as the network grows, leading to constant per-node attainable channel capacity.

A string topology and its measured end-to-end channel capacity are shown in Figure 2. Only adjacent nodes are within each other's radio range. IEEE 802.11 DCF with RTS/CTS is used for MAC protocol. The size of a data packet is 100 bytes. Figure 2 (b) shows the end-to-end channel capacity of a flow decreases quickly until the *flow hops*, which is defined as the number of hops between the source and the sink, reach 3 and stabilizes after that. This can be explained by simply assigning a time frame number, during which adjacent nodes exchange a data packet and corresponding control packets such as RTS, CTS, and ACK, to each transmission as shown in Figure 2 (a). When the flow hops are greater than 3, existing time frames can be reused concurrently. This spatial re-use of the spectrum thus prevents the abrupt throughput degradation. We call the minimally required number of time frames *interference factor* of the topology, which is 3 in Figure 2 (a). Interference factor is decided by the radio ranges of nodes and their topology. Throughout this paper, we use the interference factor of 3. We also define the reciprocal of the interference factor to be *capacity fraction*. The capacity fraction of Figure 2 (a) is $\frac{1}{3}$, indicating its end-to-end capacity cannot exceed $(C^{max} * \frac{1}{3})$, where C^{max} is the maximum attainable channel capacity imposed by the deployed MAC protocol. C^{max} is the 1 hop channel capacity, which is 634 Kbps in Figure 2 (b). From this observation, we have a following rule.

Rule 1 Minimizing the flow hops does not increase the end-to-end channel capacity much if the resulting topology has the same capacity fraction. □

Figure 2 (b) also indicates a non-interfered string topology guarantees some level of end-to-end channel capacity if local communication predominates. This means if the maximum flow hops of a flow in the network is upper-bounded by some number of hops, the end-to-end channel capacity of a string topology is lower-bounded by some capacity, which we define C^{min}. In Figure 2 (b), if the maximum flow hops of any flow are less than 15, C^{min} is around 160 Kbps. Therefore, we have a following rule.

Rule 2 If the incoming traffic volume is greater than the available channel capacity and less than C^{min}, then congestion is eliminated by re-routing the data traffic on the *non-interfered* path whose end-to-end capacity is at least C^{min}. □

Since energy is the most scarce resource in sensor networks, it is important how much capacity can be attainable for the given energy budget. Figure 3 shows the simulated and calculated *capacity per energy*. We assume the energy is consumed in proportion to the total number of nodes involved in the communication. Therefore, the simulated capacity per energy in Figure 3 is drawn relative to C^{max} by dividing the end-to-end channel capacity by the number of the deployed nodes. For example, the simulated capacities per energy of two-node, three-node, four-node, five-node string topologies are $\frac{634}{2}$Kbps, $\frac{345}{3}$Kbps, $\frac{210}{4}$Kbps, $\frac{186}{5}$Kbps, respectively. The calculated capacity per energy in Figure 3 is drawn relative to the capacity fraction of 1 by dividing the capacity fraction by the number of deployed nodes. For example, the calculated capacities per energy of two-node, three-node, four-node, five-node string topologies are $\frac{1}{2}, \frac{1}{6}, \frac{1}{12}, \frac{1}{15}$, respectively. Two graphs are exactly overlapped.

These results show minimizing the flow hops does not improve the capacity per energy much if the resulting topology has the same capacity fraction even though the number of deployed node is reduced.

3.2.2 End-to-End Channel Capacity of Splitting Flows

If we allow each source to report an event at the rate greater than C^{min} and the flow hops are sufficiently large, a source should create multiple paths and split the outgoing traffic onto these paths to forward a large volume of traffic. The maximum number of non-interfering data paths a node can create is bounded by C^{max}. Therefore,

Rule 3 The maximum number of non-interfering outgoing paths from a node is $\lfloor \frac{C^{max}}{C^{min}} \rfloor$ if each path has the end-to-end capacity of C^{min}. □

$\lfloor \frac{C^{max}}{C^{min}} \rfloor$ in Figure 2 (b) is 3. If each of these multiple paths is interfered by other traffic and thereby has the end-to-end channel capacity of less than C^{min}, more than $\lfloor \frac{C^{max}}{C^{min}} \rfloor$ paths can be established. However, the maximum number of non-interfering outgoing paths of a node is limited not only by the maximum attainable capacity but also by the geographic proximity of candidate paths.

