Hybrid Congestion Control Protocol in Wireless Sensor Networks

Jang-Ping Sheu
Dep. of Computer Science
National Tsing Hua University, Hsinchu, Taiwan

, and Wei-Kai Hu
Dep. of Computer Science and Information Engineering
National Central University, Chungli, Taiwan

***Abstract*—In wireless sensor networks, congestion occurs when every sensor node will send the event it has sensed to a sink node. This operation makes the sensors closer to the sink, resulting in congestion. Congestion may cause packets loss, lower network throughput and sensor energy waste. In this paper, we propose a hybrid congestion control protocol that considers not only the packets delivery rate but also retains the buffer size of each node. The proposed protocol may avoid packets drop due to traffic congestion and improve the network throughput. The simulation results show that the performance of the proposed protocol is better than the previous works.**

I. Introduction

A wireless sensor network is constrained by memory space, computation capacity, communication bandwidth, and energy supply. Therefore, a lot of academic research topics are discussing how to prolong the whole network lifetime. For example, the power control issue [1] and energy-aware routing [2] control the energy transmission range to save power. The load balance issues [3]–[6] average the work load of each node. Besides, the congestion control protocols [7]–[12] can mitigate network bottlenecks and improve the network performance.

Studies have been trying to address congestion because it causes a lot of problems. The energy spent by upstream neighbors on a packet is wasted when the packet is dropped. When congestions happen without any control protocol being implemented, more packets will be dropped and more energy is wasted. The congestion occurring in a node may result in a quick decline of a network throughput. The sudden surge of data from hundreds or even thousands of sensors must be delivered to a small number of sinks, which may cause congestion, especially nodes near the sinks. To address this challenge, we must solve the flow control and fairness problems. The flow control seeks to manage the date rate from upstream neighbors once congestion happens. The fairness problem aims to ensure that the nodes have equal or weighted probability to share the network bandwidth [10].

In this paper, a Hybrid Congestion Control Protocol (HCCP), considering both the packets delivery rate and remaining buffer size of each node is proposed. The scheme does not need to maintain the global flow information and each node makes use of its current remaining buffer size and net flow size to calculate its congestion degree information. The congestion degree is defined to reflect the current congestion level at each node. Then, the congestion degree is exchanged periodically between neighbors. Therefore, each node can use its congestion degree and neighbors' congestion degrees to prevent the emergence of congestion. The simulations show that our protocol can reduce packets drop rate and increase packets delivery ratio effectively.

This work was supported by the National Science Council of R.O.C. under grant NSC 95-2221-E-008-018.

II. Related Works

A. *Rate-Based Scheme*

The basic idea of the rate-based scheme is for a forwarding node to estimate the number of flows coming from each upstream neighbor and assign transmission rate based on fairness once congestion is detected. In [7] an event-to-sink reliable transport protocol is proposed for congestion control. Each sensor node monitors its local buffer and sets a congestion notification bit in the packets forwarded to the sink if the buffers overflow. When the sink receives a packet with the congestion notification, it infers congestion and broadcasts a control signal notifying all source nodes to reduce their reporting frequency. A distributed congestion detection and avoidance protocol is proposed in [8]. The authors in [9] propose a mitigating congestion protocol which combines three congestion mitigating mechanisms: hop-by-hop flow control, rate limiting and prioritized MAC layer. This scheme requires a tree routing structure to work correctly. A localized algorithm for aggregate fairness protocol is proposed in [10]. When a sensor receives more packets than it can forward, the sensor will calculate and allocate the date rates of upstream neighbors by a weighted fairness function. However, the fairness function of this congestion control protocol was not considered carefully with the remaining buffer size and transmission rate concurrently.

B. *Buffer-Based Scheme*

In the buffer-based scheme, a sensor i sends a packet to its downstream neighbor j only when j has buffer space to hold the packet. In [11] a congestion avoidance protocol based on lightweight buffer management in sensor networks is proposed. This scheme uses 1/6-buufer algorithm to solve the hidden-terminal problem. Every sensor advertises only one sixth of its remaining buffers. Although it can realize and guarantee the packet does not drop in the forwarding way, the buffer utilization is low.

