01 Nov 2007

Predictive Congestion Control Protocol for Wireless Sensor Networks

Maciej Jan Zawodniok
Missouri University of Science and Technology, mjzx9c@mst.edu

Jagannathan Sarangapani
Missouri University of Science and Technology, sarangap@mst.edu

Recommended Citation

M. J. Zawodniok and J. Sarangapani, "Predictive Congestion Control Protocol for Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, Institute of Electrical and Electronics Engineers (IEEE), Nov 2007.
The definitive version is available at https://doi.org/10.1109/TWC.2007.051035

Predictive Congestion Control Protocol for Wireless Sensor Networks

Maciej Zawodniok, *Member, IEEE,* and Sarangapani Jagannathan, *Senior Member, IEEE*

***Abstract*— Available congestion control schemes, for example transport control protocol (TCP), when applied to wireless networks, result in a large number of packet drops, unfair scenarios and low throughputs with a significant amount of wasted energy due to retransmissions. To fully utilize the hop by hop feedback information, this paper presents a novel, decentralized, predictive congestion control (DPCC) for wireless sensor networks (WSN). The DPCC consists of an adaptive flow and adaptive back-off interval selection schemes that work in concert with energy efficient, distributed power control (DPC). The DPCC detects the onset of congestion using queue utilization and the embedded channel estimator algorithm in DPC that predicts the channel quality. Then, an adaptive flow control scheme selects suitable rate which is enforced by the newly proposed adaptive back-off interval selection scheme. An optional adaptive scheduling scheme updates weights associated with each packet to guarantee the weighted fairness during congestion. Closed-loop stability of the proposed hop-by-hop congestion control is demonstrated by using the Lyapunov-based approach. Simulation results show that the DPCC reduces congestion and improves performance over Congestion Detection and Avoidance (CODA) [3] and IEEE 802.11 protocols.**

***Index Terms*— Congestion control, wireless sensor network, Lyapunov stability, control-Lyapunov functions.**

I. INTRODUCTION

NETWORK congestion, which is quite common in wireless networks, occurs when offered load exceeds available capacity or the link bandwidth is reduced due to fading channels. Network congestion causes channel quality to degrade and loss rates rise. It leads to packets drops at the buffers, increased delays, wasted energy, and requires retransmissions. Moreover, traffic flow will be unfair for nodes whose data has to traverse a significant number of hops. This considerably reduces the performance and lifetime of the network. Additionally, wireless sensor networks (WSN) have constraints imposed on energy, memory and bandwidth. Therefore, energy efficient data transmission protocols are required to mitigate congestion resulting from fading channels and excess load. In particular, a congestion control mechanism is needed in order to balance the load, to prevent packet drops, and to avoid network deadlock.

Rigorous work has been done in wired networks on end-to-end congestion control [5]. In spite of several advantages in end-to-end control schemes, the need to propagate the onset of congestion between end-systems makes the approach slow. In general, a hop-by-hop congestion control scheme reacts to congestion faster and is normally preferred to minimize packet losses in wireless networks. Therefore, the proposed scheme uses a novel hop-by-hop flow control algorithm that is capable of predicting the onset of congestion and then gradually reducing the incoming traffic by means of a backpressure signal.

In comparison, the CODA protocol [3] uses both a hop-by-hop and an end-to-end congestion control scheme to react to the congestion by simply dropping packets at the node preceding the congestion area and employing additive increase and multiplicative decrease (AIMD) scheme to control a source's generation rate. Thus, CODA partially minimizes the effects of congestion, and as a result retransmissions still occur. Similar to CODA, Fusion [2] uses a static threshold value for detecting the onset of congestion even though it is normally difficult to determine a suitable threshold value that works in dynamic channel environments. In both CODA and Fusion protocols, nodes use a broadcast message to inform their neighboring nodes the onset of congestion though this message is not guaranteed to reach the sources.

Interference-aware fair rate control (IFRC) protocol [12] uses static queue thresholds to determine congestion level whereas IFRC exercises congestion control by adjusting outgoing rate on each link based on AIMD scheme. Consequently, the IFRC reduces the number of dropped packets by reducing the throughput. By contrast, the proposed scheme varies the rate adoptively based on the current and predicted congestion level. The control parameters in the proposed scheme are updated according to changing environment, while the IFRC [12] and others [2][3] require that the parameters and thresholds have to be selected before each network deployment.

Both IFRC [12] and the proposed scheme support fair bandwidth allocation among the flows. However, IFRC requires nodes to collect rate information from their neighboring nodes thus increasing processing overhead and energy consumption. By contrast, the proposed scheme uses the adaptive back-off selection algorithms at MAC layer for fair allocation of resources among the neighbor nodes without the need for additional radio communication.

Congestion Control and Fairness (CCF) routing scheme [14] uses packet service time at the node as an indicator of congestion. However, the service time alone may be misleading when the incoming rate is equal or lower than the outgoing rate through the channel with high utilization. On the other hand, the Priority-based Congestion Control Protocol (PCCP)

Manuscript received December 17, 2005; revised July 28, 2006, March 21, 2007, and May 30, 2007; accepted June 17, 2007. The associate editor coordinating the review of this paper and approving it for publication was M. Zorzi. Research supported by NSF Grant ECCS #0216191, Air Force Contract and Intelligent Systems Center.

M. Zawodniok and S. Jagannathan are with the Department of Electrical and Computer Engineering, University of Missouri-Rolla, Rolla, MO, 65409 USA (e-mail: {mjzx9c, sarangap}@umr.edu).

Digital Object Identifier 10.1109/TWC.2007.051035.

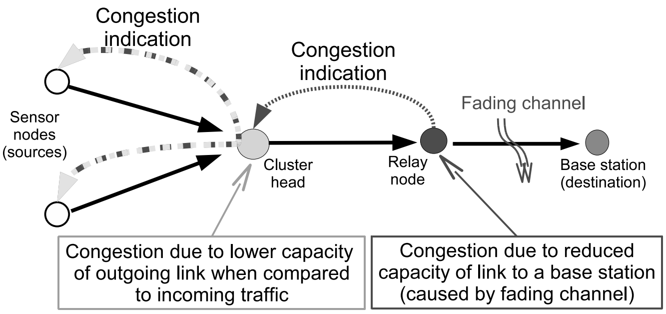

Fig. 1. Congestion in wireless sensor networks.

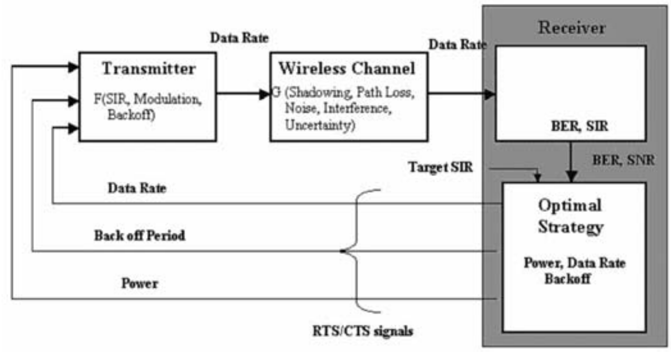

Fig. 2. DPC with rate adaptation.

