A CONGESTION CONTROL SCHEME
FOR WIRELESS SENSOR NETWORKS

A Thesis

by

YUNLI XIONG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2005

Major Subject: Electrical Engineering

A CONGESTION CONTRL SCHEME
FOR WIRELESS SENSOR NETWORKS

A Thesis

by

YUNLI XIONG

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Scott L. Miller
(Chair of Committee)

Deepa Kundur
(Member)

Narasimha Reddy
(Member)

Jennifer L. Welch
(Member)

Chanan Singh
(Head of Department)

May 2005

Major Subject: Electrical Engineering

ABSTRACT

A Congestion Control Scheme
for Wireless Sensor Networks. (May 2005)
Yunli Xiong, B.S.; M.S., Shanghai Jiao Tong University
Chair of Advisory Committee: Dr. Scott L. Miller

In wireless sensor networks (WSN), nodes have very limited power due to hardware constraints. Packet losses and retransmissions resulting from congestion cost precious energy and shorten the lifetime of sensor nodes. This problem motivates the need for congestion control mechanisms in WSN.

In this thesis, an observation of multiple non-empty queues in sensor networks is first reported. Other aspects affected by congestion like queue length, delay and packet loss are also studied. The simulation results show that the number of occupied queues along a path can be used to detect congestion.

Based on the above result, a congestion control scheme for the transport layer is proposed in this thesis. It is composed of three parts: (i) congestion detection by tracking the number of non-empty queues; (ii) On-demand midway non-binary explicit congestion notification (CN) feedback; and (iii) Adaptive rate control based on additive increase and multiplicative decrease (AIMD).

This scheme has been implemented in ns2. Extensive simulations have been conducted to evaluate it. Results show that it works well in mitigating and avoiding congestion and achieves good performance in terms of energy dissipation, latency and transmission efficiency.

To my Wife Bei Xu and my Parents

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Scott L. Miller, for his invaluable guidance and great support throughout my study and research.

I am also thankful to my committee members, Dr. Deepa Kundur, Dr. Narasimha Reddy, and Dr. Jennifer L. Welch for their valuable time and comments.

I would like to thank my wife for her constant support and confidence in me.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

In recent years, due to advances in low-power circuit and radio technologies, wireless sensor networks emerged and have received a lot of attention. A typical sensor network is formed by a large amount of nodes. There usually is no pre-determined topology for a sensor network. Instead, these sensor nodes construct and dynamically maintain the structure of the network through wireless communication.

Sensor nodes have restricted power. They are usually equipped with batteries. In many cases, replenishment of the power resource is impossible. This nature imposes the requirement of energy-efficiency on all layers of protocols. Besides, sensor nodes are also constrained by relatively weaker processors and limited memory.

Sensor networks have a variety of applications. The research of sensor networks was initially driven by military applications like battlefield surveillance and enemy tracking. Afterward, this technology was introduced into civilian sectors. Habitat monitoring, environment observation and forecast system are categories of such applications.

For applications where a sensor node reports sensed conditions of a region to one or a couple of sink nodes, sensor networks work with a light load most of the time. But, when an interesting event occur, such as enemy intrusion, the network will generate and need to transmit a sudden huge amount of data. In such cases, congestion control is of great importance. It can reduce the delay and save precious energy by regulating the transmitting rates.

Compared with wired networks, the way congestion occurs in wireless sensor

---

The journal model is *IEEE Transactions on Automatic Control.*

networks (WSN) is different. In addition, WSN will exhibit its own phenomena when congested. Existing protocols for wired networks like TCP are not suitable for WSN.

In this thesis, a congestion control scheme dedicated to sensor networks is addressed. It takes advantage of the characteristic of multiple non-empty queues in time of congestion and achieves energy-efficiency and short latency.

CHAPTER II

BACKGROUND

A. Congestion Control

In computer networks, mismatch of incoming and outgoing data rates results in congestion. For wired networks like INTERNET, there are mixed links with different bandwidths. The node with the lowest bandwidth along a path from the source to the destination is called the bottleneck. Usually, congestion occurs in the bottleneck since it receives more data than it is capable of sending out. In this situation, packets will be queued and sometimes get dropped. As a consequence, response time will increase and throughput will also degrade.

1. What Is Congestion Control?

Figure 1 [1] illustrates network performance as a function of the load. When the load is light, throughput is linearly proportional to the load and response time is almost unchanged. After the load reaches the network capacity (the knee point), throughput won't increase much with the load. Instead, packets will be queued and the response time will become longer in this period. The throughput may suddenly drop if packets get discarded due to buffer overflow, which is called the cliff point as shown in Figure 1.

Congestion control is necessary in avoiding congestion and/or improving performance after congestion. It aims to make the network operate around the knee point in Figure 1. Congestion control schemes are usually composed of three components: congestion detection, congestion feedback and sending-rate control.

The criteria for congestion vary with protocols. Congestion can be determined

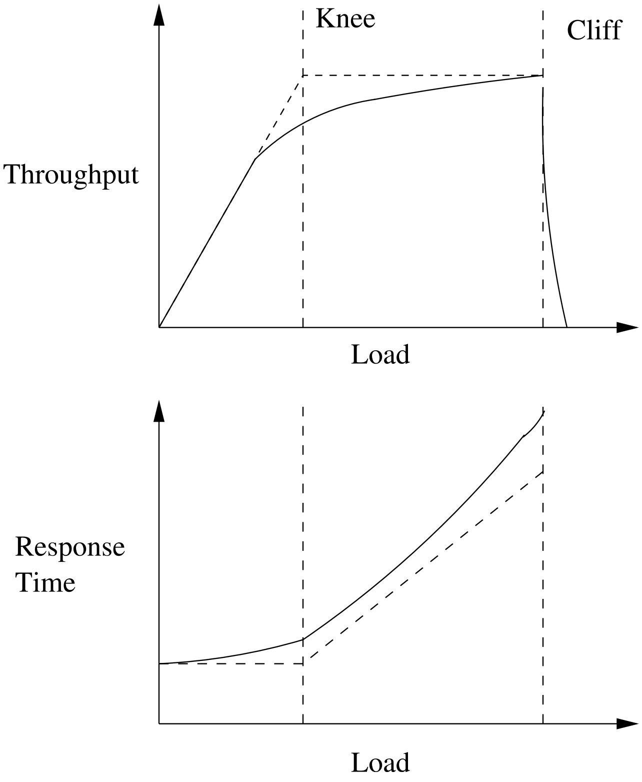

Fig. 1. Network Performance versus Load

by checking queues length. It can also be indirectly detected by monitoring the trend of throughput or response time, as depicted in Figure 1. In addition, packet loss can also be a criterion of congestion in wired networks. Practically, congestion detection can be processed in intermediate nodes or receivers.

Congestion feedback mechanisms can be categorized in several ways. In one way, they are classified into explicit or implicit feedback. Explicit feedback means that feedback is sent to the sender in an explicit form, like a dedicated bit. In implicit feedback, feedback information doesn't occupy any dedicated bits. It is realized by piggyback. The well-known example of implicit feedback is TCP, where 3-Acknowledgments implies congestion. From the aspect of information carried by feedback, they can be categorized into binary or non-binary feedback. Binary feedback can only tell if there is congestion or not. In contrast, non-binary feedback carries more information, which can indicate the congestion level. TCP is a binary feedback mechanism.

The rate control function is usually viewed as a distributed decision-making problem. Additive increase and multiplicative decrease (AIMD) is proved to be a feasible linear control algorithm by Dah-Ming Chiu [1], according to the criteria of efficiency, fairness and convergence. This algorithm is represented by.

$$\lambda_{t+1} = \begin{cases} a_I + \lambda_t & \text{Increase;} \\ b_D \cdot \lambda_t & \text{Decrease;} \end{cases} \tag{2.1}$$

where, λ_t denotes sending rate at the discrete time point t, a_I and b_D are parameters which satisfy

$$\begin{aligned} a_I &> 0 \\ 1 > \quad b_D &\geq 0 \end{aligned} \tag{2.2}$$

In practice, AIMD can be implemented as window-based or rate-based mechanisms. In a window-based scheme, there is a limit to the number of outstanding packets for a sender. A rate-based mechanism controls the rate at which packets can be sent out. TCP is a typical example of window-based control.

2. Dynamical Model for Rate-Based Flow Control

To analyze a rate-based congestion control, a dynamical model [2] shown in Figure 2 for a single unicast connection is built.

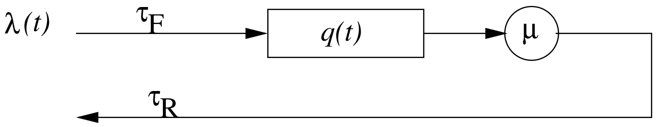

Fig. 2. Single-bottleneck Model of a Single Connection

Table 2 gives the description of those parameters in Figure 2.

Table I. Parameters Description for Single Connection Model

Parameter	Description
$\lambda(t)$	The source's sending rate
τ_F	The propagation delay from the source to the bottleneck (forward delay)
$q(t)$	The bottleneck's queue size at time t
μ	The bottleneck's service rate
τ_R	The propagation delay from the bottleneck to the source (backward delay)

Assume that there is one link with the lowest bandwidth along single connection. This slowest link is modeled as a bottleneck node. Other high-speed links are represented by propagation delays. In this model, there are three other assumptions as follows.

(a) The source's sending rate is continuous.

(b) The forward and backward propagation delays are fixed.

(c) The bottleneck has an infinite-length buffer.

Although there are different feedback mechanisms, it is assumed that one bit information is fed back to the source. It can tell if there is a congestion by comparing $q(t)$ with 0. After a time period τ_R, the source will make a proper decision to increase or decrease $\lambda(t)$. It is also noticed that the data rate entering the bottleneck is actually $\lambda(t - \tau_F)$. $\dot{\lambda}(t)$ and $\dot{q}(t)$ are used to denote the derivatives of $\lambda(t)$ and $q(t)$

respectively. They are given by Equation 2.3 and 2.4.

$$\dot{q}(t) = \begin{cases} 0 & \text{if q(t) = 0 and } \lambda(\mathrm{t} - \tau_{\mathrm{F}}) < \mu; \\ \lambda(t - \tau_F) - \mu & \text{otherwise;} \end{cases} \tag{2.3}$$

$$\dot{\lambda}(t) = \begin{cases} \alpha & \text{if q}(\mathrm{t} - \tau_{\mathrm{R}}) = 0; \\ -\frac{\lambda(t)}{\beta} & \text{if q}(\mathrm{t} - \tau_{\mathrm{R}}) > 0; \end{cases} \tag{2.4}$$

where, α is the linear increasing rate, β is the time constant of the exponential decreasing curve. Their relations with parameters a_I and b_D are

$$\begin{aligned} a_I &= \alpha \cdot \Delta \\ b_D &= e^{-\frac{\Delta}{\beta}} \end{aligned} \tag{2.5}$$

where, Δ is the time interval of decision making in the source.

Figure 3 illustrates the dynamics of $\lambda(t)$ and $q(t)$. The sending rate starts from 0 and linearly increases at the speed α. At time t_0, $\lambda(t_0)$ reaches the service rate μ of the bottleneck. After the forward propagation delay τ_F, packets start getting queued. The congestion notification is received by the source at the time instant $t_1 = t_0 + \tau_F + \tau_R$. Then, the source starts dropping the rate exponentially. As a result, the queue becomes shorter and finally empty. The empty queue triggers a non-congestion feedback. From the time point t_4, a new cycle of congestion control begins.

By solving corresponding equations, the maximal sending rate and queue length can be obtained by:

$$\lambda_{max} = \mu + \alpha\tau \tag{2.6}$$

$$q_{max} = \alpha\frac{\tau^2}{2} + \alpha\beta\tau + \mu\beta\ln(\frac{\mu}{\mu + \alpha\tau}) \tag{2.7}$$

where, $\tau = \tau_F + \tau_R$. The period T of a congestion control cycle is defined as $t_5 - t_0$.