Most congestion control protocols do not consider buffer state and date rate at the same time. Here, we show the drawback of rate-based scheme and buffer-based scheme. We assume that the packet length is fixed. The unit of buffer size is packet and the data rate is number of packets per unit time. In Fig. 2.1(a), assume that the average data rate $R_{a,d}$, $R_{b,d}$ and $R_{d,c}$ in current transmission period are 9, 6 and 5, respectively and the remaining buffer size of *d* is 9. In Fig. 2.1(b), assume that the average data rate $R_{a,d}$, $R_{b,d}$ and $R_{d,c}$ are 6, 3, and 8, respectively and the remaining buffer size of *d* is 3. Considering the buffer-based scheme, the congestion degree of *d* in Fig. 2.1(b) is higher than that of in Fig. 2.1(a). However, considering the impact of data rate, the congestion degree of *d* in Fig. 2.1(b) is smaller than that of in Fig. 2.1(a). This is because the net flow size of *d* is 10 in Fig. 2.1(a) and congestion will happen in the next time period but the net flow size of *d* in Fig. 2.1(b) is only one. To avoid the above drawback of rate-based and buffer-based schemes, our congestion control protocol takes both of the buffer capacity and the data rate into considerations.

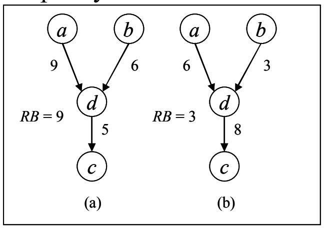

Fig. 2.1 Examples of congestion control protocols

III. HYBRID CONGESTION CONTROL PROTOCOL (HCCP)

There are two types of congestion in WSNs: channel collision and buffer congestion. A growing number of sensor networks use CSMA for medium access. The CSMA can improve channel collision but cannot solve the congestion problem. It may cause the buffer of a sensor overflow if several neighbors of the sensor have packets with high data rate to the sensor. Our protocol exists between the network and MAC layers. The relay traffic rate of node *i* (r_r^i) is received from its upstream neighbors through the MAC layer of *i*. The source traffic rate of node *i* (r_s^i) is generated by node *i*. The total data rate (r_t^i) of *i* through the network layer to MAC layer are converged both r_r^i and r_s^i. So that, the $r_t^i = r_s^i + r_r^i$. The forward data rate (r_f^i) is the total rate that all of its downstream neighbors allow it to pass. A packet could be queued at buffer at network layer when the forward rate r_f^i is smaller than the total rate r_t^i. If r_t^i is continuously bigger than r_f^i, the buffers will fill up quickly. Finally, the buffers will overflow, and the congestion will take place. In order to avoid the congestion, we can reduce the r_s^i, r_r^i or both.

In this paper, we study the problem of data gathering for a sensor network, from where all source nodes send packets to the sink. Every sensor node sends message from many-to-one convergent traffic to the sink when an event is sensed. The sensors have one or more parents and may have many children and grandchildren nodes or not. We assume each link is symmetric. Each sensor node has two type neighbor nodes: one is a group of upstream neighbors and another is a group of downstream ones. Let U_i be the set of upstream neighbors of node *i*, which pass through *i* and forward to the sink. Let D_i be the set of downstream neighbors of node *i*, which are the next hop on the routing path from *i* to the sink. We assume that each sensor node has a counter that can calculate the data rate from upstream neighbors and the data rate to downstream neighbors. An upstream neighbor must be its parent and a downstream neighbor must be its children. The remaining buffer size of *i* is represented by RB_i and the net flow size of *i* is represented by NS_i. Assume that the packet length is fixed. Each sensor node *i* has a congestion degree CD_i which is the index of congestion level. According to the congestion degree, we can classify the current traffic load of each node into light and heavy states.