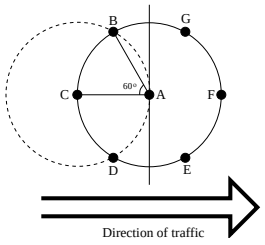


Figure 4: Nodes on the boundary of each other's radio range

Figure 4 shows the number of neighbor nodes that are on the each other's radio range boundary is 6. Therefore, we have a following rule.

Rule 4 If the radio range of each node is a identical circle, the maximum number of non-interfering outgoing paths from a node is 5. □

Due to the interference radio range which is always larger than the communication radio range of a node, the maximum number of non-interfering outgoing paths of a node can be smaller than 5. In addition, if the node in Figure 4 is an intermediate node, it can have at most 4 non-interfering outgoing paths, excluding one neighbor from which the incoming traffic is coming. If we make a more conservative assumption that the outgoing paths are not set up in the reverse direction of the incoming traffic by the deployed routing protocol [9], then the maximum number of outgoing paths will be 3, which are the paths traversing the neighbor nodes E, F, and G in Figure 4.

When a node creates additional forwarding paths and splits its incoming traffic over the multiple paths, the split traffic can be destined for either the same sink or different sinks. From the application point of view, the accuracy level is the same as long as the same amount of data is delivered to one or more sinks.

Figure 5 (a), (b), (c), and (d) show two non-interfering paths split at the different node, having the *pre-splitting hops* of 0, 1, 2, and 3, respectively. We schedule the transmissions using as small number of time frames as possible, as

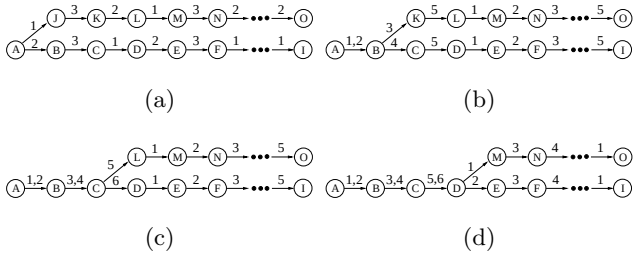


Figure 5: Traffic splitting at different nodes

was done in Figure 2 (a). For example, in Figure 5 (d) node D receives two data packets from node C at time frame 5 and 6, respectively. Then, node D forwards these two packets to node M and E at time frame 1 and 2, respectively. The sinks (node O and I) will eventually receive two packets in 6 time frames. The optimal time frame assignments are not unique. The time frame assignments in Figure 5 (a), (b), (c), and (d) give the aggregate capacity fractions (i.e. sum of two flows' capacity fractions) of $\frac{2}{3}, \frac{2}{5}, \frac{2}{6}, \frac{2}{6}$ and the per-flow capacity fractions of $\frac{1}{3}, \frac{1}{5}, \frac{1}{6}, \frac{1}{6}$, respectively.

This calculation shows the capacity fraction of the flow remains constant after when the pre-merging hops reach the interference factor (here, 3). While the aggregate capacity fraction of Figure 5 (a) is doubled by having an additional path to a different sink compared to the capacity fraction of a string topology, the capacity fractions of Figure 5 (c) and (d) are the same as that of a string topology, which is $\frac{1}{3}$.

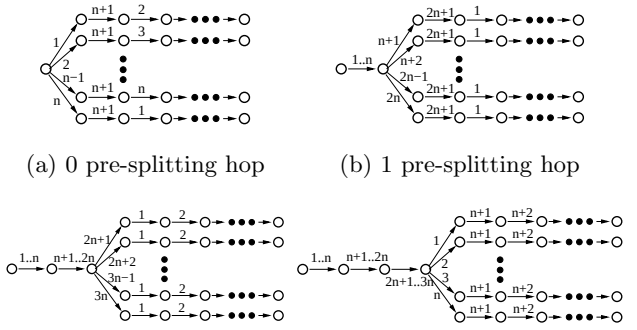


Figure 6: Splitting traffic destined for n sinks

Now, we further extend the time frame assignment in the 2-flow case to the n-flow case. Figure 6 shows the time frame assignments in the n-flow case with respect to different pre-splitting hops. The optimal assignment is not unique and Figure 6 shows one example for each pre-splitting hops. From Figure 6, we have the aggregate end-to-end channel capacity fraction, denoted as C^{agg}, and the per-flow end-to-end channel capacity fraction, denoted as C^{flow}, of n flows with respect to different pre-splitting hops h_s as follows.