[15] rectifies this deficiency by observing the ratio between packet service time and inter-arrival time at a given node to asses the congestion level. However, both CCF and PCCP ignore current queue utilization which leads to increased queuing delays and frequent buffer overflows accompanied by increased retransmissions.

Additionally, available protocols [2][3][4][12][13][14][15] do not consider congestion due to fading channels in dynamic environments. Finally, very few analytical results are presented in the literature in terms of guaranteeing the performance of available congestion control protocols. By contrast, the proposed method can predict and mitigate the onset of congestion by gradually reducing the traffic flow defined by using the queue availability and channel state. Besides predicting the onset of congestion, the proposed scheme guarantees convergence to the calculated target outgoing rate by using a novel, adaptive back-off interval selection algorithm. In CSMA/CA-based wireless networks, a back-off selection mechanism is used to provide simultaneous access to a common transmission medium and to vary transmission rates. Many researchers [7][8][9] have focused on the performance analysis of back-off selection schemes for static environments. However, these schemes lack the ability to adapt to a changing channel state, congestion level, and size of the network. By contrast, the proposed algorithm dynamically alters back-off intervals according to current network conditions, for instance the varying number of neighbor nodes and fading channels in contrast with [7] where a uniform density of transmitting nodes is assumed.

The proposed decentralized predictive congestion control (DPCC) uses weights associated with flows to fairly allocate resources during congestion. By adding an optional, dynamic weight adaptation algorithm, weighted fairness can be guaranteed in dynamic environments. Finally, using Lyapunov-based approach, the stability and convergence of the three algorithms, for buffer control, back-off interval selection and dynamic weight adaptation, is demonstrated.

II. Proposed Methodology

The network congestion, shown in Fig. 1, occurs when either the incoming traffic (received and generated) exceeds the capacity of the outgoing link or link bandwidth drops due to channel fading caused by path loss, shadowing and Rayleigh fading. The latter one is common to wireless networks. Therefore the overall objective of this paper is to develop a novel way of utilizing the channel state in rate adaptation and a new MAC protocol using the mathematical framework, capturing channel state, back-off intervals, delay, transmitted power and throughput in contrast with [7][8][9] where the time invariant channel is assumed. Next, an overview of the proposed scheme is presented. Then, the metrics are highlighted.

A. Overview of the proposed scheme

A novel scheme, shown in Fig. 2, is derived based on the channel state, transmitter intended rate, and backlog. The scheme can be summarized in the following steps:

1) 1)The onset of congestion is detected from buffer occupancies at the nodes along with the predicted transmitter power. The rate selection algorithm is then executed at the receiver to determine the appropriate rate (or available bandwidth) for the predicted channel state.
2) The available bandwidth (or rate) is allocated for the flows according to the flow weights to ensure weighted fairness. The weights can be selected initially and held subsequently or updated over time.
3) The DPC and rate information is communicated between nodes on every link.
4) At the transmitter node, a back-off interval is selected by using the proposed scheme based on the assigned outgoing rate.
5) The dynamic weight adaptation scheme can be used to further enhance the throughput while ensuring fairness. Packets at each node can be scheduled by using the adaptive and distributed fair scheduling (ADFS) scheme [6] via flow assigned weights that are updated based on the network state to ensure the fair handling of the packets.

Remark 1: The feedback information which consists of only the rate information is piggybacked to the ACK frame of the MAC protocol. This ensures that the feedback is successfully received by the node in the previous hop in contrast with CODA.

Remark 2: Though a single MAC data rate is considered, the mathematical analysis suggests that changes in routes and MAC data rates (bandwidth) will be accommodated by the outgoing traffic estimation algorithm. Some insight is presented in section IV.C using a simulation scenario.

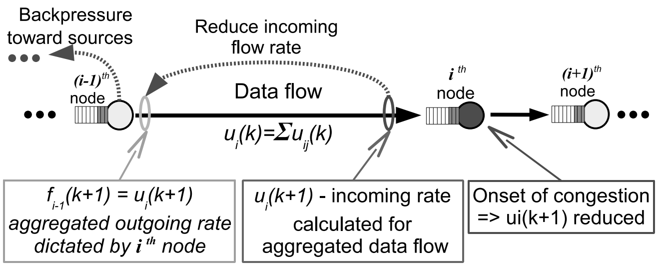

Fig. 3. Rate selection overview.

B. Performance metrics

Packets dropped at the intermediate nodes due to congestion will cause low network throughput and decrease energy efficiency due to retransmissions. Consequently, the *total number of packets dropped at the intermediate nodes* will be considered as a metric for the designed protocol. *Energy efficiency* measured as the number of bits transmitted per joule will be used as the second metric. The *network efficiency* measured as the total throughput at the base station will be taken as an additional metric. *Weighted fairness* will be used as a metric, since congestion can cause unfair handling of flows. Formally, the weighted fairness is defined in terms of fair allocation of resources as

$$\left| \frac{W_f(t_1,t_2)}{\varphi_f} - \frac{W_m(t_1,t_2)}{\varphi_m} \right| = 0 \tag{1}$$

where f and m are considered flows, φ_f is the weight of flow f and $W_f(t_1,t_2)$ is the aggregate service (in bits) received by it in the interval $[t_1,t_2]$. Finally, Fairness Index (FI) [7], which is defined as $FI = \left(\frac{\sum_f T_f}{\varphi_f}\right)^2 \Big/ \eta \sum_f \left(\frac{T_f}{\varphi_f}\right)^2$, where T_f is the throughput of flow f and η is the number of flows will be utilized as a metric.

The DPCC methodology utilizes both rate based control and back-off interval selection schemes along with distributed power control scheme (DPC). Embedded channel quality estimator in DPC is utilized to assess the onset of congestion. The proposed congestion control scheme ensures stability and performance, analytically as summarized next.

III. Adaptive Congestion Control

The adaptive rate selection scheme when implemented at each node acts as a back-pressure signal to minimize the effect of congestion on a hop-by-hop basis by estimating the outgoing traffic flow. Consequently the congestion is alleviated by a) designing suitable back off intervals for each node based on channel state and current traffic; and b) by controlling the flow rates of all nodes including the source nodes to prevent buffer overflowing. Next, we describe the rate and back-off selection algorithms in detail. Then, the data dissemination and fair scheduling are presented.

A. Rate selection based on buffer occupancy

The rate selection scheme takes into account the buffer occupancy and a target outgoing rate. The target rate at the next hop node indicates what the incoming rate should be. The selection of the incoming rate is described next. Consider buffer occupancy at a particular node, as shown in Fig. 3. The change in buffer occupancy in terms of incoming and outgoing traffic at this node is given as

$$q_i(k+1) = Sat_p\left[q_i(k) + Tu_i(k) - f_i(u_{i+1}(k)) + d(k)\right] \tag{2}$$

where T is the measurement interval, $q_i(k)$ is the buffer occupancy of node i at time instant k, $u_i(k)$ is a regulated (incoming) traffic rate, $d(k)$ is an unknown disturbance in traffic, $f_i(\cdot)$ represents an outgoing traffic which is dictated by the next hop node $i+1$ and is disturbed by changes in channel state, and Sat_p is the saturation function that represents the finite-size queue behavior. The regulated incoming traffic rates $u_i(k)$ have to be calculated and propagated as a feedback to the node $i-1$ located on the path to the source, which is then used to estimate the outgoing traffic for this upstream node $f_{i-1}(\cdot)$.