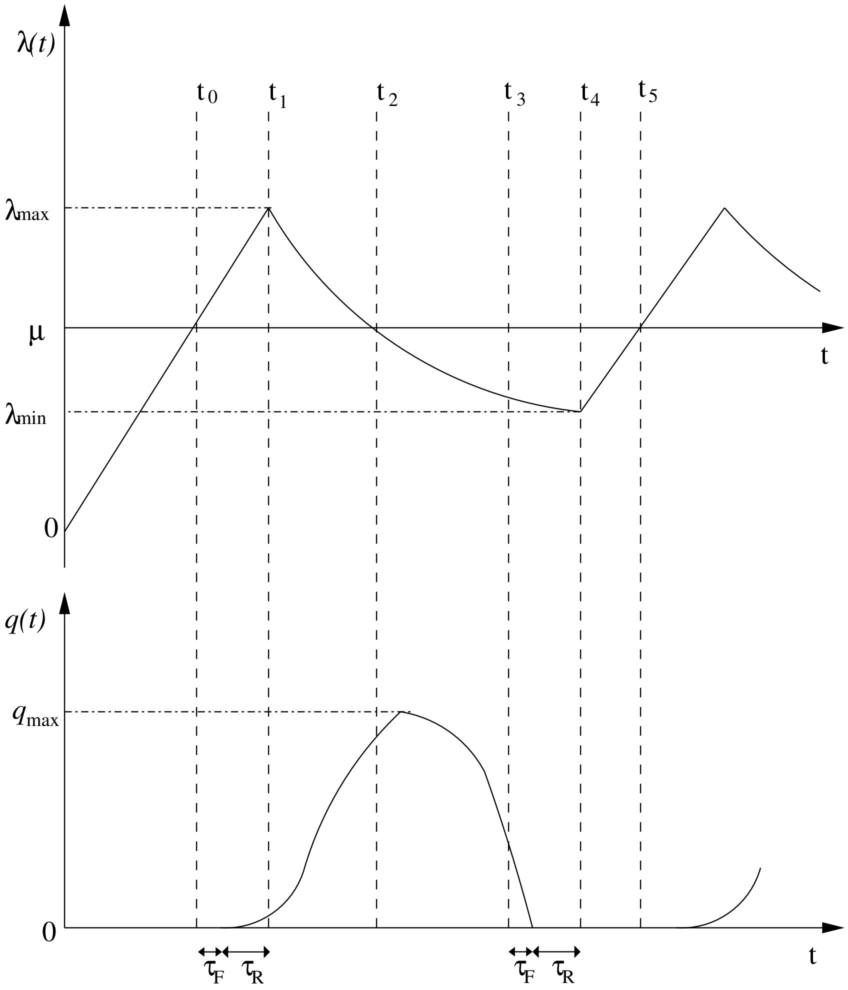

Fig. 3. Dynamics of $\lambda(t)$ and $q(t)$

It is given in Equation 2.8.

$$T = 2\tau + (t_3 - t_1) + \frac{\mu - \lambda_{min}}{\alpha} \tag{2.8}$$

where, $\lambda_{min} = (\mu + \alpha\tau)\exp -\frac{\tau + t_3 - t_1}{\beta}$, and $t_3 - t_1$ is the product of β and the root of the equation $1 - e^{-x} = \frac{\mu}{\mu + \alpha\tau}x + \frac{\alpha\tau^2}{2\beta(\mu + \alpha\tau)}$.

B. Carrier Sensing Multiple Access (CSMA)

In wireless networks, a community of nodes share a single transmission medium. To avoid collision and better utilize the bandwidth, some kind of medium access control (MAC) protocol is needed. Carrier sensing multiple access (CSMA) is a random access protocol, which allows users to transmit data in a none predetermined way.

CSMA schemes require a user to be sure the medium is idle before the transmission. This is called carrier sensing. If the medium is busy, the user has to back-off for a random period and then re-sense. The random period is to minimize collision since other users may also want to take the medium at the same time. Once the channel is idle, the user can start transmission.

In sensor networks, CSMA schemes are practically used, for example, IEEE 802.11 [12] and SMAC [11]. We will discuss a little about IEEE 802.11 in the following content.

The distributed coordination function (DCF) of IEEE 802.11 is essentially a carrier sensing multiple access with collision avoidance (CSMA/CA) scheme. In addition to physical sensing, it also employs a technique called virtual carrier-sensing. Virtual sensing is realized by a pair of control frames request-to-send (RTS) and clear-to-send (CTS).

Figure 4 illustrates the mechanism of RTS/CTS. Node $N1$ sends a RTS frame to node $N2$ before the real data transmission. Node $N0$ also receives the RTS and is blocked by it. Upon receiving the RTS, node $N2$ broadcasts the CTS frame to its neighbors. Thus, node $N3$ is also blocked. Node $N1$ starts transmitting data once receiving the CTS frame from node $N2$.

The RTS/CTS mechanism is to deal with hidden terminal problems. In Figure 4, node $N5$ is a hidden node of the transmission from $N1$ to $N2$, since $N5$ is beyond the

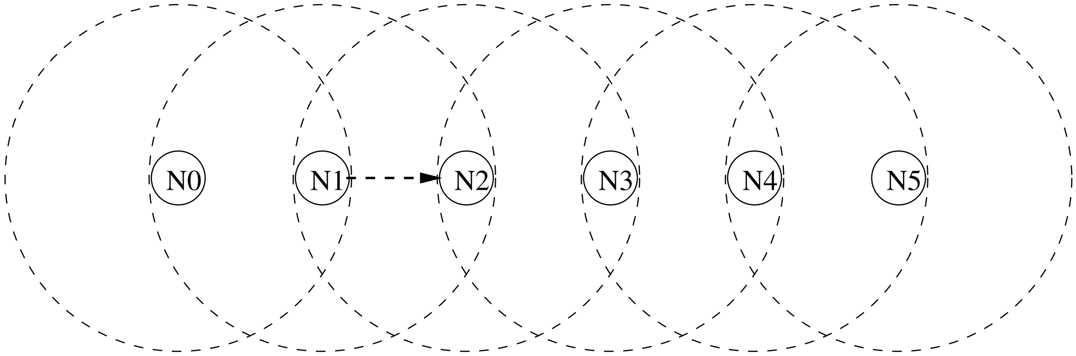

Fig. 4. IEEE 802.11 RTS/CTS

interference range of $N2$ (two hops). Node $N5$ cannot sense the data flow from $N1$ to $N2$ and will think the medium is idle. If there is no RTS/CTS, node $N5$ will directly start sending data packets to $N4$. In this case, the ACK frames from node $N4$ will be very likely to collide with the data received by $N2$. With the use of RTS/CTS, node $N5$ won't get the CTS from $N4$ and cause interference to $N1$ and $N2$ since $N4$ can detect the flow between $N1$ and $N2$.

The mechanism RTS/CTS introduces a lot of overhead especially when the data load is relatively low. Thus, sometimes, RTS/CTS is suggested to be disabled when IEEE 802.11 or its variant is used in sensor networks.

C. Directed Diffusion

Directed Diffusion [5] is a proposed routing protocol for wireless sensor networks. It manages paths in a data-centric way. A sink node publicizes its interest in some events. Such an interest will be disseminated within the network and received by nodes that have data on those events. During this process, gradients that record two-way information along all possible paths are established. Afterward, a mechanism called path reinforcement is used to select high-quality routes.

Directed Diffusion also provides two sets of application programming interface

(API). One set is network routing API, which enables upper-layer entities in sources and sinks to communicate through the network. This API takes the form of publish/subscribe paradigm. A sink node can subscribe to interesting events. Also, sources will publish their sensed events. The Directed Diffusion platform takes care of the underlying implementation.

To allow processing in intermediate nodes, Directed Diffusion also offers another API called filters. Application-specific operations can be implemented as filters to manipulate data packets as they pass through the network. Filters are provided through an interface to specify interest in particular events. For each filter, a priority is associated.

CHAPTER III

STUDY OF CONGESTION IN SENSOR NETWORKS

A wireless sensor network will suffer congestion if the load exceeds its capability. One example is many sensor nodes send or forward event reports to the sink simultaneously. Some management tasks may also introduce congestion, such as route updates by flooding. In this chapter, various aspects affected by congestion will be studied.

A. Multiple Queues

In wired networks, congestion usually occurs in a single bottleneck. Along a path from the source to the destination, packets could only be buffered in the bottleneck. However, wireless sensor networks exhibit a very different phenomenon: a number of adjacent nodes could have queues occupied at the same time.

In multi-hop wireless networks, nodes share the common media air. A node affects others within a neighborhood because of the nature of radio communication. Contention-based medium access control (MAC) protocols use carrier sensing to determine if the medium is idle before transmission. When a node is sending data, other nodes in its transmission (one hop) or interference (usually two hops) area will detect the busy medium and hold their transmissions. Sharing of the medium and the hidden terminal problem may cause multiple queues when congestion occurs.

This can be illustrated by a simple example shown in Figures 5 and 6. Assume an intermediate node $N1$ is forwarding packets and the workload is reaching its capacity limit. At this time, its neighbor $N2$ within $N1$'s interference range has some data to send and starts to contend for the channel. As a result, the available bandwidth (or time slot) for $N1$ becomes less. However, $N1$'s input traffic doesn't diminish immediately and packets will be queued in this node. The same thing will happen to

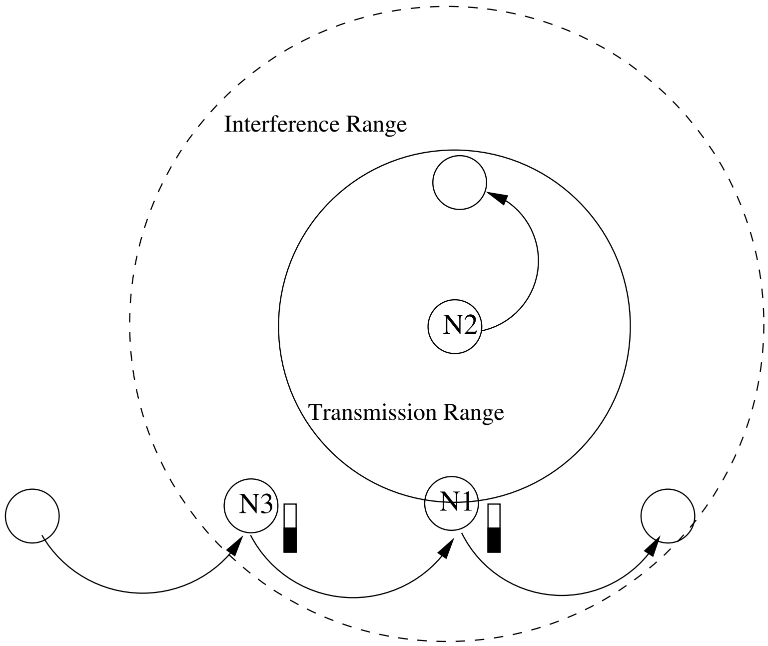

Fig. 5. Case-a of Multiple Queues

$N1$'s adjacent upstream node denoted by $N3$. $N3$ could be either within or outside the interference range of $N2$. The first case is shown in Figure 5. Node $N2$ which is transmitting can directly make $N3$ to hold, and the percentage of available time for $N3$ is reduced. Hence, the queue of $N3$ will start building up. The other case is shown in Figure 6. While $N3$ can not hear $N2$'s transmission and can't be directly influenced by it, $N2$ could be a hidden terminal to $N1$ when $N3$ sends data to $N1$ and will generate collisions. Besides, $N3$ might not receive the ACK in time from $N1$, when $N1$ senses the transmission from $N2$ and has to hold. The functionality of link layer ARQ in $N3$ will employ retransmissions and make its queue length non-empty. In both cases, nodes $N1$ and $N3$ will have non-empty buffers due to congestion.

Although the mechanism RTS/CTS in IEEE 802.11 can avoid the hidden terminal problem, some researchers [10] showed that this mechanism can induce dead lock in a community of nodes. There will be multiple queues in time of dead lock.

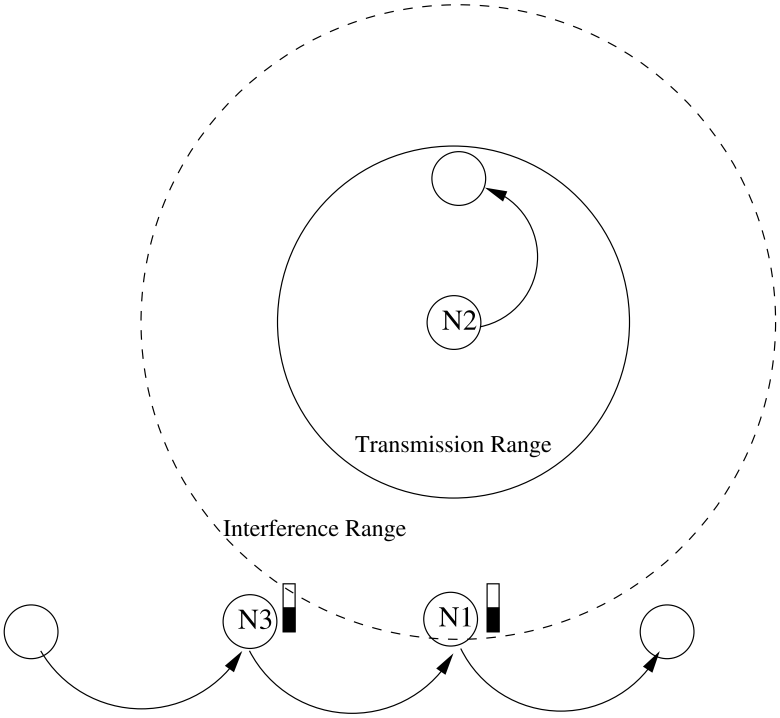

Fig. 6. Case-b of Multiple Queues

As mentioned in [9], when the traffic load keeps growing, congestion will propagate backward and more nodes will have queues building up.