$$C^{agg} = \begin{cases} \frac{n}{n+1} & if \ \ h_s = 0 \\ \frac{n}{2n+1} & if \ \ h_s = 1 \\ \frac{n}{3n} = \frac{1}{3} & if \ \ h_s \geq 2 \end{cases} \quad (3)$$

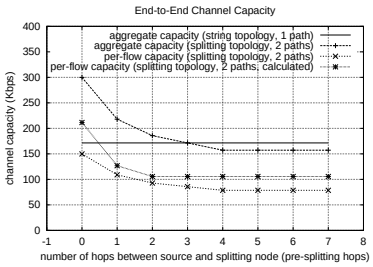


Figure 7: Aggregate end-to-end channel capacity of splitting traffic

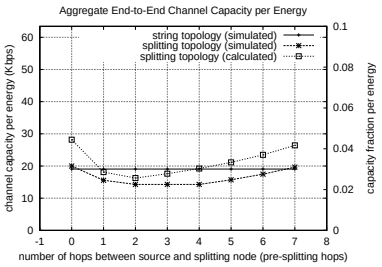


Figure 8: Aggregate end-to-end channel capacity per energy of splitting traffic

$$C^{flow} = \frac{C^{agg}}{n} = \begin{cases} \frac{1}{n+1} & if \ \ h_s = 0 \\ \frac{1}{2n+1} & if \ \ h_s = 1 \\ \frac{1}{3n} & if \ \ h_s \geq 2 \end{cases} \quad (4)$$

where n is the number of the splitting paths (or flows). Above analysis shows the aggregate end-to-end channel capacity remains constant when the pre-splitting hops are greater than 1. This is because the enlengthened pre-splitting hops bottleneck the aggregate end-to-end channel capacity.

Figure 7 shows simulation results of Figure 5 and the calculated per-flow channel capacity with respect to different pre-splitting hops. The flow hops are fixed to 8. The calculated per-flow end-to-end channel capacity is drawn by multiplying C^{max}, which is 634 Kbps in the 100-byte data packet case, by the per-flow capacity fraction calculated earlier.

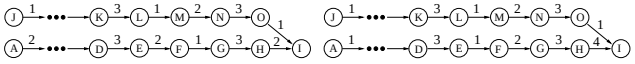
The figure shows the simulated aggregate end-to-end channel capacity decreases significantly as the pre-splitting hops increase and then later approaches that of a string topology. The aggregate end-to-end channel capacity is even lower than that of a string capacity. This is because the head-of-line (HOL) problem arises at the splitting node. For example, in Figure 5 (a) if a head-of-line packet at node A is destined for node J and node J cannot receive the packet due to the interference from node K, then another packet destined for node B in the queue cannot be sent even when node B is idle. Figure 7 also shows that the simulated per-flow end-to-end channel capacity is lower than the calculated aggregate end-to-end channel capacity. This is first because IEEE 802.11 MAC protocol does not optimally schedule packet transmissions and the head-of-line (HOL) problem arises at the splitting node. Above observation gives a following rule.

Rule 5 When an additional path is created to the existing path, the aggregate end-to-end channel capacity increases only when the pre-splitting hops are less than the interference factor. □

To investigate the energy efficiency, the capacity per energy in the splitting flow case is simulated and calculated as shown in Figure 8. The capacities per energy of splitting topology in the simulated and calculated cases are *bi-modal* with respect to the pre-splitting hops. This is because as the pre-splitting hops increase, the end-to-end channel capacity remains constant while consuming less energy. The simulated capacity per energy in the splitting traffic is always lower than that of the string topology due to the interference and the HOL problem. This indicates the energy is the best exploited in the string topology.

3.2.3 End-to-End Channel Capacity of Merging Flows

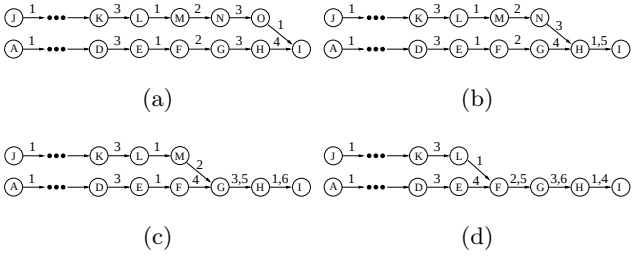
When several sources report an event to a sink at the same time, these traffic will eventually merge at some node. Depending on the traffic volume, congestion might happen around the merging node.