In the following, we present our HCCP, which mitigates congestion and allocates appropriate source rate to the sink node for sensor networks. HCCP comprises two phases: congestion detection phase and data rate adjustment phase. HCCP does not maintain the global flow information. Each node makes use of its current remaining buffer size and net flow size to calculate its congestion degree. And the congestion degree is exchanged periodically between neighbors. Therefore, each node can use its congestion degree and its neighbors' congestion degree to prevent congestion.

A. Congestion Detection Phase

Since each sensor may have one or more upstream and downstream neighbors, there exist many input flows from upstream neighbors and output flows to downstream neighbors. Congestion probably occurs when the flows cross each other complicatedly. When a sensor node detects congestion, it may cost a lot of time and network bandwidth to solve the congestion problem. Thus, we would like to detect the congestion in advance and take the preventive measures. In this phase, the congestion degree is predicted based on a time period *T*. Each sensor will count the current upstream and downstream data rates of its neighbors and predict whether congestion will happen or not in the next time period *T*. The time period *T* can be neither too long nor too short. If *T* is too short, it will cause high control overhead due to the frequently congestion detection. If *T* is too long, the congestion will happen before time is expired. It will cause low performance of congestion control.

For a sensor *i*, if the flows rate coming from the upstream neighbors is far greater than the flows rate it can forward to downstream neighbors, and the buffer of sensor *i* cannot hold the net flow size in the next time period, it will suppress the upstream neighbors to slow down their data rate and the buffer state of sensor *i* is set as heavy. If the remaining buffer size of sensor *i* in the next time period is greater than or equal to the total flows size coming from the upstream neighbors minus the total flows size forwarding to downstream neighbors, we define the buffer state of sensor *i* is in light state. When the buffer enters a light state, it guarantees that congestion will not happen in the next time period, we do not need to perform any congestion control. Otherwise, when the buffer state enters heavy state the congestion may happen in the next time period, we must trigger the congestion control process to avoid the congestion and assign the proper data rate for its neighbors according to their congestion degrees.

In order to avoid the buffer overflow, a sensor i must estimate the net flow size from all neighbors within a time period. Let $R_{i,j}\{\forall i \in N, j \in D_i\}$ be the average downstream date rate from node i to j per unit time. Let $R_{k,i}\{\forall i \in N, k \in U_i\}$ be the average upstream data rate from node k to i per unit time. The $R_{i,j}$ and $R_{k,i}$ can be easily measured at each sensor i by a counter on a packet-by-packet basis. Then a net flow size NS_i is the source traffic rate of sensor i (r_s^i) plus all flows from upstream neighbors of sensor i and minus all flows that sensor i can forward to downstream neighbors during a time period T as follows:

$$NS_i = (r_s^i + \sum_{j \in U_i} R_{j,i} - \sum_{k \in D_i} R_{i,k}) \times T, \forall i,j,k \in N \quad (1)$$

In order to indicate the index of congestion, we define a congestion degree CD_i, which is the remaining buffer size minus the net flow size of each sensor i during a time period T as follows:

$$CD_i = RB_i - NS_i \quad (2)$$

If the CD_i is smaller than 0, the buffer state of i will become heavy and congestion may happen in the next time period. The sensor i will broadcast a suppressive message to advertise its neighbors to slow down their data rates. For sensors to know the congestion degrees of their neighboring nodes, they will advertise their congestion degrees to each other. For each sensor, the advertisement is triggered by either of the following two events: (1) in the beginning of each time period T and (2) the buffer state from light to heavy. In order to reduce the control message overhead, if a sensor has the data traffic, we piggyback the congestion degree in the header of the data packet.