Simulations are conducted in ns2 [14] to verify this characteristic. The cross topology is given in Figure 7, where the flow from node 0 to 17 competes with the other flow from node 18 to 31. Directed diffusion [5] is adopted as the routing protocol. The MAC layer is IEEE 802.11 with RTS/CTS turned off. When both rates are equal to 100 packets per second, there are 3 non-empty queues in each flow, as shown in Figure 8.

In addition, the load of each flow in Figure 7 is varied and the dynamics of queues in the first flow is observed. The time-averaged mean lengths of buffers in node 3 to 8 are given in Table II. A node's queue is considered to be non-empty if its averaged length is equal to or greater than 0.5. The number of non-empty queues increases

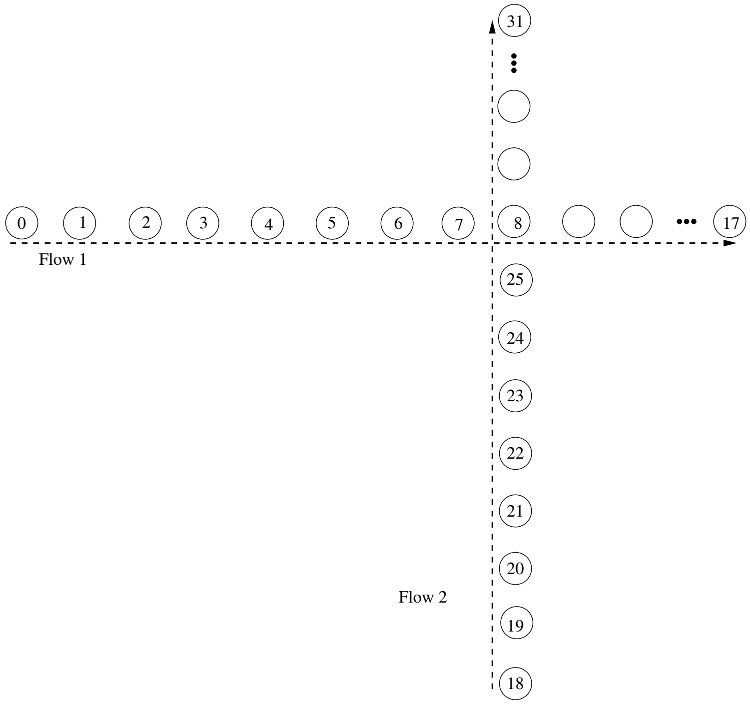

Fig. 7. Topology for Simulation of Multiple Queues

from 0 to 6 as the load increases from 80 to 150 packets per second.

The same trends can also be observed in simulating the topology with the option RTS/CTS switched on. The corresponding simulation results are showed in Table III.

B. Study of Congestion through Simulation

In this section, latency, packet loss, queue length and the number of occupied queues in time of congestion are studied through a simulation. The topology is the same as Figure 7. There are also two flows. Flow 2 keeps a constant rate 80 packets per second. Flow 1 increases its rate every 20 seconds.

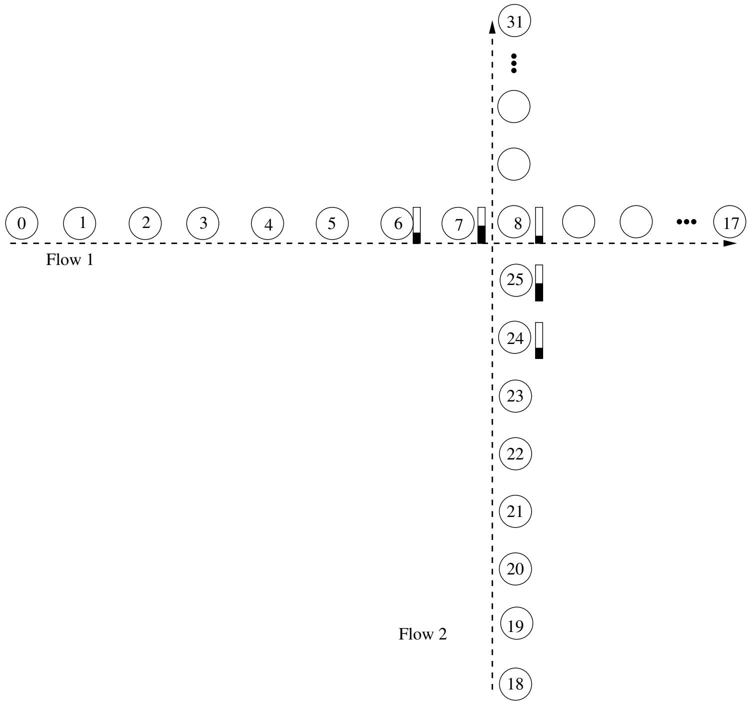

Fig. 8. Demonstration of Multiple Queues When Sending Rate=100 pkt/s

Table II. Time-averaged Queue Lengths under Different Traffic Loads with RTS/CTS Turned off

Load	Node-8	Node-7	Node-6	Node-5	Node-4	Node-3
(pkt/s)	μ	μ	μ	μ	μ	μ
80	0.06	0.13	0.12	0.00	0.00	0.00
90	0.61	2.20	1.50	0.02	0.01	0.00
100	1.82	27.06	5.27	0.08	0.01	0.01
140	1.07	39.30	14.30	0.77	0.41	0.35
150	0.88	40.73	16.66	0.89	0.50	0.94

Table III. Time-averaged Queue Lengths under Different Traffic Loads with RTS/CTS Turned on

Load	Node-8	Node-7	Node-6	Node-5	Node-4	Node-3
(pkt/s)	μ	μ	μ	μ	μ	μ
50	0.02	0.01	0.04	0.00	0.00	0.00
70	1.41	33.69	4.76	0.20	0.04	0.03
90	1.07	39.52	16.53	1.11	0.23	0.32
110	1.13	40.29	20.18	5.01	4.20	17.56

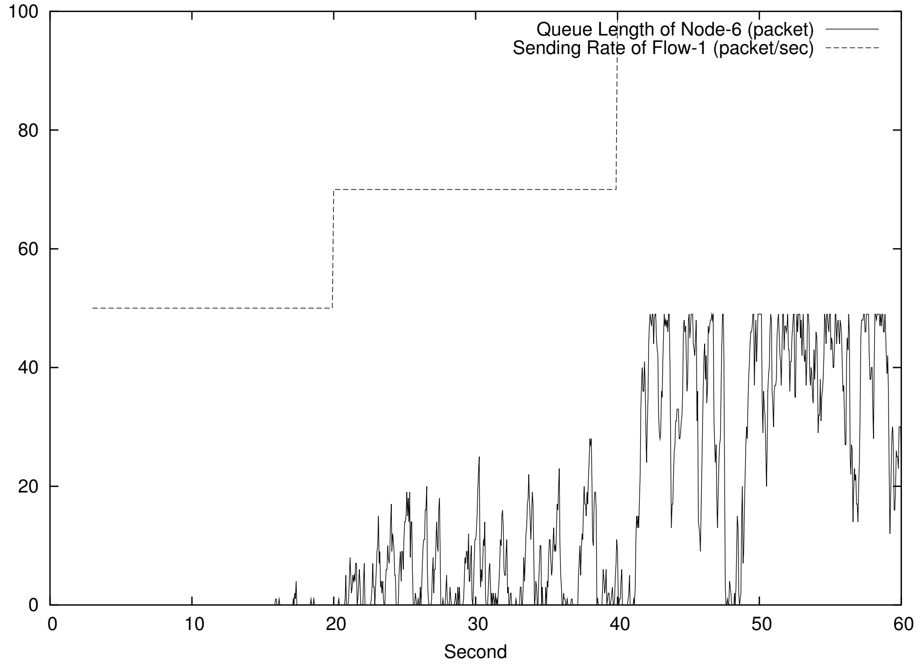

Fig. 9. Dynamics of Queue Lengths

1. Queue Length

The simulated dynamics of one queue is shown in Figure 9. During the second period from t=20 to t=40, the loads of flow 1 and 2 are not changed, while the queue of node-6 oscillates quite frequently. This is very different from the dynamics in wired networks, where the queue won't swing so much if the load is constant. Furthermore, it is observed that sometime the queue in the third 20-second stage even gets shorter than peaks in the second stage. This is another difference between WSN and wired networks. In wired networks, as the load increases, the queue length also rises. The queues of other nodes have very similar trends along the same flow.

The above phenomena can be explained by the random nature of CSMA protocols. CSMA schemes allow a community of nodes share the medium and have access to it in a random fashion. Thus, the available bandwidth for a node is not constant. In the above simulation, the queue swings in each stage since its accessible resource changes. It also explains the second difference.

2. Transmission Delay

Figure 10 shows the traces of transmission delay for the first flow. During the first 20-second period, there is no congestion and the delay has very small variations. After t=20 second, the transmission delay changes rapidly. Such oscillations are mainly due to the queuing delay and are consistent with trends shown in Figure 9. During the last 20-second period, the latency has bigger swings. It is also noticed that the latency of the first flow in the third period is not always longer than in the second period.