(a) (b)

Figure 9: (a) An ideal time frame assignment and (b) a realistic time frame assignment for merging flows

Figure 9 shows two source nodes A and J generate packets destined for the same sink node I and the traffic from node A and J are merged at node I. In Figure 9 (a), the time frame numbers are assigned from the sink to the sources, using 3 time frames to forward 2 packets to the sink. In Figure 9 (b), the time frame number are assigned from the sources to the sink, using 4 time frames to forward 2 packets to the sink. Since each source is not aware of the topology of downstream nodes, the time frame number assignment in Figure 9 (b) is more realistic. Our simulation also verifies this. Therefore, we adopt the time frame number assignment used in Figure 9 (b) for the merging flows.



(a) (b)

(c) (d)

Figure 10: Traffic merged at different nodes

Figure 9 (a), (b), (c), and (d) show two non-interfering paths merging at the different node, having the *post-merging hops* of 0, 1, 2, and 3, respectively. The realistic time frame numbers used in Figure 9 (b) is assigned to each transmission. Figure 9 (a), (b), (c), and (d) have the aggregate capacity fractions of $\frac{2}{4}$, $\frac{2}{5}$, $\frac{2}{6}$, and $\frac{2}{6}$ and the per-flow capacity fractions of $\frac{1}{4}$, $\frac{1}{5}$, $\frac{1}{6}$, and $\frac{1}{6}$, respectively. This calculation

shows the capacity fraction of each flow remains constant after when the post-merging hops are 3.

If we extend the time frame assignment in the 2 flow case to the n-flow case as was done in the merging flow case, the aggregate end-to-end channel capacity fraction C^{agg} and the per-flow end-to-end channel capacity fraction C^{flow} with respect to different post-merging hops h_m are represented as follows.

$$C^{agg} = \begin{cases} \frac{n}{n+2} & if \quad h_m = 0 \\ \frac{n}{2n+1} & if \quad h_m = 1 \\ \frac{n}{3n} = \frac{1}{3} & if \quad h_m \geq 2 \end{cases} \tag{5}$$

$$C^{flow} = \frac{C^{agg}}{n} = \begin{cases} \frac{1}{n+2} & if \quad h_m = 0 \\ \frac{1}{2n+1} & if \quad h_m = 1 \\ \frac{1}{3n} & if \quad h_m \geq 2 \end{cases} \tag{6}$$

where n is the number of the merging paths. Figure 11 shows simulation results of Figure 9 and the calculated end-to-end channel capacity with respect to different post-merging hops.

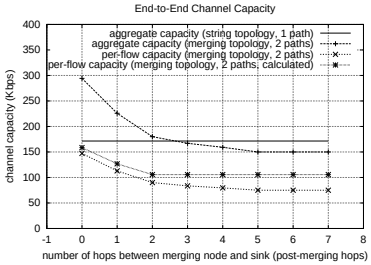


Figure 11: Aggregate end-to-end channel capacity of merging traffic

The two flows' end-to-end hops are fixed to 8 irrespective of the post-merging hops. The results are similar to those in the splitting flow case.

If we compare Figure 9 (c) and (d), their capacities are similar, but Figure 9 (c) consumes more energy by exploiting one more node. The capacity per energy in the merging flows is similar to that in the splitting flow case. Therefore, we have a following rule.

Rule 6 If the resulting post-merging hops are greater than the interference factor, reducing the post-merging hops does not increase the end-to-end channel capacity much while consuming more energy. □

Figure 12 (a) shows an congestion scenario when 4 sources generate packets to the same sink, node J. Congestion happens at the merging nodes. In Figure 12 (b), it is exemplified that congestion can be alleviated by re-routing the data path onto the non-interfered path. When the sink is congested in Figure 12 (b), each source can re-route the data packets to alternative sinks.

It is not trivial to accurately quantify the end-to-end capacities of various merging topologies, especially in the interference model. In fact, finding the optimal distributed scheduling of communication is known to be a NP-complete problem [2]. Therefore, this issue is left for the future work.

3.2.4 End-to-End Channel Capacity of Crossing Flows

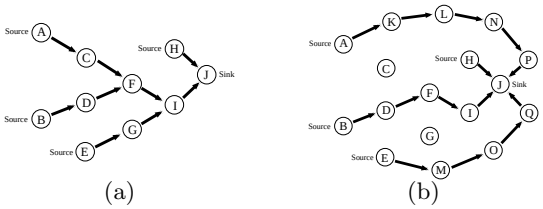


Figure 12: (a) Congestion by merging traffic and (b) congestion alleviation by re-routing

When two flows destined for different sinks intersect each other, congestion might happen around the crossing node depending on the traffic volume.