B. Data Rate Adjustment Phase

Assume that the sensors will forward the data packets to the downstream neighbors as fast as possible. When sensor i obtains the congestion degrees of its upstream and downstream neighbors, it will calculate the value of r_t^i and r_f^i, and updates its congestion degree. Once the CD_i of sensor i is larger than or equal to 0, it means that the buffer state of i is light, and therefore, it will do nothing. On the other hand, if the CD_i of sensor i is smaller than 0, it will suppress the data rate of upstream neighbors of i. In order to allocate effectively date rates to upstream neighbors, the upstream neighbors that tend to congest will be allocated more data rate. Sensor i can estimate each upstream neighbor's tendency towards congestion by CD_x and $R_{x,i}$. We define a tendency congestion degree $\alpha_i(x)$ represents the degree of congestion probability of x if the total traffic from x to i is prohibited in a time period T. Then we have

$$\alpha_i(x) = CD_x - R_{x,i} \times T, \forall x \in U_i \quad (3)$$

If $\alpha_i(x)$ is less than 0, it means that if sensor i suppress the data rate $R_{x,i}$, sensor x may congestion in the next time period. Otherwise, if $\alpha_i(x)$ is larger than or equal to 0, it represents that congestion will not happen in the next time period even if sensor i suppresses the data rate $R_{x,i}$. Therefore, the more negative the value of $\alpha_i(x)$ is, the more data rate will be allocated to x by sensor i. Let SUM_i be the summation of absolute value of $\alpha_i(x) < 0$.

$$SUM_i = \sum_{x \in U_i} |\alpha_i(x)|, \forall \alpha_i(x) < 0 \quad (4)$$

Here, we define a potential traffic capacity PC_i, which is the remaining buffer size plus the sum of flows size that sensor i can forward to its downstream neighbors during a time period T as follows:

$$PC_i = RB_i + \sum_{k \in D_i} R_{i,k} \times T \quad (5)$$

The PC_i represents how many packets that the sensor i can hold from upstream neighbors in the next period. The sensor i will calculate whether its PC_i is enough to satisfy those sensors with $\alpha_i(x) < 0$. Let $PC_i^{'} = PC_i - SUM_i$ be the remaining potential traffic capacity of i. If $PC_I^{'} \geqq 0$, it means that the potential traffic capacity of i can satisfy the requirement of upstream neighbor nodes with $\alpha_i(x) < 0$. Sensor i will first consider to allocate data rate to the sensors whose $\alpha_i(x)$ is less than 0. The remaining potential traffic capacity of i will then allocate to all upstream neighbors evenly. Let N_{U_i} be the number of upstream neighbors of sensor i. Sensor i will allocate the new data rate ($R_{x,i}^{'}$) to its upstream neighbors as follows:

$$\begin{cases} R_{x,i}^{'} = (|\alpha_i(x)| + \frac{PC_i^{'}}{N_{U_i}}) & if(\alpha_i(x) < 0) \\ R_{x,i}^{'} = (\frac{PC_i^{'}}{N_{U_i}}) & if(\alpha_i(x) \geq 0) \end{cases} \quad (6)$$

On the other hand, if $PC_i^{'} < 0$, it means that the potential traffic capacity of i cannot satisfy the requirements of the upstream neighbors with $\alpha_i(x) < 0$. Sensor i will allocate all of the potential traffic capacity to the sensors whose $\alpha_i(x)$ is less than 0 according to the value of $\alpha_i(x)$. The more negative the value of $\alpha_i(x)$ is, the more the data rate is allocated to sensor x. Thus, sensor i will allocate the new data rate to its upstream neighbors as follows:

$$\begin{cases} R_{x,i}^{'} = (PC_i \times \frac{|\alpha_i(x)|}{SUM_i}) & if(\alpha_i(x) < 0) \\ R_{x,i}^{'} = 0 & if(\alpha_i(x) \geq 0) \end{cases} \quad (7)$$

After a sensor suppresses the data rate of its upstream neighbors, it may cause their buffers to overflow and the congestion may happen at these upstream neighbors. However, these nodes will further suppress their upstream neighbors in the same way. This process repeats hop-by-hop towards the source node or leaf nodes. The whole network will reach the most effective congestion free.