Select the desired buffer occupancy at node i to be q_{id}. Then, buffer occupancy error defined as $e_{bi}(k) = q_i(k) - q_{id}$ can be expressed using (2) as $e_{bi}(k+1) = q_i(k) + T \cdot u_i(k) - f_i(u_{i+1}(k)) + d(k) - q_{id}$. Next, the controller is introduced and its stability analysis is presented.

In the simple case, where the objective is to show that the scheme works, it is assumed that the outgoing traffic $f_i(\cdot)$ value is known. Theorem 1 shows the asymptotic stability of the system. Consequently, the queue level, $q_i(\cdot)$, will closely track the ideal level, q_{id}. Moreover, if the queue level exceeds the ideal level at any time instance, the feedback controller will quickly force the queue level to the target value. The second case presented in Theorem 2 relaxes the assumption of full knowledge about the outgoing flow $f_i(\cdot)$. The stability will hold even when the full knowledge of the outgoing flow is unknown as long as the traffic flow estimation error does not exceed the maximum value f_M. On the other hand, Theorem 3 shows that an adaptive scheme is capable of predicting the outgoing traffic $\hat{f}_i(\cdot)$ with error bounded by the maximum value f_M. In consequence, the proposed controller with adaptive scheme will ensure tracking of the ideal queue level even in the presence of bounded estimation errors in traffic flow.

Case 1: The outgoing traffic $f_i(\cdot)$ is known. Now, define the traffic rate input, $u_i(k)$ as

$$u_i(k) = Sat_p\left(\frac{f_i(u_{i+1}(k)) + (\kappa_{bv}-1)e_{bi}(k)}{T}\right) \tag{3}$$

where κ_{bv} is a gain parameter. In this case, the buffer occupancy error at the time $k+1$ becomes

$$e_{bi}(k+1) = Sat_p\left[\kappa_{bv}e_{bi}(k) + d(k)\right] \tag{4}$$

The buffer occupancy error will become zero as $k \to \infty$, provided $0 < \kappa_{bv} < 1$.

Case 2: The outgoing traffic $f_i(\cdot)$ is unknown and has to be estimated. In such a case, we define the traffic rate input, $u_i(k)$ as

$$u_i(k) = Sat_p\left(\frac{\hat{f}_i(u_{i+1}(k)) + (\kappa_{bv} - 1)e_{bi}(k)}{T}\right) \quad (5)$$

where $\hat{f}_i(u_{i+1}(k))$ is an estimate of the unknown outgoing traffic $f_i(u_{i+1}(k))$. In this case, the buffer occupancy error at the time instant k becomes $e_{bi}(k+1) = Sat_p\left[\kappa_{bv} e_{bi}(k) + \tilde{f}_i(u_{i+1}(k)) + d(k)\right]$ where $\tilde{f}_i(u_{i+1}(k)) = f_i(u_{i+1}(k)) - \hat{f}_i(u_{i+1}(k))$ represents the estimation error of the outgoing traffic.

Theorem 1 (Ideal case): Consider the desired buffer length, q_{id}, to be finite, and the disturbance bound, d_M, to be equal to zero. Let the virtual-source rate for (2) be given by (3). Then the buffer occupancy feedback system is globally asymptotically stable provided $0 < \kappa_{bv\,max}^2 < 1$.

Proof: Let us consider the following Lyapunov function candidate $J = [e_{bi}(k)]^2$. Then the first difference is

$$\Delta J = [e_{bi}(k+1)]^2 - [e_{bi}(k)]^2 \quad (6)$$

Substituting error at time $k+1$ from (4) in (8) yields

$$\Delta J = (\kappa_{bv}^2 - 1)\,[e_{bi}(k)]^2 \leq -(1 - \kappa_{bv\,max}^2)\,\|e_{bi}(k)\|^2 \quad (7)$$

The first difference of Lyapunov function candidate is negative for any time instance k. Hence, the closed-loop system is globally asymptotically stable.

Remark 3: The above theorem using Lyapunov method shows that under the ideal case of no errors in traffic estimation and with no disturbances, the control scheme will ensure that the actual queue level converges to the target value asymptotically.

Theorem 2 (General case): Consider the desired buffer length, q_{id}, to be finite, and the disturbance bound, d_M, to be a known constant. Let the virtual-source rate for (2) be given by (5) with the network traffic is estimated properly such that the approximation error $\tilde{f}_i(\cdot)$ is bounded above by f_M. Then, the buffer occupancy feedback system is globally uniformly bounded provided $0 < \kappa_{bv} < 1$.

Proof: Let us consider Lyapunov function candidate $J = [e_{bi}(k)]^2$. Then, the first difference is

$$\Delta J = \left[\kappa_{bv} e_{bi}(k) + \tilde{f}_i(u_{i+1}(k)) + d(k)\right]^2 - [e_{bi}(k)]^2 \quad (8)$$

The stability condition $\Delta J \leq 0$ is satisfied if and only if

$$\|e\| > (f_M + d_M)/(1 - \kappa_{bv\,max}) \quad (9)$$

When this condition is satisfied, the first difference of Lyapunov function candidate is negative for any time instance k. Hence, the closed-loop system is globally uniformly bounded.

Remark 4: The above theorem using Lyapunov method shows that under the general case of where errors in traffic estimation is upped bounded and with bounded disturbances, the control scheme will ensure that the actual queue level converges close to the target value.

Next the outgoing traffic function is estimated, using a vector of traffic parametes θ, by, $f_i(u_{i+1}(k)) = \theta_i f_i(k-1) + \varepsilon(k)$ where $f_i(k-1)$ is the past value of the outgoing traffic, and the approximation error $\varepsilon(k)$ is assumed bounded by known constant ε_M. Now, define traffic estimate in the controller as $\hat{f}_i(u_{i+1}(k)) = \hat{\theta}_i(k) f_i(k-1)$, where $\hat{\theta}_i(k)$ is the actual vector of traffic parameters, $\hat{f}_i(u_{i+1}(k))$ is an estimate of the unknown outgoing traffic $f_i(u_{i+1}(k))$, and $f_i(k-1)$ is the past value of the outgoing traffic.

Theorem 3 (Ideal case of no traffic estimation error): Given the incoming rate selection scheme above with variable θ_i estimated accurately (no estimation error), and if the back-off interval is updated as (5), then the mean estimation error of the variable θ_i along with the mean error in queue utilization converges to zero asymptotically, if the parameter θ_i is updated as

$$\hat{\theta}_i(k+1) = \hat{\theta}_i(k) + \lambda u_i(k) e_{fi}(k+1) \quad (10)$$

provided: (a) $\lambda \|u_i(k)\|^2 < 1$ and (b) $\kappa_{fv\,max} < 1/\sqrt{\delta}$, where $\delta = 1/\left[1 - \lambda \|u_i(k)\|^2\right]$, $\kappa_{fv\,max}$ is the maximum singular value of κ_{fv}, λ is the adaptation gain, and $e_{fi}(k) = f_i(k) - \hat{f}_i(k)$ is the error between the estimated value and the actual one.

