BEST: Journal of Recent Trends in General
Engineering & Research (BEST: JRTGER)
Vol. 2, Issue 1, Jun 2017, 19-24

IMPROVEMENT IN ENERGY AND AVOIDING PACKET ERRORS IN TCP BY CODA MECHANISM

C RAM KUMAR[1], B VIJAYALAKSHMI [2] & C RAMESH[3] & S CHENTHUR PANDIAN[4]

[1]Assistant Professor, ECE, SNS College of Engineering, Coimbatore, India

[2]Assistant Professor, ECE, Sri Ramakrishna Engineering College, Coimbatore, India

[3]Senior Engineer, EEE, Igate Patni, Bangalore, India

[4]Principal, EEE, SNS College of Technology, Coimbatore, India

ABSTRACT

In order to eradicate Congestion problem, an Enhanced version of congestion Control is proposed called ECODA (Enhanced Congestion Detection and avoidance for Multiple Class of Traffic in Sensor Networks). Using three mechanisms which uses dual buffer Thresholds/Weighted buffer difference, Flexible Queue Scheduler and bottleneck based Control Schemes. ECODA effectively Controls Congestion problems for different class of traffic using MAC layer. ECODA has a flexible queue scheduler and packets are scheduled according to their priority. Many applications would require fast data transfer in high-speed wireless networks nowadays. However, due to its conservative congestion control algorithm, Transmission Control Protocol (TCP) cannot effectively utilize the network capacity in lossy wireless networks. In this paper, we propose a receiver-assisted congestion control mechanism (RACC) in which the sender performs loss-based control, while the receiver is performing delay-based control. The receiver measures the network bandwidth based on the packet interarrival interval and uses it to compute a congestion window size deemed appropriate for the sender. After receiving the advertised value feedback from the receiver, the sender then uses the additive increase and multiplicative decrease (AIMD) mechanism to compute the correct congestion window size to be used. Our mechanism can mitigate the effect of wireless losses, alleviate the timeout effect, and therefore make better use of network bandwidth and also our mechanism can outperform conventional TCP in high-speed and lossy wireless environments. It can reduce packet loss, improve efficiency and lower delay.

KEYWORDS: CODA, Queue Scheduler, RMST and Buffer State

INTRODUCTION

Wireless sensor networks (WSNs) have been widely applied to habitat monitoring, real-time target tracking, environment surveillance and healthcare, etc. They are different from traditional wireless networks in several aspects. Commonly, sensor nodes are restricted in computation, storage, communication bandwidth, and, most importantly, energy supply. Extensive studies have been carried out in recent years on the physical layer, the media access control (MAC) layer, the network layer and transport layer in WSNs.

The event-driven nature of WSNs leads to unpredictable network load. Typically, WSNs operate under idle or light load and then suddenly become active in response to a detected event. When the events have been detected, the information in transit is of great importance. However, the busty traffic that results from the detected events can easily cause congestion in the networks. When congestion happens, the network throughput and coverage fidelity are penalized.

So, congestion control is a critical issue in sensor networks. In WSNs, congestion can be divided into transient congestion and persistent congestion. Transient congestion is caused by link variations, and persistent congestion is caused by source data sending rate. Congestion control mechanism can be classified into end-to-end congestion control and hop-by-hop congestion control. End-to-end congestion control performs exact rate adjustment at source and intermediate nodes according to current QoS level at sink node. The drawback of end-to-end congestion control mechanism is that it heavily relies on round-trip time (RTT), which results in slow response and low convergence. In contrast, hop-by-hop congestion control has faster response. Currently, there are extensive studies to address congestion problems in WSNs. Some papers provide reliable end-to-end data delivery from every sensor to a sink and hop by hop congestion control at every intermediate node on the path from source to sink. However, how to ensure weighted fairness for multiple class of traffic among sensors is not well address by previous research. Congestion Detection and Avoidance (CODA) by jointly sampling the channel loading during every epoch and monitoring buffer length of being filled to judge if congestion happens or not. For transient congestion, the node sends explicit backpressure messages to its neighbors which further propagate the message to upstream source nodes depending their local buffer occupancy or channel loading, every node receive the backpressure messages will lower down their sending rate except the designed node which has the priority to access channel. However, the backpressure message may intensify congestion due to high channel loading when congestion happens. For persist congestion, CODA needs explicit ACK from sink, if insufficient ACK reaches the source, the source will lower down its sending rate. However the explicit ACK waste much energy and the loss the ACK due to link quality will give a false congestion signal to the source and affect the network throughput. CODA can’t differentiate bottleneck link either.