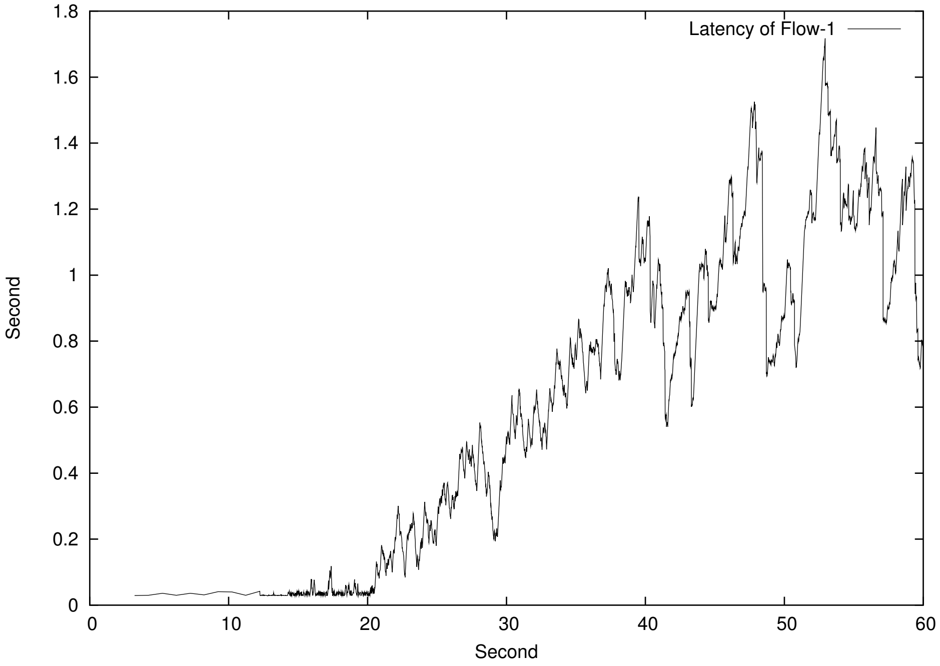

Fig. 10. Traces of Transmission Delay

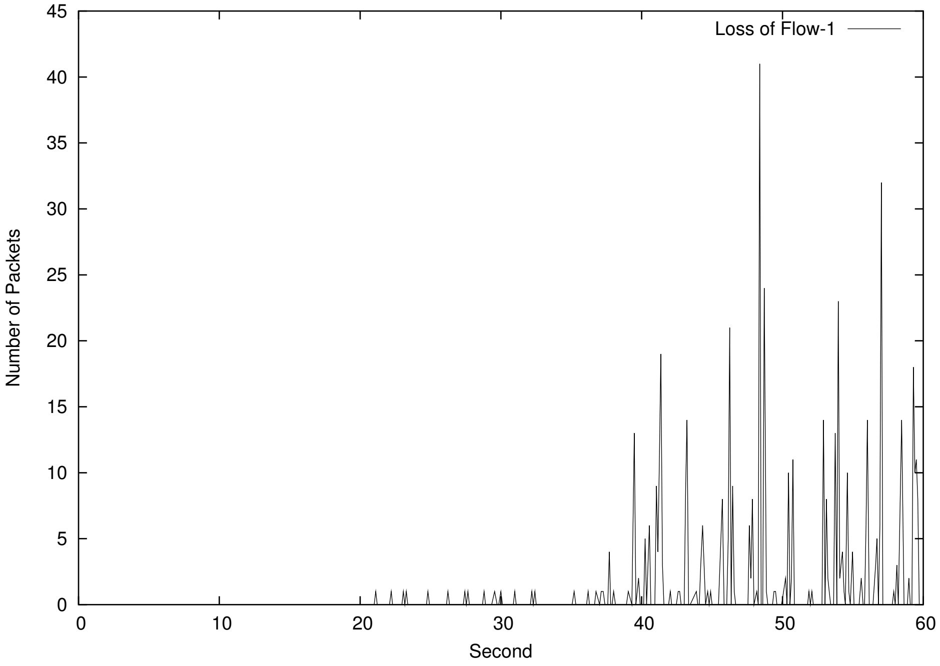

Fig. 11. Traces of Packets Loss

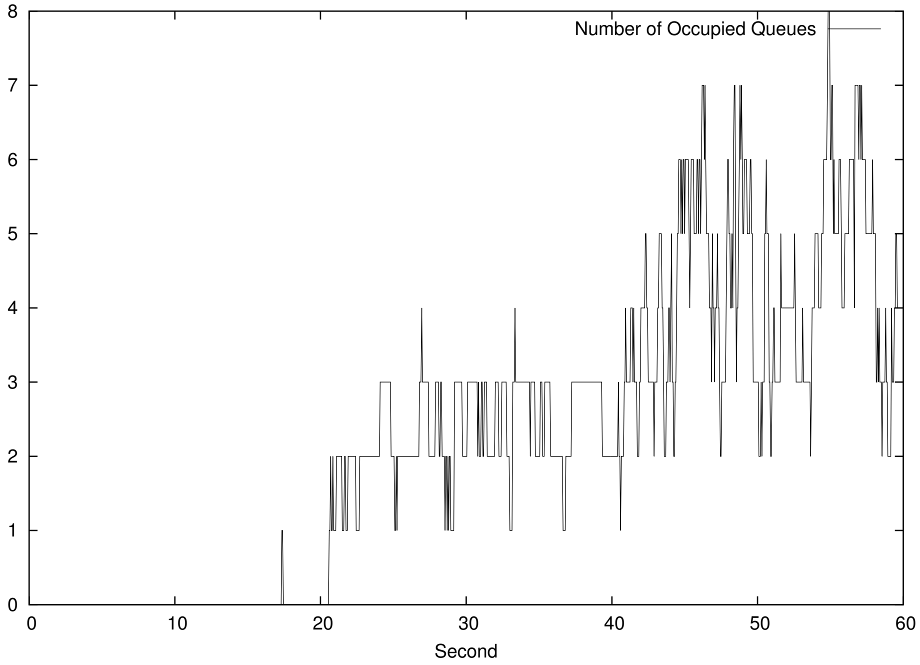

Fig. 12. The Number of Occupied Queues

3. Packet Loss

Figure 11 demonstrates the trace results of packet loss for the first flow. Before t=40, there is no buffer overflow and all packets loss is due to corruption or collision. During the last stage, patterns of consecutive losses are observed. Such patterns are mainly due to buffer overflow. Although no consecutive packet loss is observed during the second 20-second period, congestion is occurring.

4. Number of Occupied Queues

Figure 12 shows the number of occupied queues along the first flow. In the first 20 seconds, there is no congestion in the network and the number is 0 in most of the time. During the second 20-second period, this number is always greater than 0 which indicates congestion. Furthermore, this number has very small swings and is equal to 2 on the average. The last 20-second has larger load and the number of occupied queues also gets bigger as expected. It can be concluded that the number of occupied queues reflects the network situation.

C. Possible Approaches of Congestion Detection

In traditional networks, queue length, packet loss or latency can be used to detect congestion. In this section, they are examined respectively in sensor networks.

As discussed previously, a non-empty queue can tell that there is congestion. But, its trend of increasing or dropping cannot reflect the actual congestion situation. This assessment is also mentioned in [3].

Latency is not a good indicator of congestion either. It changes not only according to the load, but also according to the random allocation of the resource. Simulation results show that very low transmission delay doesn't necessarily mean no congestion in certain areas.

Without the help of other information, packet loss cannot be used to detect congestion. In WSN, corruption and collision will cause packets to be dropped. Besides, node failure due to energy depletion could also result in packet loss.

Combining information on packet loss and latency can provide a reliable approach to congestion detection. Before consecutive packet losses, the latency gets much longer than in normal conditions. This can be seen in Figures 10 and 11. But, this approach requires that the sink node dominates congestion detection and it will induce other performance problems, which will be addressed in later chapters.

The number of non-empty queues can indicate congestion level accurately. When there is a congestion, this number is larger than 0. As shown in Figure 12, this number increases as network load increases. Furthermore, its swing is much smaller than other criteria, such as queue length and delay. One more advantage of detecting congestion using this number is that it doesn't need to involve the sink node. This will be discussed in the next chapter.

D. Related Work on Congestion Control

There are several papers published on congestion control or transport protocols for wireless sensor networks.

1. CODA

In [3], a congestion detection and avoidance scheme called CODA (COngestion Detection and Avoidance) is presented. CODA employs two mechanisms to detect congestion cooperatively. First, intermediate nodes need to measure workload and infer congestion by comparing it with a maximum throughput threshold applicable to CSMA, which is

$$S_{max} \approx \frac{1}{1+2\sqrt{\beta}} \tag{3.1}$$

where,

$$\beta = \frac{\tau \cdot C}{L} \tag{3.2}$$

Parameter τ is delay, C denotes the raw channel bit rate and L is the expected number of bits in a packet.

Once an intermediate node detects local congestion, it will take a measure called "Open-Loop Hop-by-Hop Back-pressure" to regulate the congestion. Simulation results show that this back-pressure cannot deal with persistent congestion in large networks.

The second mechanism in CODA is making the sink regulate the sources in a closed-loop manner. The sink has to provide frequent ACKs back to source(s) when congestion happens.

2. ESRT

A transport protocol called ESRT (Event-to-Sink Reliable Transport) is proposed in [4]. By studying the relationship between reporting rate and reliability, the authors figure out a optimal operating region. The sink node regulates the sources' reporting frequency to make the system operate in the optimal region.

In ESRT, sinks dominate congestion control. They infer if there is congestion. ESRT also uses closed-loop signaling to control the sending rates of sources.

In times of congestion, closed-loop feedback has much longer latency before sources can really reduce reporting frequency and mitigate the congestion. This could lead to more losses (retransmission if link layer ARQ is used) and waste of energy.

PSFQ (Pump Slowly, Fetch Quickly) and RMST (Reliable Multi-Segment Transport) proposed respectively in [6] and [7] mainly discuss the reliability issue of transport protocols in lossy sensor networks.

CHAPTER IV

PROPOSED CONGESTION CONTROL SCHEME

A. Scheme Design

In this section, a congestion control scheme called MFACCS (Midway Feedback and Adaptive Congestion Control Scheme) is proposed. It is comprised of three parts: congestion detection by tracking the number of non-empty queues; on-demand midway feedback and adaptive rate control.

1. Congestion Detection

In the previous simulations, it is observed that the number of nodes with occupied queues grows if congestion gets worse. The proposed scheme takes advantage of this fact and uses the number of non-empty queues along a flow's path as a reasonable indication of congestion level.

A field called queue counter (QC) as shown in Figure 13 is reserved in each data packet to track the number of consecutive occupied queues along a path. Before a packet is sent out, its QC is initialized to zero. As the packet traverses the network from the source to the sink, the QC field will be processed by intermediate nodes according to their local queue lengths. An intermediate nodes monitors its local queue and compares the time-averaged length *Qavg* with a threshold *Qth*. If *Qavg* is equal to or larger than *Qth*, the node will increment the QC field by 1; If *Qavg* is

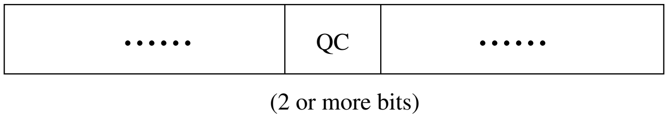

Fig. 13. Illustration of QC Field

less than Qth, this means the local queue is empty. In this case, the QC field of the packet is checked. If its value is zero, it means that the immediate upstream node also has empty queue and there is no congestion; If the QC's value is greater than 0, it suggests congestion in upstream and the QC's value can be an indicator of the congestion level.

In this mechanism, the QC field needs to occupy two or more bits, which depends on the application.

2. Feedback

When congestion is detected, the sources should be notified. There are several approaches to do this. A widely used approach is closed-loop feedback, in which, notification is sent to the source by the sink node.

To save energy for low-power sensor nodes, a light-weight and effective mechanism, midway feedback as shown in Figure 14 is proposed. It has two characteristics: midway and on-demand. In this mechanism, an intermediate node directly sends notification back to the source via its upstream nodes. The data packet will still move downstream to the sink. But, the sink doesn't get involved in the feedback process. This approach shortens the feedback latency and saves energy dissipation.

If no congestion is detected, nothing will be sent back to the source. Thus, notification packets appear only when there is congestion. This is the notion of on-demand. The sender infers no congestion if no notification packet is received.

Further, lack of energy imposes constraints on the frequency of feedback in WSN. Frequent feedback used in TCP (ACK piggybacks implicit congestion notification) and other transport protocols like TFRC[13] will bring a large amount of signaling burden. More power-efficient ways are needed. In this proposed scheme, intermediate nodes perform midway feedback once in a short period, denoted as T_{cn}, when

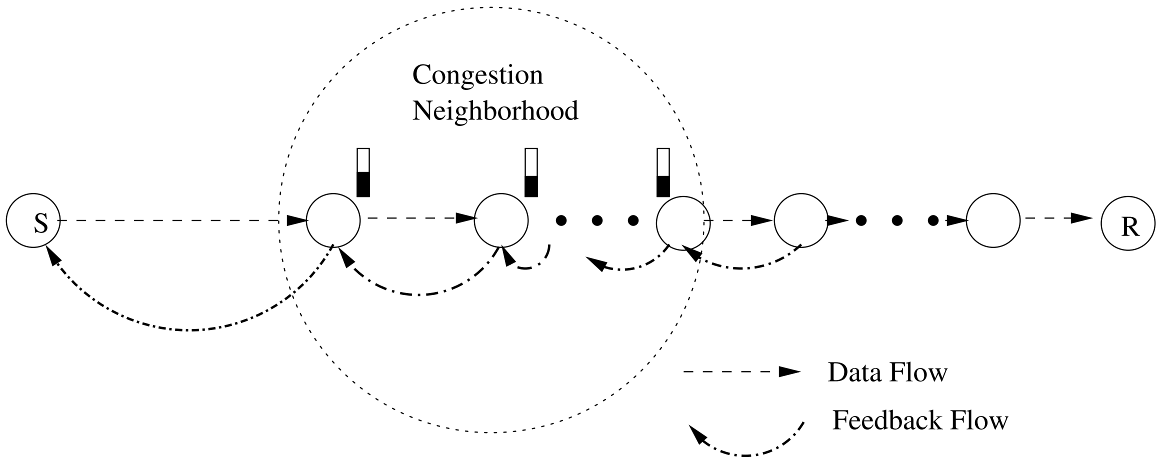

Fig. 14. Illustration of Midway Feedback

congestion is detected.

On the reverse path, feedback is transmitted in unicast. Intermediate nodes need to record the addresses of upstream nodes when data packets are received. Such that, the feedback can be sent to the anticipated upstream nodes and finally to the correct sources.

From the view point of information carried by feedback, this mechanism is a kind of M-ary feedback. The QC field records the number of occupied queues. Once it is fed back, the information it contains can help the source take proper measures according to the congestion level.

3. Adaptive Rate Control

In this scheme, sources are chosen to control their sending rates according to network status. Such a distributed mechanism is scalable to the network size. Each source is responsible for handling congestion occurring on its path. When several sources share a common congested channel, all of them will receive notification and control their rates respectively.

As a task requirement, the sink sends sources an upper-bound on reporting frequency, denoted by R_{max}. The rate control algorithm is shown in Equation 4.1:

$$R_{n+1} = \begin{cases} min\,(R_n + \alpha\Delta, R_{max}) & \text{no feedback received;} \\ R_n \cdot e^{-\beta\Delta} & \text{congestion feedback received;} \end{cases} \tag{4.1}$$

where, R_n means the sending rate at the nth decision making point, α is the rate increasing step size, β is the exponentially decreasing parameter, and Δ is the interval of rate change. Considering an intermediate node sends feedback once in a period T_{cn}, Δ is set to T_{cn}.

β is not constant during the entire controlling process. It varies as the value of QC. A number denoted by M reflects the congestion level. In this scheme, a very simple form is applied for β.

$$\beta = \beta_0 + (M - 1) \cdot p \tag{4.2}$$

In equation 4.2, β_0 is a constant, and p acts as a penalty coefficient. Adjustable β embodies the notion of adaptiveness to congestion level.