The rate selected by the above algorithm in equation (5) does not take into account the fading channels whereas it only detects the onset of congestion by monitoring the buffer occupancy. Under the fading wireless channels, the transmitted packets will not be decoded and dropped at the receiver thereby requiring retransmissions. To mitigate congestion due to channel fading at a given node, the rate from (5) has to be reduced when the transmission power calculated by the DPC scheme exceeds the maximum threshold. This is accomplished by using an adaptive scheme for selecting virtual rates and back-off intervals for a given node although the back-off intervals of all neighboring nodes are normally unknown.

B. Back-off interval selection

Since multiple nodes in a wireless sensor network compete to access the shared channel, back-off interval selection for nodes plays a critical role in deciding which node gains access to the channel. Thus, the proposed rate selection is implemented by suitably modifying the back-off intervals of the nodes around the congested node to achieve the desired rate control. For a given node, a relationship between transmission rate and back-off interval exists which depends upon the back-off intervals of all nodes within a sensing range of a transmitting node in the case of CSMA/CA paradigm. To know the back-off intervals of all the neighbors is not feasible due to a large traffic overhead resulting from communication.

Therefore, we propose using a distributed and predictive algorithm to estimate back-off intervals, such that a target rate is achieved. The main goal is to select back-off interval, BO_i, at the i^{th} transmitting node such that the actual throughput meets the desired outgoing rate $f_i(k)$. To simplify calculations, we consider the inverse of the back-off interval, which is denoted as $VR_i = 1/BO_i$, where VR_i is the virtual rate at i^{th} node, and BO_i is the corresponding back-off interval. The fair scheduling algorithm, discussed in the next subsection, schedules the packet transmissions according to the calculated node's back-off interval. The interval is counted-down when a node does not detect any transmission, and pauses otherwise. Consequently, a node will gain access to the channel proportional to its virtual rate and inversely proportional to the sum of virtual rates of its neighbors. The actual rate of the i^{th} node is a fraction of the channel bandwidth, B(t), defined as

$$R_i(t) = \frac{B(t) \cdot VR_i(t)}{\sum_{l \in S_i} VR_l(t)} = \frac{B(t) \cdot VR_i(t)}{TVR_i(t)} \tag{11}$$

where $TVR_i(t)$ is the sum of all virtual rates for all neighbor nodes, S_i.

Since the scheme considers only a single modulation scheme, bandwidth, B, is assumed time-invariant until the back-off interval is selected. However, when the severe fading occurs, the bandwidth will drop to zero. In such a case, back-off intervals are set at a large value, lar, to prevent unnecessary transmissions when a suitable signal to noise ratio (SNR) cannot be achieved at a destination node due to power constraints. Additionally, under normal circumstances, the algorithm presented below is used to calculate the back-off interval BO_i, which is then randomized in order to minimize probability of collision on access between nodes. Consequently, the MAC layer back-off timer BT_i value is defined as

$$BT_i = \begin{cases} \rho \cdot BO_i(k), & for\ B(k) = 1 \\ lar, & for\ B(k) = 0 \end{cases} \tag{12}$$

where ρ is a random variable with mean one, lar is a large value of the back-off interval and $B(k)$ is the variable that is used to identify whether there is an onset of channel fading or not.

Equation (11) represents the relationship between the back-off intervals and the outgoing flow rate. In order to design a controller which will track the target rate, the system equation is differentiated and then transformed into discrete-time domain. This allows the design of a feedback controller for the selection of the appropriate back-off interval.

Theorems 4 and 5 present that the proposed back-off selection scheme ensures convergence of traffic and stability in the sense of Lyapunov [10][11] in both cases: *ideal* where the throughput dynamics are known and *general* where the dynamics are estimated by an adaptive scheme. In the latter case, the estimation error is bounded by known value ε_N.

1) Adaptive back-off interval selection: Differentiating (11) to get

$$\dot{R_i}(t) = \frac{B}{TVR_i^2(t)} \left[V\dot{R_i}(t) TVR_i(t) - VR_i(t) TV\dot{R_i}(t) \right] \tag{13}$$

To transform the differential equation into the discrete-time domain, Euler's formula is used as

$$\begin{aligned} R_i(k+1) - R_i(k) = \frac{B}{TVR_i^2(k)} & \left[(VR_i(k+1) - VR_i(k)) \right. \\ & \left. TVR_i(t) - VR_i(t)(TVR_i(k+1) - TVR_i(k)) \right] \end{aligned} \tag{14}$$

After applying (11) we can transform (14) to get

$$R_i(k+1) = \frac{R_i(k) VR_i(k+1)}{VR_i(k)} + R_i(k) \left(1 - \frac{TVR_i(k+1)}{TVR_i(k)} \right) \tag{15}$$

Now, define $\alpha_i(k) = 1 - TVR_i(k+1)/TRV_i(k)$, $\beta_i(k) = R_i(k)/VR_i(k)$, and $v_i(k) = VR_i(k+1) = 1/BO_i(k+1)$. The variable $\alpha_i(k)$ describes a variation of back-off intervals of flows at the neighboring nodes from the time instant k to $k+1$. This variation is caused due to congestion resulting from traffic and fading channels. Since this information is not available locally, it is considered an unknown parameter, and thus estimated by the algorithm. The parameter $\beta_i(k)$ is the ratio between actual and the used virtual rate at time instant k, and can be easily calculated. The term $v_i(k)$ is the back off interval that needs to be calculated for each node.

Now, (15) can be written as

$$R_i(k+1) = R_i(k)\alpha_i(k) + \beta_i(k) v_i(k) \tag{16}$$

Equation (16) indicates that the achieved rate at the instant, $k+1$, depends on the variations of back-off intervals in the neighboring nodes. Now, select the back-off interval as

$$v_i(k) = \frac{f_i(k) - R_i(k)\hat{\alpha}_i(k) - \kappa_v e_i(k)}{\beta_i(k)} \tag{17}$$

where $\hat{\alpha}_i(k)$ is estimate of $\alpha_i(k)$, $e_i(k) = R_i(k) - f_i(k)$ is defined as throughput error, and κ_v is the feedback gain parameter. In this case, the throughput error is expressed as

$$\begin{aligned} e_i(k+1) &= \kappa_v e_i(k) + \alpha_i(k) R_i(k) - \hat{\alpha}_i(k) R_i(k) \\ &= \kappa_v e_i(k) + \tilde{\alpha}_i(k) R_i(k) \end{aligned} \tag{18}$$

where $\tilde{\alpha}_i(k) = \alpha_i(k) - \hat{\alpha}_i(k)$ is the error in estimation.

The throughput error of the closed-loop system for a given link is driven by the error in back-off intervals of the neighbors which are typically unknown. If these uncertainties are properly estimated a suitable back-off interval is selected for the node under consideration such that a suitable rate is selected to mitigate potential congestion. If the error in uncertainties tends to zero, equation (18) reduces to $e_i(k+1) = \kappa_v e_i(k)$. In the presence of back-off interval variations of neighboring nodes, the congestion control scheme will ensure that the actual throughput is close to its target value but it will not guarantee convergence of actual back-off interval to its ideal target for all the nodes. Unless suitable back-off intervals are selected for all the nodes, congestion cannot be prevented.