Now we give an example to illustrate our congestion control scheme. In Fig. 3.1, the two fields in brackets above each node denote the total flow size from upstream neighbors and the total flow size to downstream neighbors, respectively. The number on each link represents the data rate with which the sensor forwards to its downstream neighbor. For example, in Fig. 3.1, the symbol [3/2] of sensor x represents that the total flow size from upstream neighbors to x is 3 and the total flow size to downstream neighbors of x is 2. Assume that the remaining buffer size of sensors x, y, z, and i are 2, 2, 3, and 0, respectively. The congestion degrees of sensors x, y, z, and i are 1, 1, 0, and -1, respectively. The potential traffic capacity PC_i is 4, and $\alpha_i(x)$, $\alpha_i(y)$, and $\alpha_i(z)$ are 0, -2,

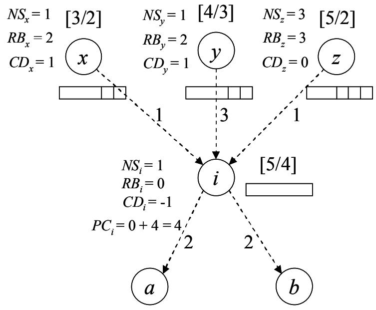

Fig. 3.1 Example of data rate adjustment

-1, respectively. We have $PC_i^{'} = 4 - 3 = 1$. Since $PC_i^{'} > 0$, sensor i will first allocate data rates 2 and 1 to sensors y and z, respectively. Then, the remaining potential traffic capacity of i will be evenly distributed to all upstream sensors. Based on (6), the new data rates of $R^{'}_{x,i}$, $R^{'}_{y,i}$, and $R^{'}_{z,i}$ are 1/3, 7/3, and 4/3, respectively.

IV. Simulation Results

This section presents the simulation results. We measure the performance of our scheme with others in terms of the following metrics: packet drop rate, total source rate, and control overhead. We use ns-2 simulator for our simulations. Five hundreds sensors are randomly placed in a 1,000 m × 1,000 m area. The transmission range of the sensor is 100 m with the transmission rate of 512 kbps. A sink is deployed at the center of the deployment area. There are 100 date source nodes, randomly selected from the 500 sensors. The initial data rate of each source node is configured to four packets per unit of time. It may generate at a lower data rate due to congestion control. Each packet is 40 bytes long. The buffer at each sensor can hold 32 data packets. The simulation time is 200 seconds. We compare our HCCP scheme with rate-based scheme AFA [10] and buffer-based scheme BB [11].

A. *Packet Drop Rate Comparison*

In AFA scheme, a sensor i has a packet to forward j, only if the buffer of j is not full. If the buffer of j is full, i will hold the packet until it overhears a packet piggybacking a non-full buffer state from j. Thus, it does not cause packet drop. The BB scheme must make sure that a sensor i sends a packet to its downstream neighbor j only when j has buffer space to hold the packet. It does not cause packet drop, too. Our congestion control protocol can detect the congestion in advance and take the preventive measures. Therefore, HCCP achieves no packets drop as well as BB and AFA.

B. *Total Source Rate Comparison*

The total source rate is defined as the total number of data packets generated by the data sources per second. Fig. 4.1 demonstrates, for HCCP scheme, how the time period T affects the total source rate changes with respect to simulation time. After time = 160 seconds, the source rates of all simulations is stable. In Fig. 4.1, the smaller time period T is, the larger the total source rate is. The total source rate will not increase anymore when T is smaller than 0.25 seconds. However, a smaller time period will cause higher control overhead due to the frequent congestion detection.

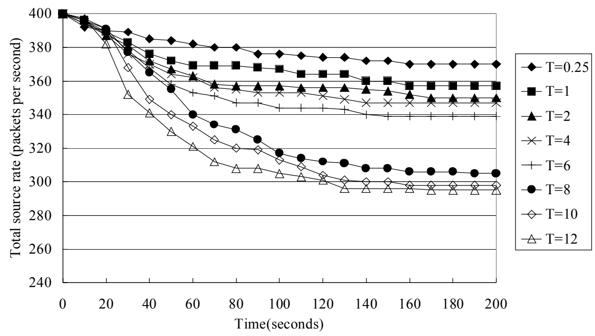

Fig. 4.1 The total source rate changes with various time periods *T*

Fig. 4.2 confirms how much overhead of congestion in HCCP can control with respect to the time period T. The overhead of congestion control is defined as the number of congestion control packets over the number of total delivery packets. When congestion in the next time period is detected, the sensor i will broadcast its congestion degree to advertise the neighbors to suppress the data rate of upstream neighbors. The more the number of congestions detected, the higher cost needed. In Fig. 4.2, the percentage of congestion control overhead is minimum at $T = 4$ and maximum at $T = 0.25$. If T is too long, the congestion will happen before time is expired. It will cause low performance of congestion control. Therefore, the time period T is set as 1, 2, and 4 seconds in the following simulations.