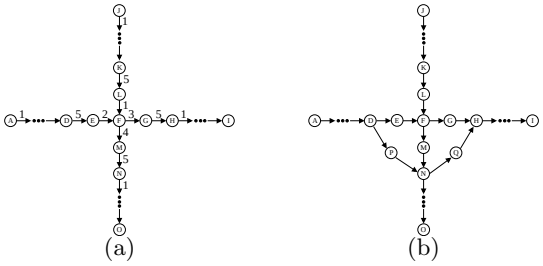


Figure 13: (a) Forwarder congestion by crossing traffic and (b) congestion alleviation by an additional path

Figure 13 (a) shows two crossing flows have the aggregate capacity fraction of $\frac{2}{5}$ if the flow hops are sufficiently large. This is 20 % greater than the capacity fraction of a string topology, $\frac{1}{3}$. However, Figure 14 shows the simulated aggregate capacity of two flows, whose flow hops are 8 each, is slightly less than the capacity of a string topology of 8 hops. This is because cross traffic experiences not only contention around the hotspot when they are merged into the crossing node but also the head-of-line (HOL) problem when they are split from the crossing node. In Figure 13 (a), if a head-of-line packet at node F is destined for node G and node G cannot receive the packet due to the interference from node H, then another packet destined for node M cannot be sent even when node M is idle.

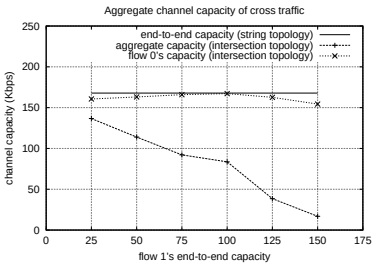


Figure 14: The capacity of one flow with respect to different data rates of the other flow in the cross traffic case

Figure 14 also shows the aggregate channel capacity around the crossing node is almost constant and approximates to C^{min} irrespective of two flows' traffic rates. We only allow

5 % of packet drop rate per each flow when measuring the maximum channel capacity of each flow. The simulations with different flow hops have similar results. Therefore,

Rule 7 If flow A and B whose flow hops are sufficiently large intersect each other and flow A's traffic rate is k, then, flow B's perceived end-to-end channel capacity without hurting flow A approximates to $C^{min} - k$. □

In Figure 13 (b), the horizontal flow can increase its available channel capacity by creating an additional path that also intersects the vertical path. Unlike the merging and splitting traffic, the additional path is not only split from the original path but also merged into the downstream node in the original path. Figure 15 shows a general topology with N multiple path sharing the same splitting node and merging node.

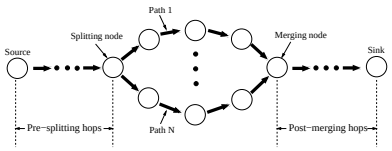


Figure 15: N multiple paths sharing the same splitting node and merging node

As already indicated by **Rule 5** and **Rule 6**, the additional paths do not increase the end-to-end channel capacity if the pre-splitting hops or the post-merging hops exceed the interference factor. In addition, the multiple paths should be created not to interfere each other.

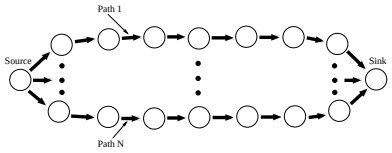


Figure 16: N multiple paths with zero pre-splitting hops and zero post-merging hops

Figure 16 shows that multiple paths are created with zero pre-splitting and zero post merging hops. If the paths don't interfere each other, this configuration maximizes the end-to-end channel capacity. The maximum number of non-interfering outgoing and incoming paths is limited as already indicated by **Rule 3** and **Rule 4**.

Multiple paths can be created at different splitting points and merging points. The end-to-end channel capacities of those various topologies are left for the future work.

4. RELATED WORK

Gupta and Kumar [5] show that if n nodes are optimally placed in a disk of unit area, traffic patterns are optimally assigned, and each transmission range is optimally chosen, the throughput for each node for a far away destination is only $\Theta(\frac{1}{\sqrt{n}})$.

In [6], they examine the capacity of wireless ad hoc network by simulations and analysis. They show that as the expected path length increases, the bandwidth available for each node to originate packets decreases, having a $O(\frac{1}{\sqrt{n}})$. This implies that the traffic pattern has a great impact on scalability. Therefore, a large ad hoc networks are feasible only when the locality of traffic predominates.