Theorem 4 (Back-off selection algorithm under ideal circumstances): Given the back-off selection scheme above with variable $\alpha_i(k)$ estimated accurately (no estimation error), and the back-off interval updated as (17), then the mean estimation error of the variable $\alpha_i(k)$ along with the mean error in throughput converges to zero asymptotically, if the parameter $\alpha_i(k)$ is updated as

$$\hat{\alpha}_i(k+1) = \hat{\alpha}_i(k) + \sigma R_i(k) e_i(k+1) \tag{19}$$

provided

$$\text{(a) } \sigma \|R_i(k)\|^2 < 1 \text{ and (b) } \kappa_v max < 1/\sqrt{\delta} \tag{20}$$

where $\delta = 1/\left(1-\sigma\|R_i(k)\|^2\right)$, $\kappa_{vmax}$is the maximum singular value of κ_v, and σ is the adaptation gain.

Proof: Define the Lyapunov function candidate

$$J = e_i^2(k) + \frac{1}{\sigma}\tilde{\alpha}_i^2(k) \tag{21}$$

whose first difference is

$$\begin{aligned}\Delta J &= \Delta J_1 + \Delta J_2 \\ &= e_i^2(k+1) - e_i^2(k) + \frac{1}{\sigma}\left(\tilde{\alpha}_i^2(k+1) - \tilde{\alpha}_i^2(k)\right)\end{aligned} \tag{22}$$

Consider from (22) and substituting (18) to get

$$\begin{aligned}\Delta J_1 &= e_i^2(k+1) - e_i^2(k) \\ &= \left[\kappa_v e_i(k) + \tilde{\alpha}_i(k)R_i(k)\right]^2 - e_i^2(k)\end{aligned} \tag{23}$$

Taking the second term of the first difference from (22) and substituting (19) yields

$$\begin{aligned}\Delta J_2 = \frac{\tilde{\alpha}_i^2(k+1) - \tilde{\alpha}_i^2(k)}{\sigma} = -2\kappa_v e_i(k)\tilde{\alpha}_i(k)R_i(k) \\ -2\left[\tilde{\alpha}_i(k)R_i(k)\right]^2 + \sigma R_i^2(k)\left[\kappa_v e_i(k) + \tilde{\alpha}_i(k)R_i(k)\right]^2\end{aligned} \tag{24}$$

Combining (23) and (24) to get

$$\begin{aligned}\Delta J = &-\left(1-\sigma R_i^2(k)\right)\left[\tilde{\alpha}_i(k)R_i(k)\right]^2 \\ &+ 2\sigma R_i^2(k)\kappa_v e_i(k)\left[\tilde{\alpha}_i(k)R_i(k)\right] - \sigma R_i^2(k)\kappa_v^2 e_i^2(k) \\ \le &-\left(1-\delta\kappa_{v\,max}^2\right)\|e_i(k)\|^2 - \left(1-\sigma\|R_i(k)\|^2\right) \\ &\left\|\tilde{\alpha}_i(k)R_i(k) - \frac{\sigma\|R_i(k)\|^2\kappa_v e_i(k)}{1-\sigma\|R_i(k)\|^2}\right\|^2\end{aligned} \tag{25}$$

where δ is given after (20). Taking now expectations on both sides yields

$$\begin{aligned}E(\Delta J) \le &-E\left\{\left(1-\delta k_{v\,max}^2\right)\|e_i(k)\|^2\right. \\ &-\left(1-\sigma\|R_i(k)\|^2\right) \\ &\left.\left\|\tilde{\alpha}_i(k)R_i(k) + \frac{\sigma\|R_i(k)\|^2}{1-\sigma\|R_i(k)\|^2}\kappa_v e_i(k)\right\|^2\right\}\end{aligned} \tag{26}$$

Since $E(J) > 0$ and $E(\Delta J) \le 0$, this shows the stability in the mean, in the sense of Lyapunov [10][11] provided the conditions (20) and (20) hold, so $E[e_i(k)]$ and $E[\tilde{\alpha}_i(k)]$ (and hence $E[\hat{\alpha}_i(k)]$) are bounded in the mean if $E[e_i(k_0)]$ and $E[\tilde{\alpha}_i(k_0)]$ are bounded. Sum both sides of (26) and taking limits $lim_{k\to\infty}E(\Delta J)$, the error $E\left[\|e_i(k)\|\right] \to 0$.

Consider the closed loop throughput error with estimation error, $\varepsilon(k)$, as

$$e_i(k+1) = \kappa_v e_i(k) + \alpha_i(k)R_i(k) + \varepsilon(k) \tag{27}$$

Theorem 5 (Back-off selection algorithm in general case): Assume the hypothesis as given in Theorem 4, and let the uncertain parameter α_i be estimated using (18) with $\varepsilon(k)$ the error in estimation which is considered bounded above such that $\|\varepsilon(k)\| \le \varepsilon_N$, where ε_N is a known constant. Then the mean error in throughput and the estimated parameters are bounded provided (20) and (20) hold.

Proof: Define a Lyapunov function candidate as in (21) whose first difference is given by (22). The first term ΔJ_1 and ΔJ_2 the second term can be obtained respectively as

$$\begin{aligned}\Delta J_1 =& e_i^2(k)\kappa_v + 2\kappa_v e_i(k)\tilde{\alpha}_i(k)R_i(k) + \left[\tilde{\alpha}_i(k)R_i(k)\right]^2 \\ &+ \varepsilon^2(k) + 2\kappa_v e_i(k)\varepsilon(k) + 2\varepsilon(k)e_i(k) - e_i^2(k)\end{aligned} \tag{28}$$

$$\begin{aligned}\Delta J_2 =& -2\kappa_v e_i(k)\tilde{\alpha}_i(k)R_i(k) - 2\left[\tilde{\alpha}_i(k)R_i(k)\right]^2 \\ &+ \sigma R_i^2(k)\left[\kappa_v e_i(k) + \tilde{\alpha}_i(k)R_i(k)\right]^2 \\ &- 2\left[1-\sigma R_i^2(k)\right]e_i(k)\varepsilon(k) \\ &+ 2\sigma R_i^2(k)\kappa_v e_i(k)\varepsilon(k) + \sigma R_i^2(k)\varepsilon^2(k)\end{aligned} \tag{29}$$