B. Discussion of the Scheme

1. Objective of Rate Control

In sensor networks, energy dissipation is the primary concern. That is why congestion avoidance is given higher priority than utilization of bandwidth in this scheme. As seen from Equation 4.1 and 4.2, the rate decreases adaptively to the congestion level, while the rate increasing step size α won't adjust even when the network capacity is not fully utilized.

2. Data Suppression

When a neighborhood of source nodes are placed within the sensing region of an event, multiple highly correlated data tuples will be generated. In energy-restricted sensor networks, these redundant data should be suppressed. In this scheme, an ideal case is assumed that only one copy of data tuple representing the occurred event will be reported to the subscribing sink node, while the others will be locally suppressed.

This kind of suppression is application-specific and should be done in application layer protocols. Data suppression should not be applied to different events.

3. Many-to-one Traffic Pattern

In a typical sensor network, one sink node collects information from a large number of sensor nodes. Many-to-one traffic pattern dominates.

One situation of many-to-one is that a neighborhood of nodes sense a common event and all of them try to report it to the sink. This type of many-to-one can be transformed to one-to-one by the above data suppression.

Another situation is when multiple independent events occur and need to be reported at the same time. To support congestion control on each event reporting, some measure has to be taken to discriminate them. A locally allocated ID named as event-ID is associated with each event. In implementation, the combination of some physical address of the reporting node, event occurring time and a random number can be used to achieve a locally unique identifier.

Since it is possible that multiple events pass through one common node, this node has to record last-hop information for each flow to correctly deliver congestion notification. In addition, taking into account the fact that mobility and node failure could cause routing change, these last-hop addresses must be updated very often.

4. Multiple Congested Spots

When a flow passes through multiple congestion areas, as described in previous design, the source node will receive a series of congestion notifications. Each of them denotes the congestion level of a certain congested area. Considering that the metric of congestion level is not additive (no apparent link between two congested area), the source is designed to pick the most severe congestion level as the target of rate control.

5. Flooding-induced Congestion

The proposed scheme relies on the underlying protocols like routing. In sensor networks, some routing protocols use flooding to build the routing table. For example, in Directed Diffusion, interest of the sink node is disseminated by broadcast and finally floods over the network.

This kind of flooding-induced congestion is not the subject of this scheme. It is better to devise a cross-layer protocol to combine information from the routing and congestion control protocols. In current implementation of this scheme on top of Directed Diffusion, the function of rate control doesn't work during the routes exploratory period.

6. Applicability

The proposed scheme is based on the assumption that multiple nodes along one path could have non-empty queues in time of congestion. For CSMA-based MAC (medium access control) protocols such as IEEE 802.11 and SMAC, this assumption can be held. But, this scheme is not applicable to TDMA-based MAC protocols, since channel utilization is scheduled.

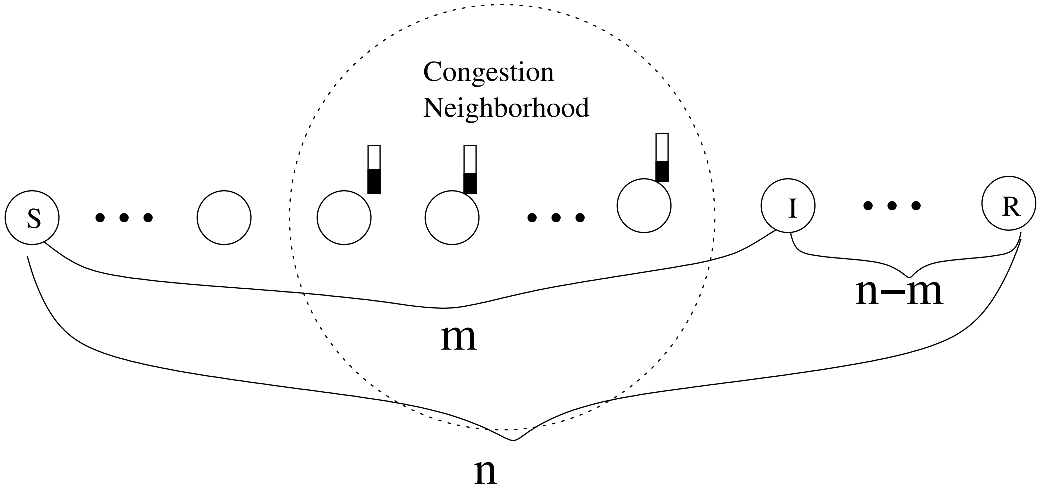

Fig. 15. A Single Connection in WSN

C. Comparison of Feedback Approaches

In this section, the performance of two feedback approaches, midway and closed-loop, is compared. A simple model shown as Figure 15 with one source and one sink is used. Node S, R, and I denote the source, sink and intermediate node respectively. The total number of hops along the path is n. The source is m hops away from the intermediate node I which sends congestion notification back in the midway feedback approach.

1. Signaling Energy Cost

It is assumed that energy dissipation in transmission of one packet per hop is a constant, regardless of the configuration of MAC and link layer. It is denoted by ϵ.

In Figure 15, it is easy to obtain the signaling energy expenses per cycle T_{cn} for the two schemes.

$$E_{midway} = m \cdot \epsilon \tag{4.3}$$

$$E_{closed-loop} = n \cdot \epsilon \tag{4.4}$$

If the number m is uniformly distributed between 0 and n, the expectation of signaling energy for midway feedback is:

$$E[E_{midway}] = \frac{n}{2} \cdot \epsilon = \frac{1}{2} E_{closed-loop} \tag{4.5}$$

For closed-loop approach, the dedicated feedback packet is sent from the sink to the source.

When there is only one congestion area along a path, the midway feedback costs less energy than closed-loop. In case of large-scale congestion, the midway feedback mechanism may send multiple notifications per cycle T_{cn} and therefore would cost more energy than closed-loop feedback. But, such situations are not common in sensor networks.

2. Propagation Delay

Let τ_h denote the propagation delay per hop. The total propagation delay from the source via the network and back to the source again, denoted by τ_{midway} and $\tau_{closed-loop}$ for the two methods are

$$\begin{aligned} \tau_{midway} &= 2m \cdot \tau_h \\ \tau_{closed-loop} &= 2n \cdot \tau_h \end{aligned} \tag{4.6}$$

Under the assumption of uniformly distributed m between 0 and n, the expectation of τ_{midway} is

$$E[\tau_{midway}] = \frac{1}{2} \cdot \tau_{closed-loop} \tag{4.7}$$

3. Aggregate Queues Length

As shown in Equation 2.7, the maximal queue length is proportional to squared propagation delay τ. From the above comparison, midway feedback's averaged delay is half of closed-loop's in average. Hence, the conclusion can be drawn that the expectation of aggregate queue lengths in closed-loop scheme during one rate control cycle is around four times as that in midway feedback approach.

Furthermore, queuing delay and the number of packet losses in midway feedback are also expected to be smaller than the closed-loop approach. Simulation results given in the next chapter verify this analysis.

CHAPTER V

SIMULATION AND RESULTS ANALYSIS

A. Implementation in NS2

The scheme has been implemented in ns2 version 2.27. On top of the platform of directed diffusion, three types of entities, source, filter and sink constitute the whole congestion control system.

1. Source Entity

The entity "source" represents a node that has events to report. It realizes the following functions:

1. Accept subscription: Before sending event reports, the source needs to know who and where the sink node is. It obtains such information by accepting matched interest subscribed by the sink.

2. Report events: After accepting subscription, the source node will monitor such interesting events and report them to the sink(s).

3. Regulate sending rate: In the course of reporting events, the sending rate will be controlled according to the condition of network. Adaptive additive increase and multiplicative decrease (AIMD) is implemented.

The source entity is implemented as an "Application" in the platform of "Directed Diffusion".

2. Sink Entity

The sink entity carries out the functions of a sink node (It is also called base station in some networks). Its main routines are listed below.

1. Subscribe events: The sink node subscribes its interested events from the network by calling the interface *subscribe()*. The interest is represented by a set of attributes.

2. Process received reports: The sink evaluates the performance of event reports in terms of latency, efficiency.

Like the source entity, it is also implemented as an "Application" of "Directed Diffusion".

3. Filter Entity

The filter entity is associated with each sensor node and implements intermediate nodes' functions related to congestion control. Its tasks include:

1. Monitor queue: the filter periodically monitors the queue length and averages it in the axis of time.

2. Detect congestion: the filter determines local congestion if the averaged queue length exceeds a threshold.

3. Record events: the filter records the event-ID and the last-hop information of any event report which passes an intermediate node.

4. Handle feedback: once congestion is detected, intermediate nodes will initiate or forward the feedback packets to the corresponding source entities.

Figure 16 shows the flow diagram of the filter entity upon receiving an interested packet.

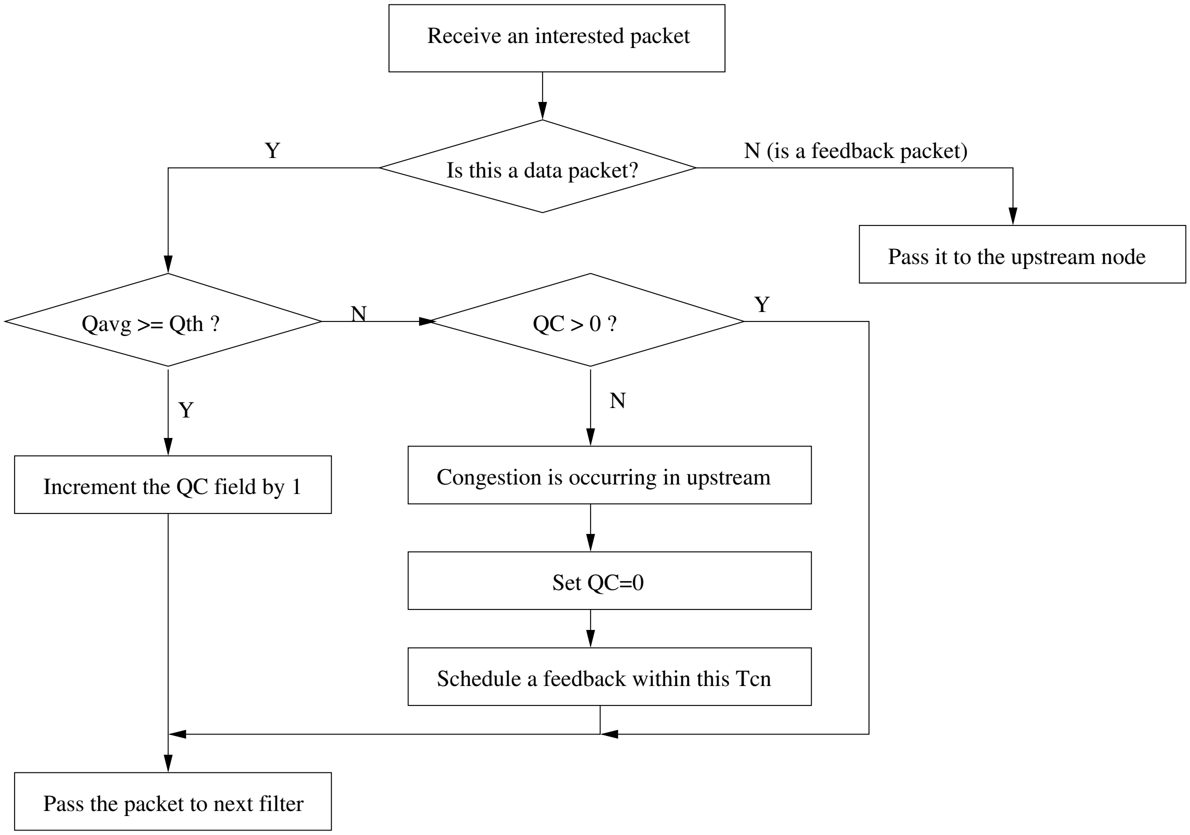

Fig. 16. Flow Chart of the Filter Entity Upon Receiving an Interested Packet

The filter entity is built upon the filter_core of Directed Diffusion. Because intermediate nodes know the reverse path, the feedback packets don't need to go through the underlying routing procedures of Directed Diffusion. In implementation, the priority of the congestion control filter is set higher than the priority of Directed Diffusion routing filter.

B. Simulation Environment

Directed Diffusion is used as the routing protocol in all simulations. The routing core works in two-phase-pull mode.

IEEE 802.11 is chosen as the MAC protocol. There is another CSMA-based protocol called SMAC in ns2. But the SMAC version in ns2.27 has some problems when used together with our scheme. The data rate of IEEE 802.11 is set to be 1 Mbps. Simulations are conducted for IEEE 802.11 both with and without RTS/CTS.

To imitate practical systems, the energy model adopts the parameters shown in Table IV.