Grossglauser and Tse [4] show that per-user throughput can increase dramatically when nodes are mobile rather than fixed. Under the delay-tolerant application, nodes distribute their packets to nearby neighbor nodes and the packets are forwarded to the ultimate destination when the relaying nodes are closer to the destination.

Chiang [3] present a distribute power control algorithm to increase the end-to-end throughput and energy efficiency in wireless multihop networks. He proves that power control and congestion control can be converged to the global optimization for both synchronized and asynchronous implementations. He uses the buffer occupancy information as an indication of congestion.

5. CONCLUSION AND FUTURE WORK

In this paper, we investigate the end-to-end channel capacity of a flow with multiple paths and their configurations under several congestion scenarios. The end-to-end channel capacity quantification is important not only for congestion control but also for the network design such as calibrating the transmission power, deciding the number of deployed sensors, controlling duty cycles of nodes. The quantified channel capacity can also be a basis for routing protocols to find a routing path that has a higher channel capacity.

In this paper, however, the end-to-end channel capacity of various configurations are not extensively studied. In practice, due to the various congestion scenarios and interference factors, it is very difficult to expect the end-to-end channel capacity accurately. As such, this paper should be viewed as an incipient work on this potentially complex research topic. The followings are the issues for the future works.

- During congestion, several flows interfere each other, affecting each other's effective channel capacity. Therefore, the end-to-end channel capacity under the various interference models should be further studied.
- As mentioned before, the end-to-end channel capacity is decided by many factors. The channel capacity in conjunction with various MAC protocols and their parameters has been rigorously studied. However, changing these MAC parameters to increase the channel capacity during congestion has many potentials for congestion alleviation. For example, in [1], they change the contention window of IEEE 802.11 protocol to adapt to the congestion level.
- Increasing the end-to-end channel capacity during congestion is one example of the resource control. Transmission power control, dynamic modulation scheme, compression in conjunction with the dynamic congestion level should be further explored.

6. REFERENCES

[1] I. Aad, Q. Ni, C. Barakat, and T. Turletti. Enhancing IEEE 802.11 MAC in congested environments. In *Proceedings of 4th Workshop on Applications and Services in Wireless Networks*, August 2004.

[2] E. Arikan. Some Complexity Results about Packet Radio Networks. *IEEE/ACM Transactions on Information Theory, 30(4):681-685*, July 1984.

[3] M. Chiang. To Layer or Not To Layer: Balancing Transport and Physical Layers in Wireless Multihop Networks. In *Proceedings of IEEE INFOCOM*, March 2004.

[4] M. Grossglauser and D. Tse. Mobility Increases the Capacity of Ad-Hoc Wireless Networks. In *Proceedings of IEEE INFOCOM*, April 2001.

[5] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. In *IEEE Transactions on Information Theory, 46(2):388-404*, March 2000.

[6] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris. Capacity of Ad Hoc Wireless Networks. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2001)*, July 2001.

[7] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar. Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW Protocol. In *Proceedings of the European Wireless Conference - Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, February 2002.

[8] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu. Data-Centric Storage in Sensornets with GHT, a Geographic Hash Table. *Mobile Networks and Applications (MONET), Journal of Special Issues on Mobility of Systems, Users, Data, and Computing: Special Issue on Algorithmic Solutions for Wireless, Mobile, Ad Hoc and Sensor Networks*, 8(4):427–442, August 2003.

[9] E. M. Royer and C. K. Toh. A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. *IEEE Personal Communications*, 6(2):46–55, April 1999.

[10] Y. Sankarasubramaniam, O. Akan, and I. Akyildiz. ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks. In *proceedings of the ACM MobiHoc Conference*, 2003.

[11] C. Wan, S. Eisenman, and A. Campbell. CODA: Congestion Detection and Avoidance in Sensor Networks. In *proceedings of the ACM SenSys 2003*, 2003.

[12] T. Yan, T. He, and J. A. Stankovic. Differentiated Surveillance Service for Sensor Networks. In *Proceedings of the ACM SenSys 2003*, 2003.

[13] F. Ye, G. Zhong, S. Lu, and L. Zhang. PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*, May 2003.

[14] Y. Yi and S. Shakkottai. A Hop-by-hop Congestion Control over a Wireless Multi-hop Network. In *Proceedings of IEEE INFOCOM*, March 2004.