Following (25) and completing the squares for $\tilde{\alpha}_i(k)R_i(k)$ yields

$$\begin{aligned}\Delta J \le& -(1-\delta\kappa_{v\,max}^2)\left[\|e_i(k)\|^2 - \frac{\delta\varepsilon_N^2}{1-\delta\kappa_{v\,max}^2}\right. \\ &\left.-\varepsilon_N\|e_i(k)\|\frac{2\sigma\kappa_{v\,max}\|R_i(k)\|^2}{1-\delta\kappa_{v\,max}^2}\right] - \left(1-\sigma\|R_i(k)\|^2\right) \\ &\cdot\left\|\tilde{\alpha}_i(k)R_i(k) - \frac{\sigma\|R_i(k)\|^2}{1-\sigma\|R_i(k)\|^2}\left(\kappa_v e_i(k) + \varepsilon(k)\right)\right\|^2\end{aligned} \tag{30}$$

with δ is given after (20). Taking expectations on both sides to get

$$\begin{aligned}&E(\Delta J) \le \\ &-E\left\{(1-\delta\kappa_{v\,max}^2)\left[\|e_i(k)\|^2 - \frac{\delta\varepsilon_N^2}{1-\delta\kappa_{v\,max}}\right.\right. \\ &\left.-\varepsilon_N\|e_i(k)\|\frac{2\sigma\kappa_{v\,max}\|R_i(k)\|^2}{1-\delta\kappa_{v\,max}^2}\right] - \left(1-\sigma\|R_i(k)\|^2\right) \\ &\left.\left\|\tilde{\alpha}_i(k)R_i(k) - \frac{\sigma\|R_i(k)\|^2}{1-\sigma\|R_i(k)\|^2}\left(\kappa_v e_i(k) + \varepsilon(k)\right)\right\|^2\right\}\end{aligned} \tag{31}$$

as long as (20) and (20) hold, and $E\left[\|e_i(k)\|\right] > \dfrac{\varepsilon_N(\sigma\kappa_{v\,max} + \sqrt{\sigma})}{1-\sigma\kappa_{v\,max}^2}$. This demonstrates that $E(\Delta J)$ is negative outside a compact set U. According to a standard Lyapunov extension [10][11], the throughput error $E[e_i(k)]$ is bounded for all $k \ge 0$. It is required to show that $\hat{\alpha}_i(k)$ or equivalently $\tilde{\alpha}_i(k)$ is bounded. The dynamics in error in the parameters estimates are

$$\tilde{\alpha}_i(k+1) = \left[1-\sigma R_i^2(k)\right]\tilde{\alpha}_i(k) - \sigma R_i(k)\left[\kappa_v e_i(k) + \varepsilon(k)\right] \tag{32}$$

where the error, $e_i(k)$, is bounded and estimation error, $\varepsilon(k)$, is bounded. Applying the persistency of excitation condition [1], one can show that $\tilde{\alpha}_i(k)$ is bounded.

2) Rate propagation: This total incoming rate is then divided among the upstream nodes proportionally to the sum of flow weights passing through a given node j as $u_{ij}(k) = u_i(k)\dfrac{\sum_n^{\text{flows at } j^{th} \text{ node}}\varphi_n}{\sum_m^{\text{flows at } i^{th} \text{ node}}\varphi_m}$, where $u_{ij}(k)$ is the rate allocated for a transmitting node j at receiving node i, $u_i(k)$ is the rate selected for all incoming flows at i^{th} node and given by

(3), and φ_n, φ_m are pre-assigned weights of the n^{th} and m^{th} flows respectively. Next, the selected rate $u_{ij}(k)$ is communicated to the upstream node j to *mitigate congestion*. This feedback continues recursively to the nodes upstream from the congested link so that they will also reduce transmission rates and thus prevent overflowing buffers. One can update the pre-assigned weights to guarantee weighted fairness and to improve throughput as discussed next.

C. Fair scheduling

Data packets at a receiver are first scheduled using the adaptive dynamic fair scheduling (ADFS) scheme [6]. Weights that correspond to the packets flows are used to build a schedule for transmission. This algorithm ensures weighted fairness defined in (1) among the flows passing a given node. The proposed scheme offers an additional feature of dynamic weight adaptation that further boosts the fairness and guarantees performance analytically as presented in this paper.

This ADFS feature increases throughput while ensuring fairness of the flows by adjusting per-packet weight during congestion. This feature, though utilized here, can be optional in the congestion control scheme since it introduces additional overhead, although shown to be low [6], in the form of: a) extra bits in each packet to carry the weight, and b) additional calculations performed to evaluate fairness and update the packet weight at each hop. However, this algorithm is necessary in a dynamic environment.

Dynamic weight adaptation given in (33) is utilized in ADFS scheme [6]. The initial weights are selected by using the user-defined QoS criteria. Then, the packet weights are dynamically adapted with network state defined, as a function of delay experienced, number of packets in the queue and the previous weight of the packet. In fact, analytical results are included in [6] to demonstrate the throughput and end-to-end delay bounds in contrast with the existing literature. The weights are updated as follows.

1) Dynamic weight adaptation: To account for the changing traffic and channel conditions that affect the fairness and end-to-end delay, the weights for the flows are updated dynamically as

$$\hat{\varphi}_{ij}(k+1) = \xi\hat{\varphi}_{ij}(k) + \varsigma E_{ij} \qquad (33)$$

where $\hat{\varphi}_{ij}(k)$ is the actual weight for the i^{th} flow, j^{th} packet at time k, ξ and ς are design constants, $\{\xi, \varsigma\} \in [-1, 1]$, and E_{ij} is defined as $E_{ij} = e_{bi} + 1/e_{ij,delay}$, where e_{bi} is the error between the expected length of the queue and the actual size of the queue, and $e_{ij,delay}$ is the error between the expected delay and the delay experienced by the packet so far. According to E_{ij}, when queues buildup or delay increases, the packet weights will be increased to clear the backlog and send the packet sooner. Note that the value of E_{ij} is bounded due to finite queue length and delay, as packets experiencing delay greater than the delay error limit will be dropped. The updated weights are utilized to schedule packets for subsequent transmission.

2) Fairness and throughput guarantee: To prove that the dynamic weight adaptation is fair, we need to show a bound on $\left|\frac{W_f(t_1,t_2)}{\varphi_f} - \frac{W_m(t_1,t_2)}{\varphi_m}\right|$ for a sufficiently long interval $[t_1, t_2]$ in which both flows, f and m, are backlogged. Next, several theorems (not shown) can guarantee proportional fairness, minimal throughput and finally guaranteeing overall performance of the proposed scheme. The proofs closely follow the ones given by authors in [6].

Remark 5: In fact, the weight update (13) ensures that the actual weight assigned to a packet at each node converges close to its target value.

Remark 6: φ_{ij} is finite for each flow at a given node.

Remark 7: Notice that fairness holds regardless of the service rate of the cluster head. This demonstrates that algorithm achieves fair allocation of bandwidth and thus meets a fundamental requirement of fair scheduling algorithm for integrated services networks.

IV. Simulation Results

First, the performance of the proposed scheme in case of variations in outgoing flow rate was assessed using MATLAB. Next, the performance of the DPCC is analyzed in Ns2 simulator using a tree topology, which is typical for a sensor network, with clusters at leaf nodes generating traffic that is sent to the base station at the tree's root. This scenario allows observing the performance improvement of the congestion control algorithm over the DPC scheme [1] alone. Finally, the proposed scheme is compared with CODA scheme in the unbalanced tree topology where one source in the tree topology is moved closer to the base station thus giving this source advantage over others. The CODA scheme has been implemented in Ns2 by carefully following the description in [3]. Next the simulation results are discussed.

A. Performance in case of outgoing flow variation

The MATLAB simulations are used to evaluate performance of the controller proposed in section III.A. The outgoing flow rate variations can be viewed as MAC data rate changes, thus providing indication how the proposed protocol performs in networks that support multiple modulations rates. In this simulation, the maximum and ideal queue size is set to 20 and 10 packets respectively, the controller parameters are $\kappa_{vb} = 0.1$ and $\lambda = 0.001$.