Table IV. Energy Model

State	Power (watts)
Idle	0
Receiving	0.395
Transmitting	0.660

In all simulations, two-ray ground reflection model is employed as the propagation model. Parameters shown in Table V are tuned to make the receiving distance to be 40 meters and the sensing distance to be about 80 meters.

Six randomly generated networks are used to test the proposed scheme. These networks described in Table VI scale from 50 nodes to 300 nodes with a relatively constant density of node per unit area. Sensor nodes are static in all topologies. In each network, only one sink node (base station) is placed near the edge of the network. Figure 17 illustrates the topology of the 150-node simulation network.

To better evaluate the performance of MFACCS by comparison, two other schemes

Table V. Parameters for Propagation Model

Parameter	Value
P_t	8.5872e-4 (w)
G_t	1
G_r	1
H_t	1.5 (m)
H_r	1.5 (m)
L	1

Table VI. Information about Simulated Networks

Network	Number of Nodes	Width (m)	Height (m)
1	50	100	100
2	100	140	140
3	150	173	173
4	200	200	200
5	250	220	220
6	300	250	250

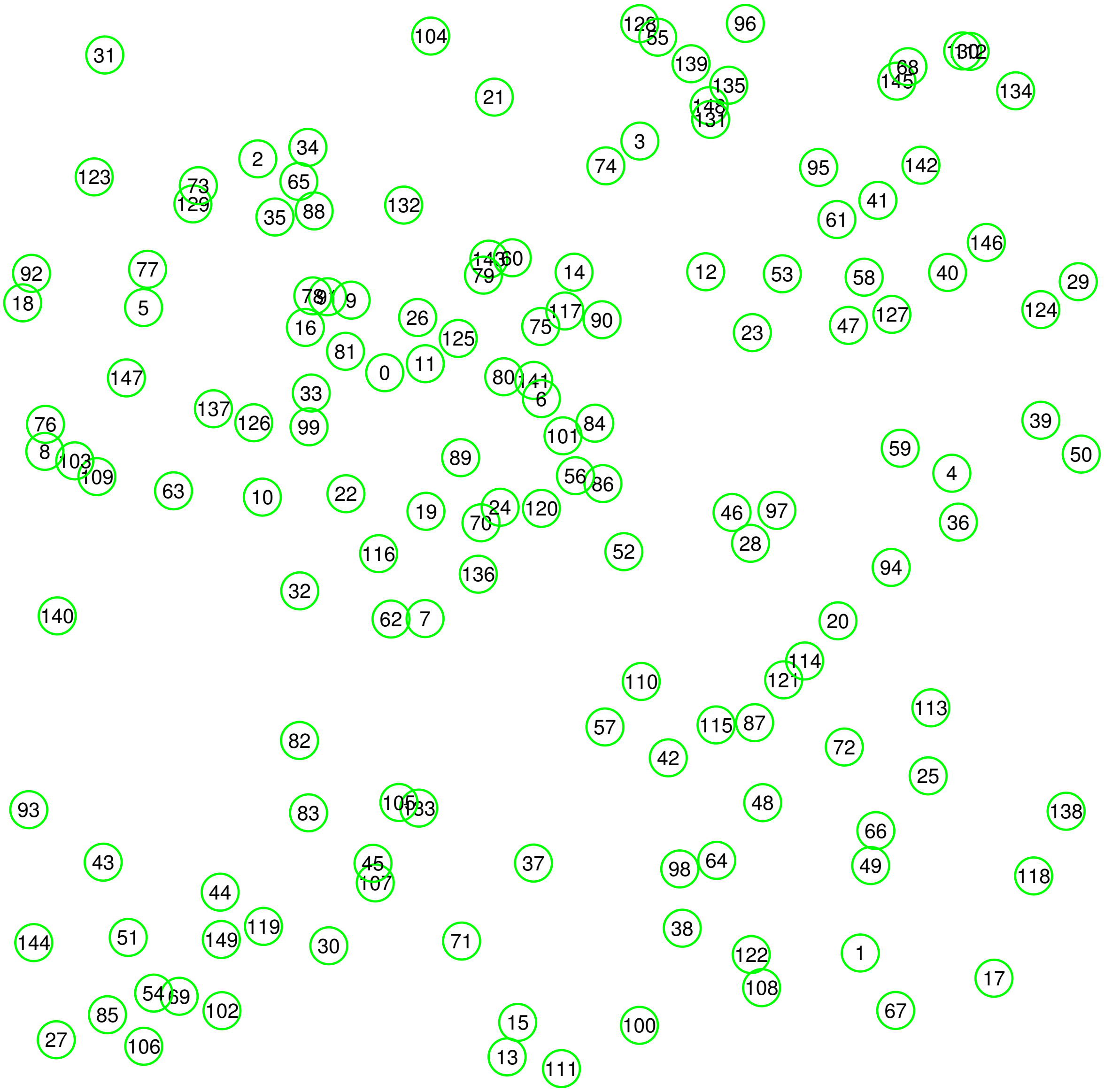

Fig. 17. 150-node Random Simulation Network

Table VII. Simulated Schemes for Performance Comparison

Scheme	Feedback	Rate-control
NCC	None	None
CLS	closed-loop	AIMD
MFACS	midway	Adaptive AIMD

are defined and simulated under the same conditions. One is called No-Congestion-Control (NCC), the other is called Closed-Loop Scheme (CLS). NCC doesn't have any congestion control and sources report events in a predefined fixed frequency R_{max}. CLS uses the closed-loop feedback and applies classical AIMD. Table VII shows the configuration of the three schemes.

In each network, a number of traffic patterns are simulated. In each pattern, several random source nodes report events to the sink node at a predefined rate of 50 packets per second. The same pattern and network will be simulated three times by applying the three schemes to compare the performance of these schemes.

Table VIII lists major parameters for the proposed scheme, which remain constant in all simulations.

Table VIII. Parameters for the Scheme MFACCS

Parameter	Description	Value
α	rate increasing parameter	13 (packets/s^2)
β_0	basic rate decreasing parameter	0.65 (s^{-1})
p	penalty coefficient for β_0	0.5 (s^{-1})
T_{cn}	interval of congestion control	0.25 (s)

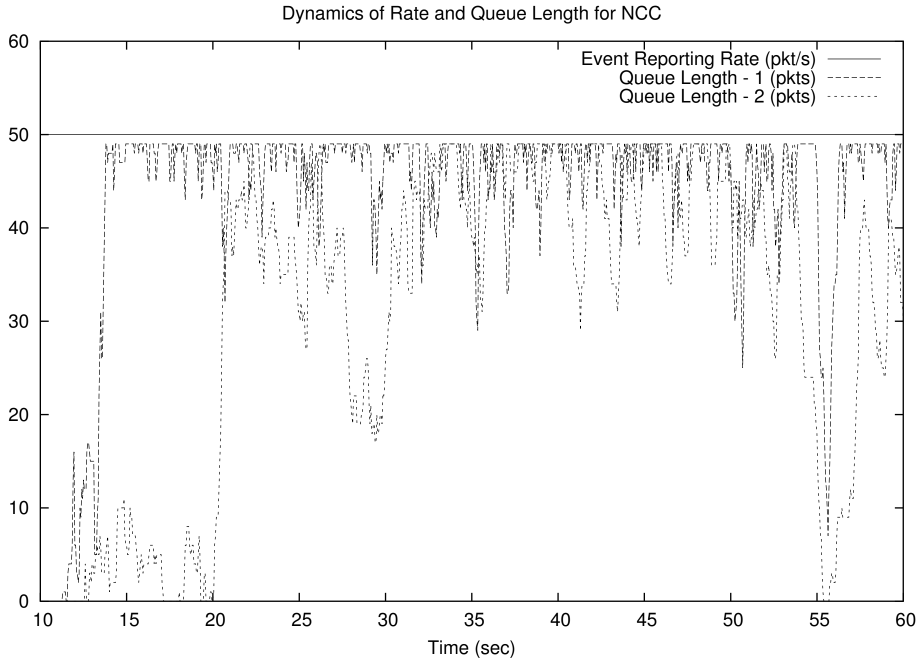

Fig. 18. Dynamics in Extreme Congestion for NCC Scheme

C. Cases Study

1. Extreme Congestion

A case of extreme congestion is discussed in this section. In this case, six sources report their own events simultaneously to a single sink in the 150-node network shown in Figure 17. Node 149 is selected to be the sink and node 22, 23, 39, 118, 140 and 142 are randomly selected as sources. RTS/CTS option is enabled in this set of simulations.

The simulation results show that the most severe congestion occurs in the flow from node 142 to the sink. Figures 18 through 20 depict the sending rate and most congested queues which appear in node 124 and 20 of this flow for each scheme.

Figure 18 shows that, for NCC scheme, both queues shoot up to the limit of 49 packets very quickly. After that, congestion and packet dropping continue until the

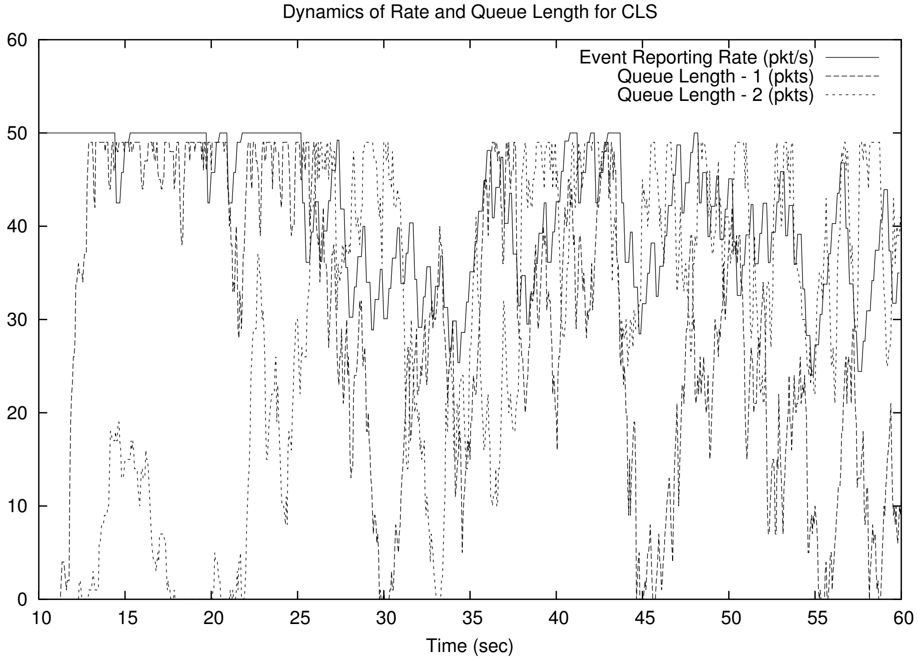

Fig. 19. Dynamics in Extreme Congestion for CLS Scheme

end of the simulation.

Under the same circumstances, the CLS scheme fails to control congestion in time because the first queue rises to the limit and dropped the critical feedback packets as shown in Figure 19. This situation continues to the point t=25 second. Then, CLS decreases the reporting rate but not enough to get rid of congestion. Due to the influence of other flows, the path falls in extreme congestion again around t=36 second.

In contrast, MFACCS, as shown in Figure 20, promptly reduces the rate in the beginning and avoids dropping of any packets. Furthermore, it regulates the reporting rates and maintains the queue lengths to be low for the rest of the simulation.

The comparison of five averaged performance metrics for the three schemes are given in Table IX. MFACCS beats NCC in all aspects except the good-put (received reports). Compared with CLS, MFACCS achieves 81 percent latency shortening,

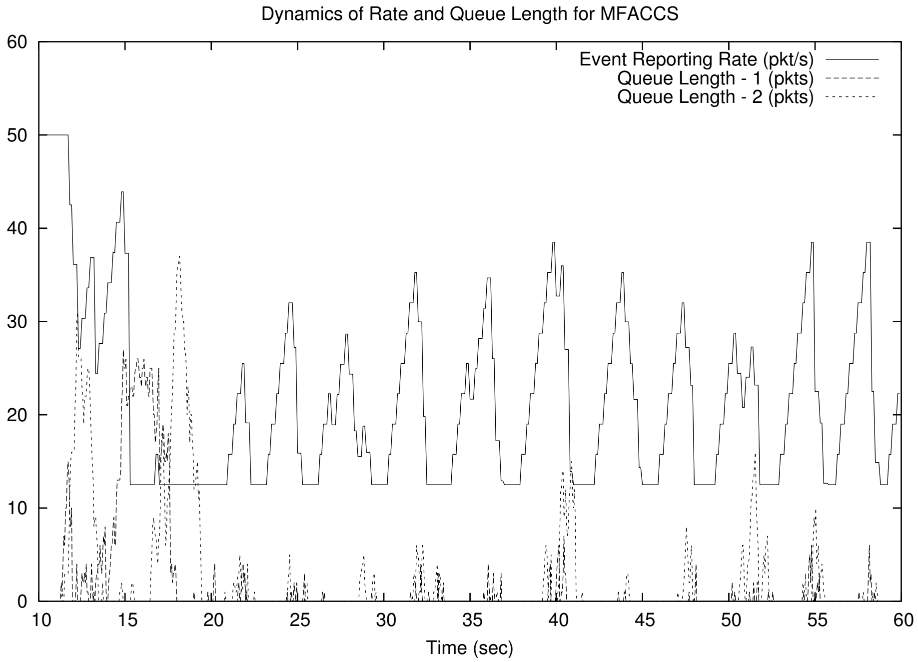

Fig. 20. Dynamics in Extreme Congestion for MFACCS Scheme

reduces 84 percent of dropped packets. MFACCS also saves one tenth energy while providing 20 percent more reports than CLS. It is consistent with the difference of transmission efficiency between them.