LITERATURE SURVEY

An Energy-Efficient Mac Protocol

The sensor-MAC (S-MAC), a new MAC protocol explicitly designed for wireless sensor networks. While reducing energy consumption is the primary goal in my design and protocol also has good scalability and collision avoidance capability. It achieves good scalability and collision avoidance by utilizing a combined scheduling and contention scheme. To achieve the primary goal of energy efficiency, It is required to identify what are the main sources that cause inefficient use of energy as well as what trade-offs It is can make to reduce energy consumption.

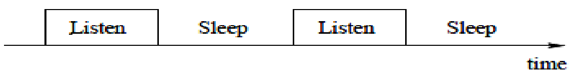

Figure 1: Periodic Listen and Sleep

The basic scheme is shown in Fig 1. Each node goes to sleep for some time, and then wakes up and listens to see if any other node wants to talk to it. During sleep, the node turns off its radio, and sets a timer to awake it later. The duration of time for listening and sleeping can be selected according to different application scenarios. For simplicity these values are the same for all the nodes.

SENSOR NETWORKS COMMUNICATION

The sensor nodes are usually scattered in a sensor field as shown in Figure 2. Each of these scattered sensor nodes has the capabilities to collect data and route data back to the sink and the end users. Data are routed back to the end user by a multihop infrastructure less architecture through the sink as shown in Figure 2.The sink may communicate with the task manager node via Internet or Satellite.

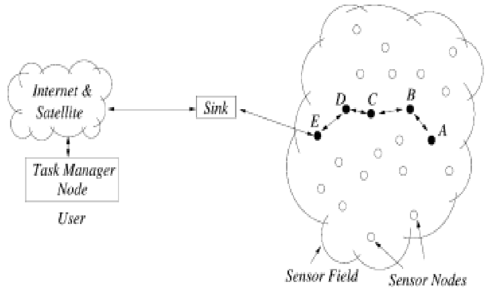

Figure 2: Sensor Nodes Scattered in a Sensor Field

The protocol stack used by the sink and all sensor nodes is given in Figure 2. This protocol stack combines power and routing awareness, integrates data with networking protocols, communicates power efficiently through the wireless medium, and promotes cooperative efforts of sensor nodes. The protocol stack consists of the application layer, transport layer, network layer, data link layer, physical layer, power management plane, mobility management plane, and task management plane.

Existing System

In the literature, many works have been conducted on congestion control, congestion mitigation, and reliable transmission in WSNs. Existing works can generally be classified into three categories.

The first category consists of transport protocols that provide end-to-end reliability without congestion control. Reliable multi segment transport (RMST) is an example of these protocols. RMST is a hop-by-hop reliable transport protocol specially designed to run on top of directed diffusion in which packet loss is recovered hop by hop using caches in the intermediate nodes. RMST guarantees reliability but is designed for more capable sensor nodes. In addition, in RMST, the node’s transmit rate is manually set by a system administrator.

The second category consists of protocols with centralized congestion control schemes. ESRT classifies a network into five regions. It adjusts source packet data sending rate such that network stays in a state where sufficient number of packets arrive at a sink without producing congestion. ESRT’s rate allocation is centrally computed i.e., the base station periodically counts the number of received sensor readings and ret asks the sensors by broadcasting a new transmission rate. Due to the drawbacks of centralized scheme, ESRT cannot deal with transient congestion efficiently. RCRT is another centralized transport protocol in which all functionalities including congestion detection, rate adaptation, and rate allocation are implemented at sink node. Although RCRT’s performance is good; it can’t differentiate flows unconstrained in bottleneck regions. Also, RCRT’s convergence is too slow when the network has highly varying RTTs.