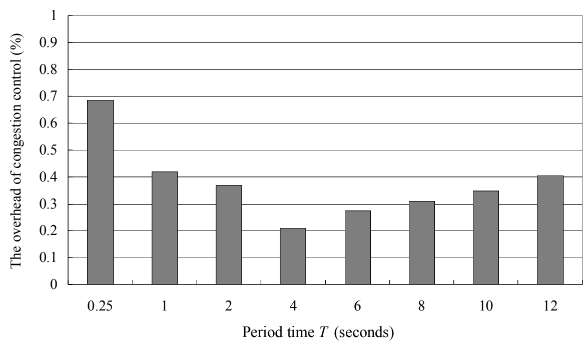

Fig. 4.2 The overhead of congestion control in HCCP with time period *T*

Fig. 4.3 compares the total source rates of three schemes with respect to simulation time. The BB scheme uses the 1/6-buufer algorithm to solve the hidden-terminal problem. Every sensor advertises only one sixth of its remaining buffers.

Therefore, the total source rate of BB is lower than others. AFA is not only utilizing the buffer effectively than BB but it can effectively allocate the data rate of upstream neighbors. The HCCP scheme combines the advantages of buffer-based and rate-based schemes. HCCP considers the packets delivery rate and remaining buffer size of each node concurrently. It can allocate effectively the data rate of upstream neighbors according to their tendency of congestion degrees. These simulations show that our congestion protocol HCCP is able to adjust effectively the proper data rate for sensors and obtains the better total data rate than other schemes. Fig. 4.4 shows the total reduced source rate. The total reduced source rate is defined as the reduction of total source rate over the total source rate. Our proposed protocol has a better performance than other schemes with various initial source rates.

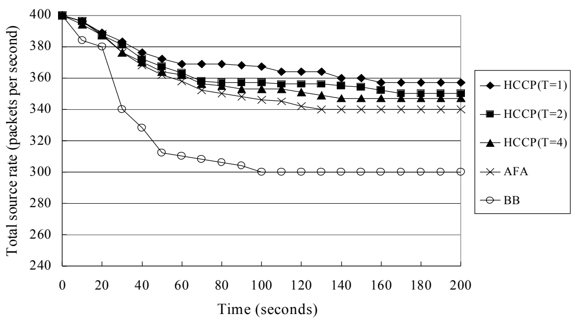

Fig. 4.3 The total source rate comparison with three schemes

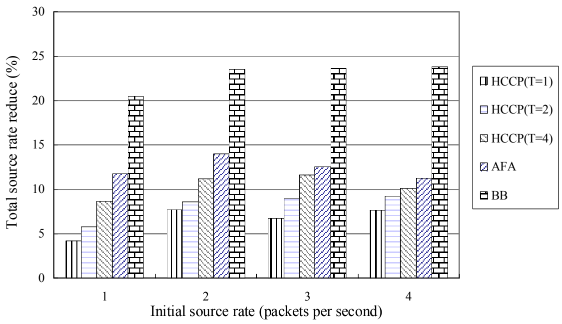

Fig. 4.4 The reduction of total source rate with initial source rates

C. Control Overhead Comparison

This simulation compares the overhead of congestion control with each scheme. The BB scheme always piggybacks its current buffer state by one bit in the frame header of each data packet. The BB scheme does not need to broadcast suppressive massage. Thus, we ignore the control overhead of BB scheme. The AFA and HCCP will broadcast the suppressive message when congestion is detected. Since the HCCP is more conservative than AFA scheme for congestion control, the HCCP needs higher control overhead than AFA scheme. Fig. 4.5 shows the overhead of congestion control with respect to initial source rate. In various initial source rates, the control overhead of our scheme is a little higher than other schemes. Although the maximum ratio of the control overhead in HCCP is up to 0.42% as the initial source rate is 4, the difference between HCCP and AFA is very small.