2. Moderate Congestion

In this part, a case of moderate congestion is presented. This case also occurs in the 150-node network. There are four sources (node 7, 40, 98 and 130) reporting to the sink node 149. RTS/CTS option is enabled in this case.

Figures 21, 22 and 23 depict the dynamics of one single congested flow from node 130 to the sink for each scheme.

As shown in Figure 21, since NCC has no congestion control, a queue stays close to the limit 49 packets in most of the time.

For CLS scheme, queue lengths start increasing in the beginning, as shown in

Table IX. Performance Comparison in Case of Extreme Congestion

	NCC	CLS	MFACCS
Latency (second)	1.533	1.322	0.251
Loss (packets)	5416	2373	375
Efficiency	0.603	0.701	0.947
Energy (Joules)	1750.7	1543.8	1409.3
Received Reports	8197	5561	6671

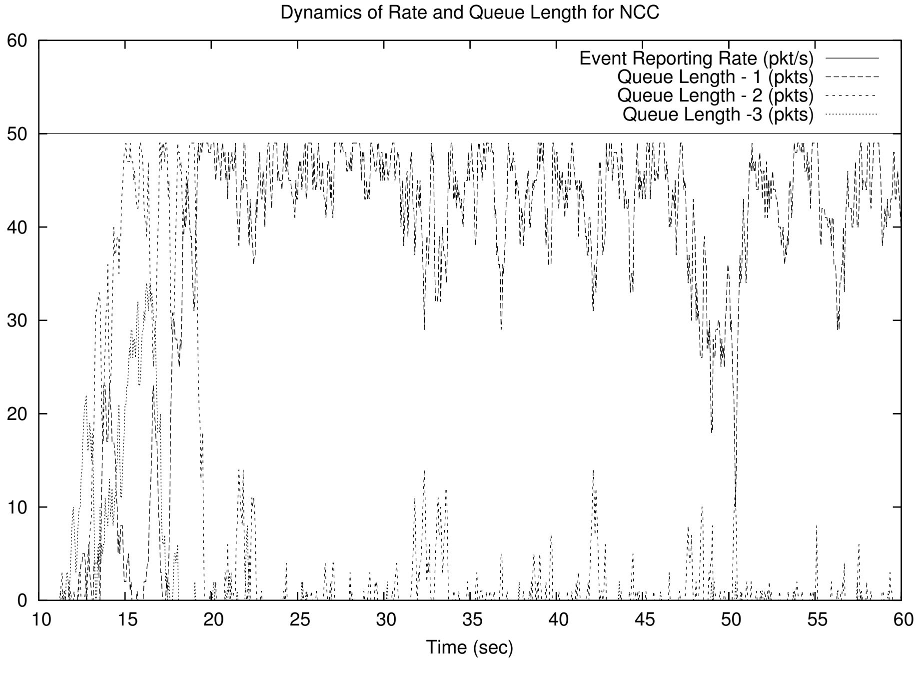

Fig. 21. Dynamics in Moderate Congestion for NCC Scheme

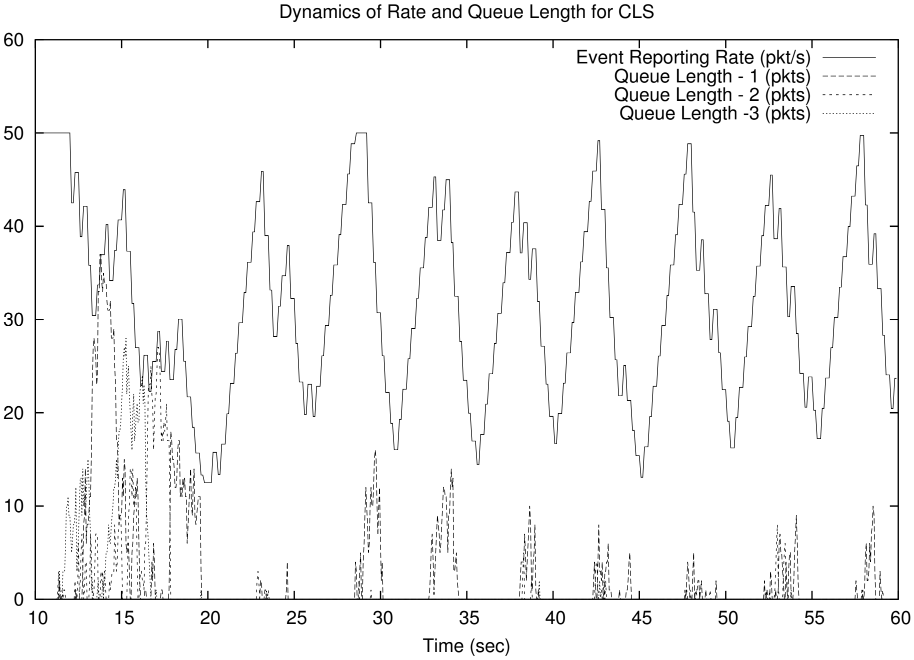

Fig. 22. Dynamics in Moderate Congestion for CLS Scheme

Figure 22. But, there is no packet dropped before the feedback packets sent from the sink arrive at the source successfully. Subsequently, the dropping of the sending rate releases all queued packets at the time t=20 second. After that, a low level of congestion occurs periodically.

MFACCS detects the congestion and starts reducing the rate earlier than CLS, as shown in Figure 23. It is also noticed that the decreasing slope of Figure 23 is steeper than that in Figure 22. This is due to MFACCS' adaptive AIMD. Because of its faster response and adaptive rate control, MFACCS makes queue lengths lower and the duration of congestion shorter than CLS.

Performance evaluation of this case is shown in Table X. Compared with NCC, MFACCS obtains much better performance in latency, loss, efficiency and energy dissipation although losing 12 percent of delivered reports. CLS spends more energy than MFACCS and has 2.3 times latency, 3.5 times loss and 7 percent less successfully

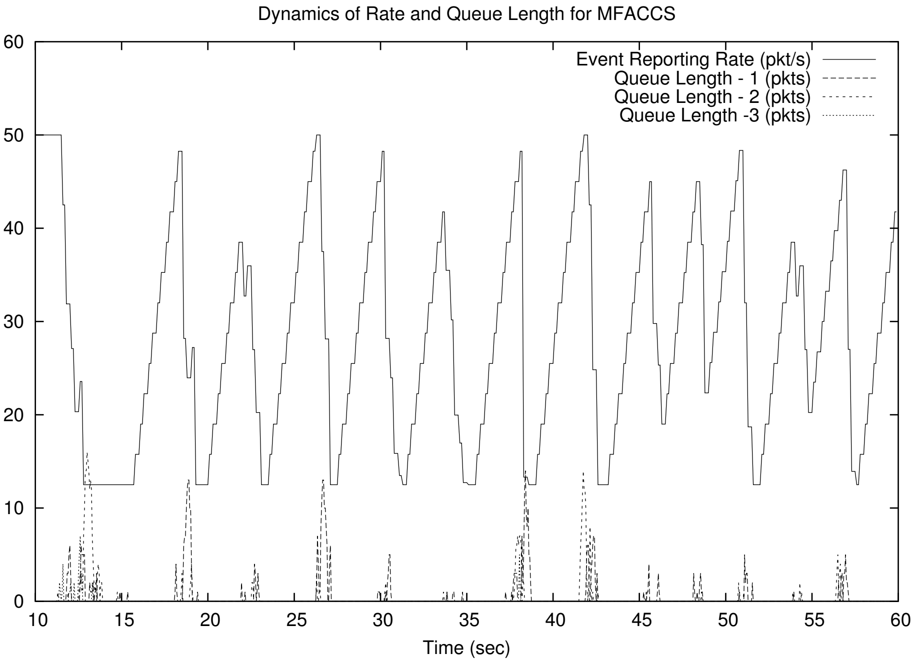

Fig. 23. Dynamics in Moderate Congestion for MFACCS Scheme

delivered reports.

Simulations without using RTS/CTS in 802.11 also exhibit very similar dynamics in the above two congestion scenarios.

3. No Congestion

As designed, the proposed scheme MFACCS won't induce any communication overhead in non-congested cases, except for some computation of queues length. To test its functionality, simulations without congestion are conducted. Results show that MFACCS performs the same as NCC with no congestion control in the above mentioned five aspects. Table XI shows the result of a 3-flow simulation (with disabled RTS/CTS) in the 50-node network.

Table X. Performance Comparison in the Case of Moderate Congestion

	NCC	CLS	MFACCS
Latency (second)	0.610	0.167	0.072
Loss (packets)	1739	272	78
Efficiency	0.819	0.960	0.989
Energy (Joules)	1553.7	1277.8	1208.6
Received Reports	7874	6505	6937

Table XI. Performance Measures in Case of No Congestion

	NCC	CLS	MFACCS
Latency (second)	0.007	0.007	0.007
Loss (packets)	0	0	0
Efficiency	1.0	1.0	1.0
Energy (Joules)	484	484	484
Received Reports	11730	11730	11730

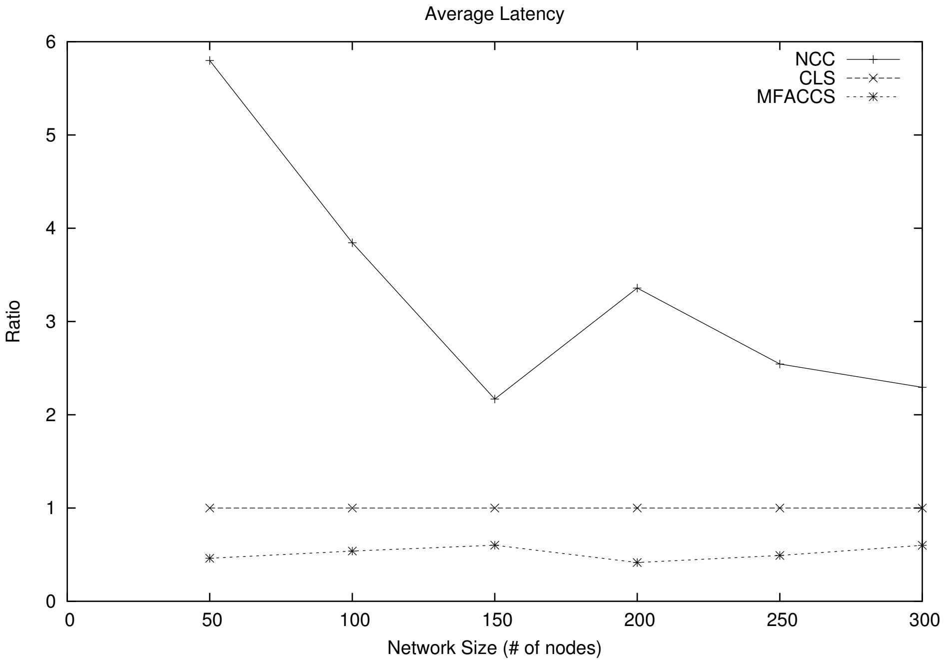

Fig. 24. Average Latency in Various Networks

D. Averaged Results

Besides the above cases, a large number of simulations are conducted to evaluate the MFACCS scheme more accurately. For each of those six networks, three to six sources are randomly selected to report events. The same traffic pattern applies to three schemes NCC, CLS and MFACCS. After filtering out non-congested cases from all experiments, averaged performance metrics are computed for each network. Figures 24, 25, 26, 27 and 28 show the detailed results in terms of latency, loss, energy dissipation, successfully delivered reports and transmission efficiency for using RTS/CTS in 802.11.

The following facts are observed from those plots:

- Latency: MFACCS only has almost half latency of CLS. Latency of NCC can increase up to 12 times as MFACCS.