The third category consists of protocols with distributed congestion control schemes. Fusion uses hop-by-hop flow control, rate limiting, and prioritized MAC to alleviate congestion. With this combination, Fusion achieves higher good put and better fairness with heavy loads than previous schemes. Congestion detection and avoidance (CODA) in sensor networks is another congestion mitigation strategy, it provides a comprehensive discussion on congestion control and proposes an open-loop hop-by-hop backpressure mechanism and closed-loop multi-source regulation scheme.

For transient congestion, each sensor monitors channel utilization and buffer occupancy level to detect congestion. For persistent congestion, source requires sink's feedback to maintain its data rate. Unlike Fusion, CODA doesn't explicitly focus on per-source fairness. IFRC and CCF are both congestion control protocols to ensure fairness. In IFRC, every node adopts multi-level buffer thresholds. When a node's buffer exceeds the threshold, it asks its neighbors to decrease data sending rate and maintain its buffer utilization less than a predefined level.

PROPOSED SYSTEM

Congestion Detection

In order to precisely measure local congestion level at each node, we propose dual buffer thresholds and weighted buffer difference for congestion detection. Buffer is defined as three states, "accept state", "filter state" and "reject state", as Fig. 4 indicate. Two thresholds Q_{min} and Q_{max} are used to border different buffer states. Different buffer states reflect different channel loading, corresponding strategy is adopted to accept or reject packets in different states. It is necessary to point out that, in this paper, "reject state" not means to reject all incoming packets, but it means that most of packets will be rejected because buffer utilization is too high. Every node which has data to send monitors its buffer and piggybacks its *WR* and *WQ* in its outgoing packets. If a node's buffer occupancy exceeds a certain threshold and its data has higher priority among neighborhood, the corresponding congestion level bit in the outgoing packet header is set.

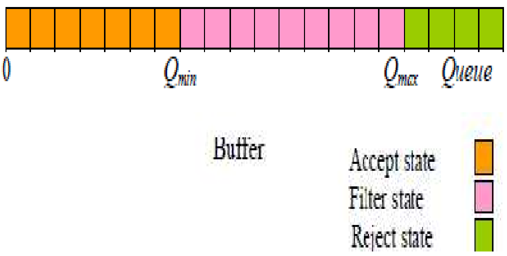

Figure 3: Buffer State

Where $k \in$ neighbor (node i); N is the total number of packets in the buffer. If $WQDnode\ i\ (t+\Delta\ t) \geq 0$, it means that the data of node i is the most important among its neighbors. If congestion happens, other nodes should lower down their data sending rate to mitigate node i's congestion.

Flexible Queue Scheduler and Weighted Fairness

When congestion occurs, packets are dropped to alleviate congestion. Most of the queue schedulers, both in wired and wireless networks, drop packets from the tail rather than any position in the queue. But tail-dropping does not work well. For instance, if the queue in a sensor node is nearly full and dominated with low priority packets, when a high priority packet arrives, it is better to drop a low priority packet rather than the high priority packet. With tail-dropping, the high priority packet may be dropped due to queue overflow. As indicated in Fig. 4, there are two sub-queues. One is for local generated traffic, and the other is for route-through traffic. To ensure fairness, the algorithm scans the route-through traffic queue from head to tail.