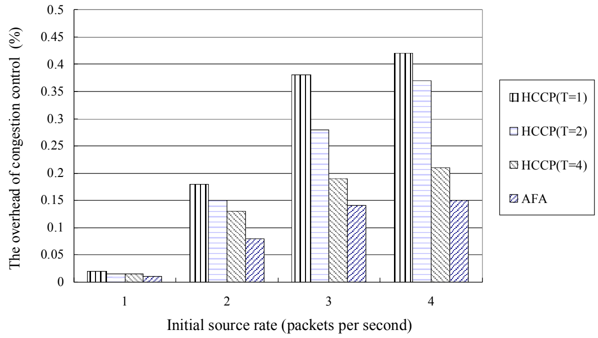

Fig. 4.5 The overhead of congestion control with initial source rates

V. Conclusions

In this paper, we have addressed the problem of congestion control in the sensor networks. We propose a hybrid congestion control protocol, which considers both the packets delivery rate and remaining buffer size of each node. We discuss the congestion control problem including the congestion detection and data rate adjustment. In congestion detection phase, our HCCP detects the congestion in advance with a time period *T* and takes the preventive measures. In data rate adjustment phase, the upstream neighbors that tend to congest will be allocated more data rate. Simulation results show that the performance of our proposed protocol is better than the previous works in terms of total source rate. The control overhead of our proposed scheme is only a little higher than other schemes.

References

[1] S. Lin, J. Zhang, G. Zhou, G. Lin, H. Tian and J. A. Stankovic "ATPC: Adaptive transmission power control for wireless sensor networks" in *Proceedings of* SenSys, pp. 223-236, Nov. 2006

[2] S. C. Huang and R. H. Jan, "Energy-aware, load balanced routing schemes for sensor networks," in *Proceedings of* ICPADS, pp. 419- 425, July 2004.

[3] H. Yang, F. Ye and B. Sikdar, "A dynamic query-tree energy balancing protocol for sensor networks," in *Proceedings of* WCNC, pp. 1715- 1720, March 2004

[4] T. S. Chen, H. W. Tsai and C. P. Chu, "Gathering-load-balanced tree protocol for wireless sensor networks," in *Proceedings of* SUTC, pp. 8-13, 2006.

[5] H. Dai and R. Han, "A node-centric load balancing algorithm for wireless sensor networks," in *Proceedings of* GLOBECOM, pp. 548-552, Dec. 2003

[6] Z. Yang, L. Yuan, X. Du, Q. Zhang, "Multipath load balancing delivery based on decisive energy ratio in wireless sensor networks," in *Proceedings of* RTCSA, pp. 277-280, Aug. 2005.

[7] Y. Sankarasubramaniam, Özgür B. Akan and Ian F. Akyildiz, "ESRT: event-to-sink reliable transport in wireless sensor networks" in *Proceedings of* MobiHoc, pp. 1003-1016, Oct. 2003.

[8] C. Y. Wan, S. B. Eisenman, and A. T. Campbell, "Congestion detection and avoidance in sensor networks" in *Proceedings of* SenSys, pp. 266-279, Nov. 2003.

[9] B. Hull, K. Jamieson and H. Balakrishna, "Mitigating congestion in wireless sensor networks," in *Proceedings of* SenSys, pp. 134-147, Nov. 2004.

[10] S. Chen and Z. Zhang. "Localized algorithm for aggregate fairness in wireless sensor networks," in *Proceedings of* MobiCOM, pp. 274-285, Sep. 2006.

[11] S. Chen and N. Yan, "Congestion avoidance based on lightweight buffer management in sensor networks," in *Proceedings of* ICPADS, pp. 934-946, Sep. 2006.