Fig. 4 illustrates the actual and estimated value of the outgoing flow, together with the queue utilization, and Fig. 5 presents the error in estimation of outgoing flow, e_f, and error of queue utilization, e_{bi}. The errors are bounded and quickly converge to zero since the scheme adapts to the changed outgoing flow rate and is able to track the actual value, f_{out}. The errors occur when the sudden change in the outgoing flow occurs, since the outgoing traffic estimation could not predict such abrupt changes. However, in just a few next iterations the algorithm converged to the ideal state since the estimation scheme quickly detected and accommodated the changed bandwidth.

B. Tree topologies results

The Ns2 simulations were setup to use 2Mbps channel with path loss, shadowing and Rayleigh fading with AODV routing

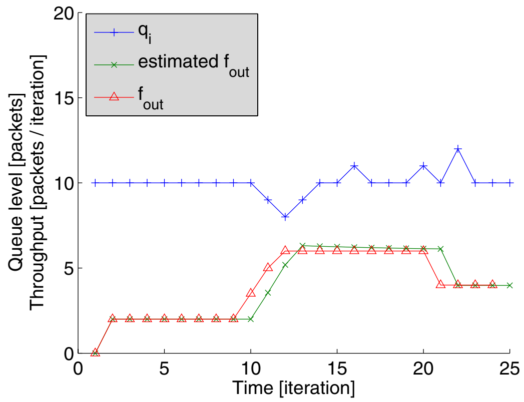

Fig. 4. Queue utilization and estimation of the outgoing flow.

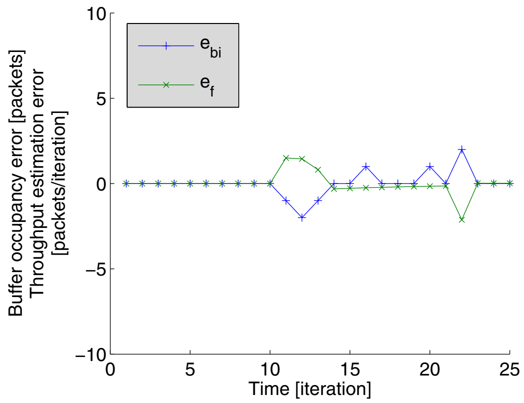

Fig. 5. Queue utilization error and outgoing traffic estimation error.

TABLE I

DELAY, THROUGHPUT AND ENERGY-EFFICIENCY

Protocol	Average delay [s]	Fairness Index (FI) (mix weights)	Network efficiency [kbps]	Energy efficiency [kbps/joule]
Proposed	0.8	1.00	400.99	13.05
802.11	–	0.91	77.86	3.23
Proposed	1.06	0.91	368.55	11.79

protocol at the cluster head level. The queue limit is set to 50 with the packet size taken as 512 bytes. In the tree topology, traffic accumulates near the destination node thus causing congestion at the intermediate nodes. All the sources generate traffic proportional to their weights that exceeds channel capacity so that congestion can be created. The calculations of the rate and back-off intervals are performed periodically for 0.5 second intervals.

The results for end-to-end delay do not include the IEEE 802.11 scheme since the protocol is quickly staled due to congestion and only very few packets are received at the destination. Consequently, the observed delay cannot be compared with the other protocols. The DPC protocol improves the channel utilization in presence of collisions, as described in [1]. However, the imbalance between incoming and outgoing flows due to congestion is not addressed by the DPC thus still resulting in buffer overflows and a significant drop rate.

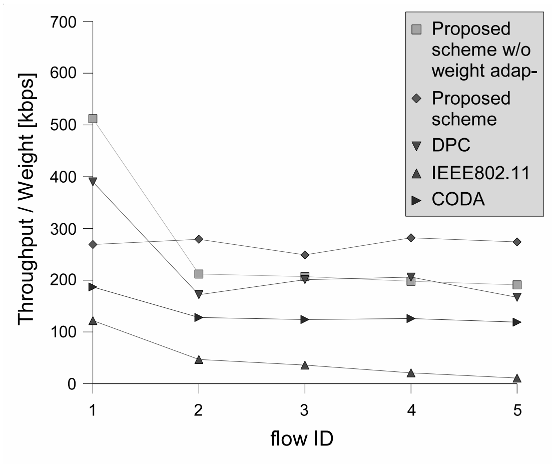

Fig. 6. Performance for unbalanced tree topology.

1) Balanced tree topology with weight variation: In the simulations, the traffic consists of five flows, which had been simulated with weights equal to 0.4, 0.1, 0.2, 0.2, and 0.1 respectively. Table 1 summarizes the overall performance of the protocols. The Fairness Index (FI) in Table 1 shows the fairness in case of varying flow weights. An ideal protocol will have the FI equal to 1.00. Both 802.11 and DPC have FI smaller than one indicating unfair handling of flows, while the proposed scheme achieves fairness index equal to one indicating fair allocation of resources to the flows. The proposed DPCC protocol achieves an end-to-end fairness by recursively applying the proposed scheme at every node, which in turn, guarantees the hop-by-hop fairness.

2) Unbalanced tree topology results: For the unbalanced tree topology, the flow number one is located closer to the destination than other sources. Consequently, without adaptive weights, the first flow is favored at the expense of the others. Fig. 6 depicts the '*throughput/weight*'(normalized weights) ratio, and Fig. 7 presents '*weight*delay*' metric for all flows and protocols. The DPCC protocol with weight adaptation outperforms other schemes, since besides alleviating congestions it also identifies that the flows 2 through 5 are hindered due to congestion and network topology. Consequently, the DPCC adjusts their weights at next hops to meet fairness criteria. As a result, at the destination, all the flows achieve the same weighted throughput and end-to-end delay.

In comparison, the CODA scheme improves the performance of the network over the 802.11 protocol since it restricts network traffic during the congestion by using the backpressure mechanism. However, CODA is not able to achieve throughput comparable to the proposed DPCC protocol since CODA uses a binary bit to identify the onset of congestion and has no precise control over the incoming flows. In contrast, the DPCC mitigates onset of congestion by precisely controlling queue utilization, thus completely preventing buffer overflows. Moreover, the end-to-end delay for CODA scheme increases since during congestion the node's transmission is halted for a random period of time as stated by the congestion policy [3]. Overall, the proposed protocol improves the performance of the network by 93-98% when compared with CODA scheme, thus justifying increased processing requirements of the proposed scheme.

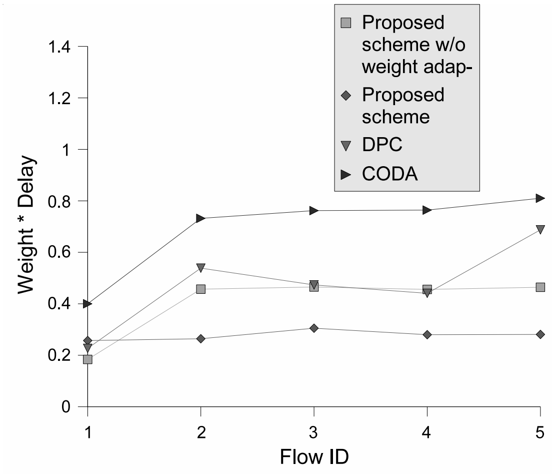

Fig. 7. Weighted delay with equal flow weights (const=0.2).