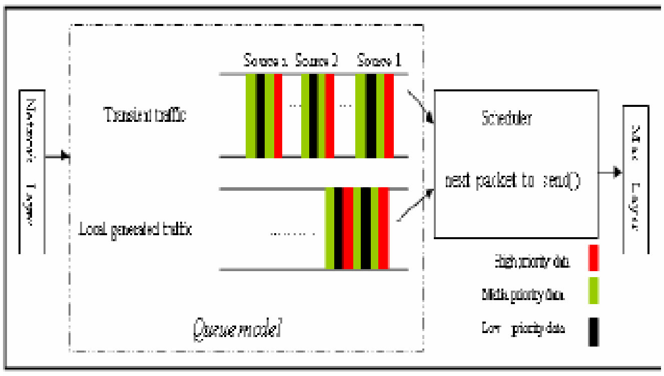

Figure 4: Queue Model

One packet from one source is sent from route-through traffic queue, and then a local generated packet is sent. Suppose that the total buffer size of a node is *Q*. In the queue, the dynamic priorities of packets are respectively denoted as *DP1, DP2 DPn*, where n is the total number of priorities. The total number of packets are *N=NDP1+NDP2+NDP3+NDPn*.

SIMULATION RESULTS

Topology is built using NS2 simulator. With tail-dropping, the high priority packet may be dropped due to queue overflow. In this technique, TCP layer is used to avoid packet drop even though the transmission speed of the packets is very high. This algorithm appears to be robust to packet errors by maintaining a high throughput. Throughput is measured in simulation. Throughput is shown in the following graph.

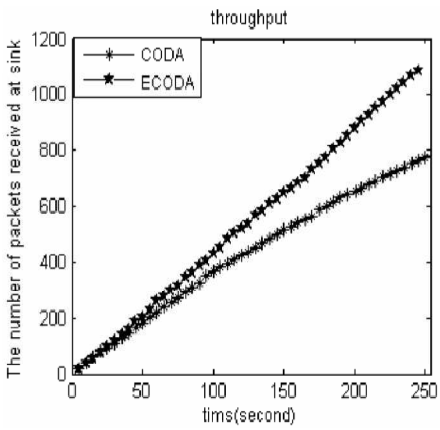

Figure 5: Throughput Comparison

Figure 5 shows Throughput comparison between (Existing system) shows that the throughput in proposed system and * shows that the throughput in existing system. Throughput is maximized in the proposed system and thus has better performance than the existing system.

CONCLUSIONS

In this paper, a congestion control protocol (ECODA) is proposed. ECODA has a flexible queue scheduler and packets are scheduled according to their priority. In ECODA, it has overcome the drawbacks of packet drop and improves the energy efficiency. In this technique, TCP layer is used to avoid packet drop even though the transmission speed of the packets is very high. This algorithm appears to be robust to packet errors by maintaining a high throughput. It can reduce packet loss; improve energy efficiency, and lower delay. On comparison of TCP layer with MAC layer, we can attain maximum throughput and improve the efficiency.

REFERENCES

1. H.Kim, J. Song, and S. Lee "Energy-Efficient Traffic Scheduling in IEEE 802.15.4 for Home Automation Networks," *IEEE Transaction on Consumer Electronics*, vol.53, issue.2. 369-374, May 2007.
2. H.Oh, H.Bahn, and K. Chae, "An Energy-Efficient Sensor Routing Scheme for Home Automation Networks," *IEEE Transaction on Consumer Electronics*, vol. 51, issue. 3, pp. 836-839, August 2005.
3. D.Lee and K.Chung, "Adaptive Duty-cycle Based Congestion Control for Home Automation Networks," *IEEE Transaction on Consumer Electronics*, vol.56, No.1, February 2010.
4. A.Cerpa, J.Elson, M.Hamilton, and J. Zhao, "Habitat monitoring: Application drive for wireless communications technology," in Proc.*ACM SIGCOMM Workshop Data Commun*. Latin Amer, Caribbean. San Jose, Costa Rica, Apr. 2001.
5. T.He, et al. Achieving Real-Time Target Tracking Using Wireless Sensor Networks, *ACM Trans. On Embedded Computing System*, 2007.
6. Holman et al, R.Holman, J.Stanley and T. Ozkan-Haller, Applying video sensor networks to nearshore environment monitoring, *IEEE Pervasive Computing* 2(4) (2003), pp. 14-21
7. Y. Sankarasubramaniam, O. Akan, I. Akyildiz, "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks," in Proc. of *ACM MobiHoc* '03.