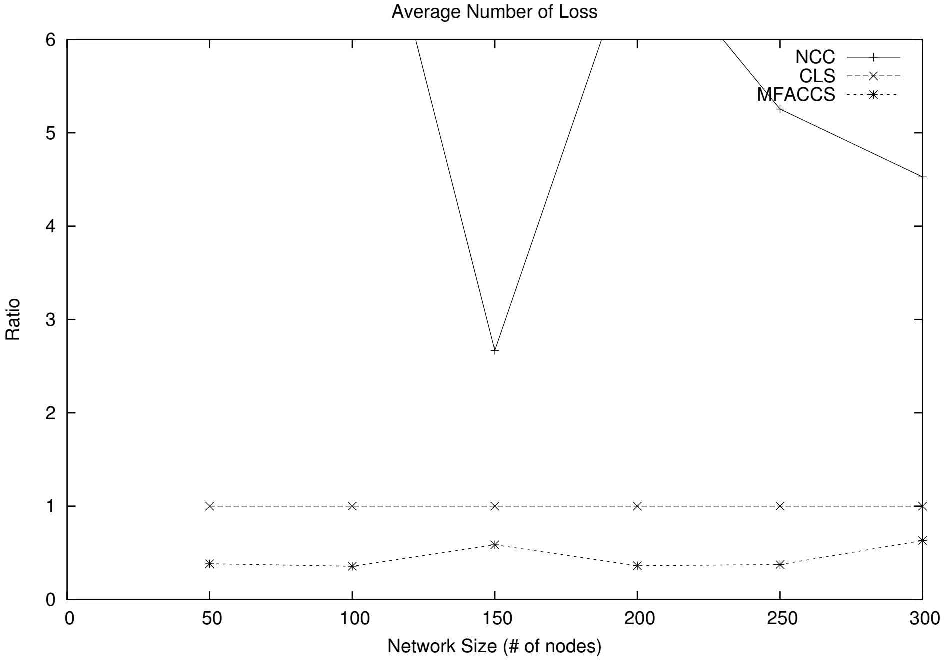

Fig. 25. Average Number of Loss in Various Networks

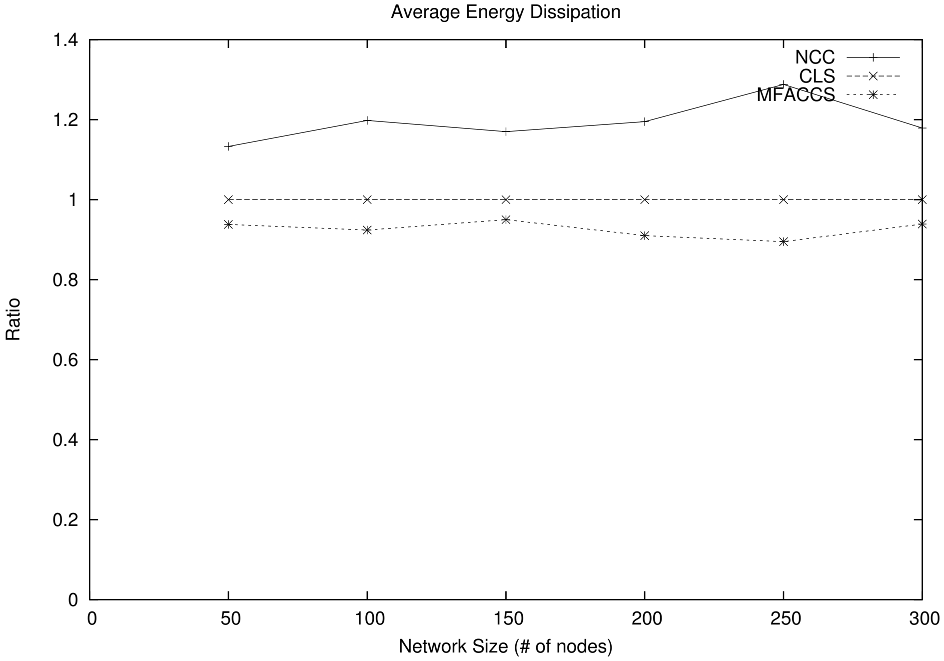

Fig. 26. Average Energy Dissipation in Various Networks

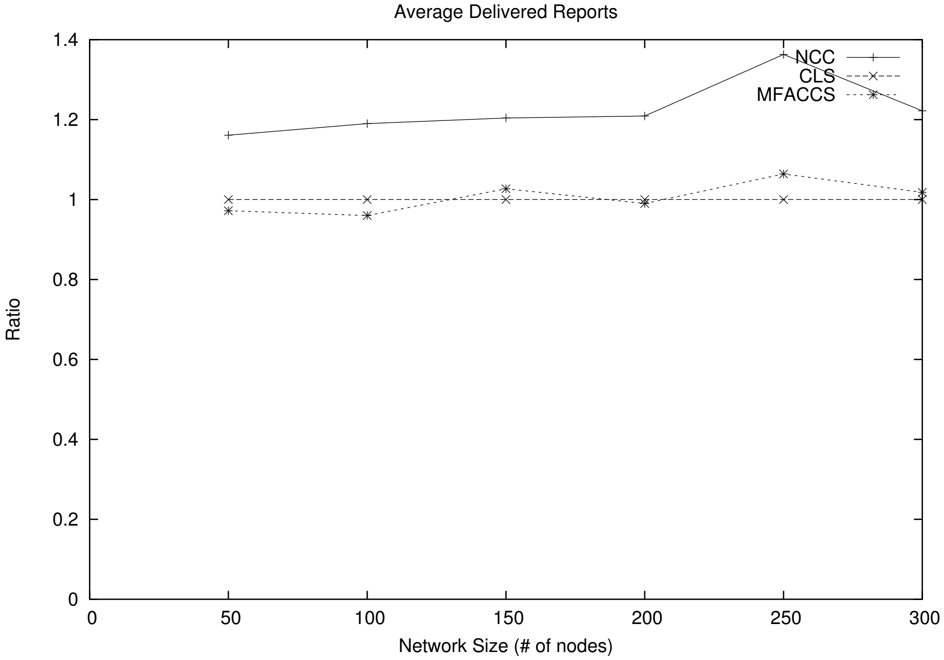

Fig. 27. Average Delivery in Various Networks

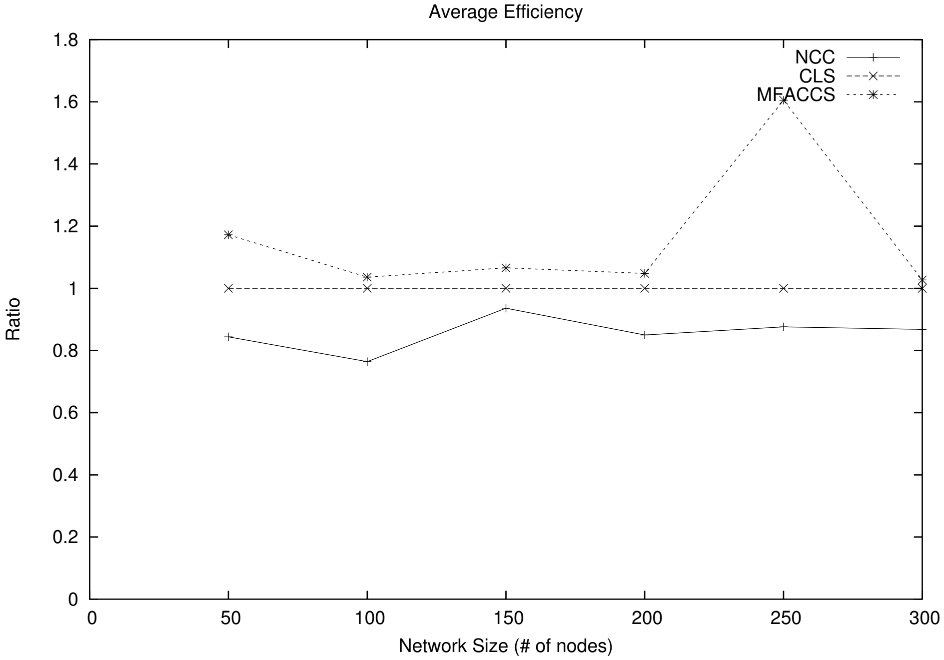

Fig. 28. Average Transmission Efficiency in Various Networks

- Loss: Compared with CLS, MFACCS reduces around 60 percent of loss.
- Energy Dissipation: MFACCS saves around 30 percent energy by its congestion control and spends almost one tenth less energy than CLS.
- Successfully Delivered Reports: MFACCS performs almost equally as CLS and suffers 20 percent loss of delivered reports than NCC.
- Efficiency: MFACCS achieves higher transmission efficiency in all six networks.

Similar results also hold for scenarios with disabled RTS/CTS in 802.11.

E. Conclusions

First, MFACCS is confirmed to be capable of controlling congestion in all above simulations.

Second, as simulation results show, MFACCS can greatly shorten the latency in various sizes of networks. This feature grants MFACCS very good applicability to delay-sensitive applications in sensor networks.

Third, MFACCS displays a characteristic: energy-efficient, which is of great importance in sensor networks. One thing needs to point out is that the proposed scheme MFACCS obtains energy saving while losing some amount of event reports. For applications with strict demand of reliability (in term of received reports), more gentle rate decreasing parameters (β_0 and p) should be considered.

CHAPTER VI

SUMMARY

In this thesis, several properties of sensor networks are studied in time of congestion. Due to the random access of the medium, queue length and transmission latency cannot provide accurate indication of congestion level. Packet loss cannot be used to detect congestion since corruption, collision and node failure can also cause packet dropping.

In sensor networks, a couple of adjacent nodes will have their queues building up when congestion occurs. This thesis address this phenomenon by analysis and simulation. The number of non-empty queues along a path provides an indication of congestion level.

Based on the observation of multiple queues, a congestion control scheme is proposed in this thesis. It uses a queue counter (QC) in each data packet to track the network state. Once congestion is determined to be occurring, a midway on-demand feedback carrying information of congestion level will be sent to the source. Upon receiving the feedback, the source can take proper actions to mitigate or get rid of the congestion. Adaptive additive increase and multiplicative decrease (AIMD) is implemented by using the number of occupied queues.

Extensive simulations have been done to compare the proposed scheme with two other defined schemes. Results show that the proposed scheme can handle congestion very well. This scheme is verified to be energy-efficient and greatly shorten the transmission latency.

REFERENCES

[1] D.-M. Chiu and R. Jain, “Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks,” *Computer Networks and ISDN Systems*, vol. 17, pp. 1-14, 1989.

[2] J. Bolot and A. Shankar, “Dynamical Behavior of Rate-Based Flow Control Mechanisms,” *ACM SIGCOMM Computer Communication Review*, vol. 20, no. 2, pp. 35-49, April 1990.

[3] C.-Y. Wan, S. Eisenman and A. Campbell, “CODA:Congestion Detection and Avoidance in Sensor Networks,” in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, Los Angeles, California, USA, November 2003, pp. 266-279.

[4] Y. Sankarasubramaniam, O. Akan and L. Akyildiz, “ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks,” in *MobiHoc’03*, Annapolis, Maryland, USA, June 2003, pp. 177-188.

[5] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, “Directed Diffusion for Wireless Sensor Networking,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2-16, February 2003.

[6] C.-Y. Wan, A. Campbell and L. Krishnamurthy, “PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks,” in *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, Georgia, USA, September 2002, pp. 1-11.

[7] F. Stann and J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks," in *Proceedings of the 1st International Workshop on Sensor Net Protocols and Applications*, Anchorage, Alaska, USA, May 2003, pp. 102-112.

[8] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang and M. Gerla, "The Impact of Multihop Wireless Channel on TCP Throughput and Loss," *INFOCOM 2003*, vol. 3, pp. 1744-1753, 2003.

[9] K. Xu, M. Gerla, L. Qi and Y. Shu, "Enhancing TCP Fairness in Ad Hoc Wireless Networks Using Neighbourhood RED," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, San Diego, California, USA, September 2003, pp. 16-28.

[10] S. Ray, J. Carruthers and D. Starobinski, "RTS/CTS-Induced Congestion in Ad Hoc Wireless LANs," *WCNC 2003*, vol. 3, pp. 1516-1521, 2003.

[11] W. Ye, J. Heidemann and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *INFOCOM 2002*, vol. 3, pp. 1567-1576, June 2002.

[12] "Part 11:Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specifications," *ANSI/IEEE Std 802.11 1999 Edition*, 1999.

[13] S. Floyd, M. Handley, J. Padhye and J. Widmer, "Equation-Based Congestion Control for Unicast Applications," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Stockholm, Sweden, August 2000, pp. 43-56.

[14] "The Network Simulator - ns-2", Information Science Institute of the University of Southern California, March 2004, Available from http://www.isi.edu/nsnam/ns/.

VITA

Yunli Xiong was born and raised in China. He received his B.S. and M.S. degrees from Shanghai Jiao Tong University, China. Before studying at Texas A&M University, he worked in the communication industry for three years.

The typist for this thesis was Yunli Xiong.