V. Conclusions

This paper presents a novel predictive congestion control scheme whereby the congestion is mitigated by suitably predicting the back-off interval of all the nodes based on the current network conditions. The network conditions include the traffic flow through a given region and channel state. Simulation and experimental results show that the proposed scheme increases throughput, network efficiency and energy conservation. With the addition of a fair scheduling algorithm, the scheme guarantees desired quality of service (QoS) and weighted fairness for all flows even during congestion and fading channels. Finally, the proposed scheme provides a hop by hop mechanism for throttling packet flow rate, which will help in mitigating congestion. The convergence analysis is demonstrated by using a Lyapunov-based analysis. Extensive simulation results are included to verify the performance. Future work will include evaluation of the proposed scheme in a realistic wireless test-bed and comparison with other implemented congestion control schemes and it will be submitted as part of future publication.

References

[1] M. Zawodniok and S. Jagannathan, “A distributed power control MAC protocol for wireless ad hoc networks,” in *Proc. IEEE WCNC’04*, vol. 3, pp. 1915–1920, March 2004.

[2] B. Hull, K. Jamieson, and H. Balakrishnan, “Mitigating congestion in wireless sensor networks,” in *Proc. ACM SenSys 2004*, pp. 134–147, Nov. 2004.

[3] C.-Y. Wan, S. Eisenman, and A. Campbell, “CODA: congestion detection and avoidance in sensor networks,” in *Proc ACM SenSys’03*, pp. 266–279, Nov. 2003.

[4] Y. Yi and S. Shakkottai, “Hop-by-hop congestion control over wireless multi-hop networks,” in *Proc. IEEE INFOCOM ’04*, vol. 4, pp. 2548–2558, March 2004.

[5] S. Jagannathan, “End to end congestion control in high-speed networks,” in *Proc. IEEE LCN*, pp. 547–556, 2002.

[6] N. Regatte and S. Jagannathan, “Adaptive and distributed fair scheduling scheme for wireless ad hoc networks,” in *Proc. World Wireless Congress*, pp. 101–106, May 2004.

[7] N. H. Vaidya, P. Bahl, and S. Gupta, “Distributed fair scheduling in a wireless LAN,” in *Proc. 6th Annual International Conference on Mobile Computing & Networking*, pp. 167–178, Aug. 2000.

[8] S.-C., Wang, Y.-M. Chen, T. H. Lee, and A. Helmy, “Performance evaluations for hybrid IEEE 802.11b and 802.11g wireless networks,” in *Proc. 24th IEEE IPCCC 2005*, pp. 111–118, April 2005.

[9] W.-K. Kuo and C.-C. J. Kuo, “Enhanced backoff scheme in CSMA/CA for IEEE 802.11,” in *Proc. 58th IEEE Vehicular Technology Conference (VTC Fall 2003)*, vol. 5, pp. 2809–2813, Oct. 2003.

[10] K. J. Astrom and B. Wittenmark, *Adaptive Control*. Reading, MA: Addison-Wesley, 1989.

[11] S. Jagannathan and F. L.Lewis, “Robust implicit self tuning regulator/MRAC convergence and stability,” *Automatica*, vol. 32, Dec. 1996.

[12] S. Rangwala, R. Gummadi, R Govindan, and K. Psounis, “Interference-aware fair rate control in wireless sensor networks,” in *Proc. ACM SIGCOMM’06*, pp. 63–74, Sept. 2006.

[13] C. Wang, K. Sohraby, and B. Li, “Sentcp: a hop-by-hop congestion control protocol for wireless sensor networks,” in *Proc. IEEE INFOCOM 2005*, Mar. 2005.

[14] C.-T. Ee and R. Bajcsy, “Congestion control and fairness for many-to-one routing in sensor networks,” in *Proc. ACM Sensys ’04*, pp. 148–161, Nov. 2004.

[15] Chonggang Wang, Sohraby, K. Lawrence, V., Bo Li,Yueming Hu, *Priority-based congestion control in wireless sensor networks,* IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing 2006, vol. 1, pp. 22- 31, June 2006.

Maciej Zawodniok was born in Knurow, Poland. Mr. Zawodniok graduated from the Silesian University of Technology with a Master of Science degree in Computer Science in 1999. He received a Ph.D. degree in Computer Engineering from the University of Missouri-Rolla in 2006. During 1999 to 2002, he worked for Ericsson AB as a consultant, where he delivered courses about cellular telecommunication for Ericsson company and its customers around the world. Then, during 2002 to 2003 he worked as a Software Engineer in Motorola Polska. There, he was responsible for preparation and implementation of a novel automatic testing of cellular network’s equipment. During 2003-2006, Mr. Zawodniok was a Research Assistant at University of Missouri-Rolla, and a Post Doctoral Fellow from 2006.

Mr. Zawodniok’s research focuses on developing adaptive and energy-efficient protocols for wireless networks, network-centric systems and network security. Mr. Zawodniok is also involved in interdisciplinary projects related to application of RFID technology and wireless sensor networks in manufacturing and maintenance. He is a student member of IEEE.

S. Jagannathan was born in Madurai, India and he received the Bachelor’s degree in Electrical Engineering from College of Engineering, Guindy at Anna University, Madras India in 1987, the Master of Science Degree in Electrical Engineering from the University of Saskatchewan, Saskatoon, Canada in 1989, and the Ph.D. degree in Electrical Engineering from the University of Texas in 1994. During 1986 to 1987, he was a junior engineer at Engineers India Limited, New Delhi, as a Research Associate and Instructor from 1990 to 1991, at the University of Manitoba, Winnipeg Canada, and worked at Systems and Controls Research Division, in Caterpillar Inc., Peoria as a consultant during 1994 to 1998. During 1998 to 2001 he was at the University of Texas at San Antonio, and since September 2001, he is at the University of Missouri-Rolla where he is currently a professor. He has coauthored more than 170 refereed conference and juried journal articles and several book chapters and one text book entitled *Neural Network Control of Robot Manipulators and Nonlinear Systems*, published by Taylor & Francis, London in 1999, two more books in process entitled *Discrete-Time Neural Network Control of Industrial Nonlinear Systems* and *Quality of Service Control in High Speed and Wireless Networks.* His research interests include adaptive and neural network control, MEMS and embedded sensor networks, computer/communication/sensor networks, prognostics, and autonomous systems/robotics.

Dr. Jagannathan received several gold medals and scholarships. He was the recipient of Teaching Commendation award in 2005, Presidential Award for Research Excellence at UTSA in 2001, NSF CAREER award in 2000, Faculty Research Award in 2000, Patent Award in 1996, and Sigma Xi Doctoral Research Award in 1994. He currently holds 17 patents and several are in process. He has served and currently serving on the program committees of several IEEE conferences. He is currently serving as the Associate Editor for the *IEEE Transactions on Control Systems Technology* and *IEEE Transactions on Neural Networks*. He is a member of Tau Beta Pi, Eta Kappa Nu, and Sigma Xi. He is a Senior Member of the IEEE.