WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. (2008)
DOI: 10.1002/wcm.705

E^2SRT: enhanced event-to-sink reliable transport for wireless sensor networks

Sunil Kumar*,†, Zhenhua Feng, Fei Hu and Yang Xiao

Electrical and Computer Engineering Department, San Diego State University, San Diego, CA 92182–1309, U.S.A.

Summary

An event-to-sink reliable transport (ESRT) control scheme was recently proposed to address the event-to-sink reliability issues in wireless sensor network (WSN). In this paper, we study the performance of ESRT in the presence of 'over-demanding' event reliability, using both the analytical and simulation approaches. We show that the ESRT protocol does not achieve optimum reliability and begins to fluctuate between two inefficient network states. With insights from update mechanism in ESRT, we propose a new algorithm, called enhanced ESRT (E^2SRT), to solve the 'over-demanding' event reliability problem and to stabilize the network. Simulation results show that E^2SRT outperforms ESRT in terms of both reliability and energy consumption in the presence of 'over-demanding' event reliability. Besides, it ensures robust convergence in the presence of dynamic network environments.

KEY WORDS: wireless sensor networks; reliable transport protocol; event-to-sink reliability; congestion control; ESRT

1. Introduction

Wireless sensor networks (WSNs) are usually deployed to monitor a set of events, such as structural defects [1], habitat [2,3], and surveillance [4]. The main task of sensors in a WSN is to sense the target event(s) and forward the relevant event information to the sink. The sink, installed with more powerful software and hardware, analyzes the event information and takes appropriate action(s). Sometimes, the sink may also forward the information to upper-level host(s).

This event-based characteristic brings a new perspective on transport reliability control in WSN. The traditional TCP (transmission control protocol) [5] based transport protocols, which consider the conventional end-to-end reliability, have the following limitations when used in WSN. *First*, they do not address the energy conservation needs of sensors. *Second*, the reliability in WSN depends on the collective event information received by the sink, and not necessarily on individual packets from any particular sensor node. To address the unique requirements of WSN, there is a need to design an event-to-sink reliability oriented transport protocol which computes the rreliability associated with specific events in a collective manner rather than end-to-end manner.

The event-to-sink reliable transport (ESRT) scheme proposed by Akan and Ian [6] was the first scheme to study the transport reliability issues from this perspective. It aimed at reliable event detection in WSN with reduced energy consumption. ESRT scheme has following features: (i) it introduces a congestion control

---

*Correspondence to: Sunil Kumar, Electrical and Computer Engineering Department, San Diego State University, San Diego, CA 92182-1309, U.S.A.
†E-mail: skumar@mail.sdsu.edu

algorithm that enforces reliability while conserving energy; (ii) most operations are done at the sink; (iii) it is an adaptive algorithm that converges to the optimal operating region (OOR) state in a finite number of iterations. The adaptability makes ESRT robust to potentially random and dynamic changes in WSN.

However, we have observed that ESRT cannot converge to the OOR state when the desired event reliability is sufficiently larger than the maximum capacity of the network. We call this condition as '*over-demanding event reliability*' (OR). In such a case, the ESRT scheme causes the network to fluctuate between two states: 'congestion-low-reliability' (C, LR) and 'no-congestion-low-reliability' (NC, LR). In this paper, we propose a new transport scheme for WSN, called *enhanced ESRT* (E^2SRT), which inherits all the merits of ESRT while eliminating the undesirable fluctuations in OR case. The proposed E^2SRT scheme has the following features:

- *Robust Convergence*: it has no assumption on the 'desired event reliability requirements' at the server/sink side. The user or application can pose any reliability requirement on the network. The network recursively estimates the maximum achievable event reliability and converges to the realistic (but sub-optimal) maximum operating region (MOR).
- *Awareness of Dynamic Environment*: many factors affect the available resources in WSN, such as varying application requirements, node movements, sensor power, etc. The desired event reliability might be achievable for some situations but may be unrealistic in others. The E^2SRT scheme can dynamically accommodate such unpredictable changes and converges to achieve the best performance for a given set of network parameters without causing congestion.

The remainder of this paper is organized as follows. We discuss the major features of ESRT scheme in Section 2, followed by its performance analysis in Section 3. In Section 4, we discuss our proposed E^2SRT scheme, including formal definition of the 'over demanding event reliability' problem and analytical results. We present the simulation results of E^2SRT scheme in Section 5. In Section 6, we discuss the related transport schemes and finally conclude our work in Section 7.

2. Overview of ESRT Scheme

Many micro sensors are typically deployed in WSN, including at least one sink that coordinates the sensors and collects the required event information. The sensors promptly or periodically communicate with the sink to report specific event(s). The success of WSN in many applications is determined by the amount of information reaching the sink during a time period. To address this WSN requirement, the concept of 'event-to-sink reliability' was introduced in ESRT [6], which uses the number of packets received by the sink in certain time duration as a measure of event reliability. We use the following parameters as used in ESRT [6].

- *Observed event reliability* (r_i): number of received data packets in a decision interval i at the sink.
- *Desired event reliability* (R): number of data packets required for reliable event detection as determined by the application.
- *Reporting frequency rate* (f): number of packets sent out per unit time by a sensor node.
- *Normalized reliability* (η_i): denoted by r_i/R at the end of each decision interval i.
- *Protocol parameter* (ε): denotes the width of tolerance zone for the optimal operation region (OOR) state.

The *transport problem in WSN* is to configure the reporting rate of source nodes so as to achieve the required event detection reliability at the sink with minimum resource utilization. The event reliability closely follows a curve shown in Figure 1 when the sensors adjust their reporting frequency f in order to achieve a certain reliability $R \leq R_{\max}$. The observed event reliability r increases almost linearly with f until certain $f = f_{\max}$ is reached. When $f > f_{\max}$, the network becomes congested and packets are dropped leading to decrease in r [6]. Thus there is a maximum achievable reliability $R_{\max}$ for a given WSN set up.

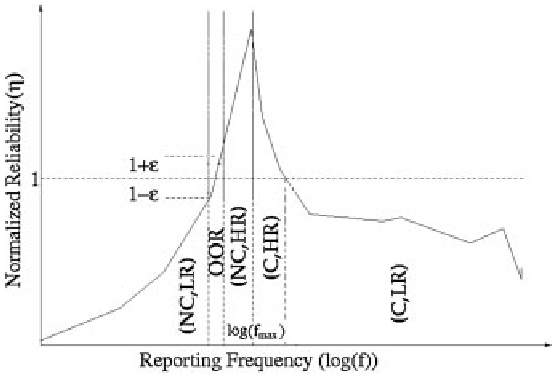


Fig. 1. A representative curve to show the effect of reporting rate of sensor nodes (f) on normalized event reliability, η [6].

DOI: 10.1002/wcm

The ESRT segments the performance curve into five regions, which represent five distinct working states of sensor network [6].

- (NC, LR): $f < f_{\max}$ and $\eta < 1 - \varepsilon$ (no congestion, low reliability);
- (NC, HR): $f \leq f_{\max}$ and $\eta > 1 + \varepsilon$ (no congestion, high reliability);
- (C, HR): $f > f_{\max}$ and $\eta > 1$ (congestion, high reliability);
- (C, LR): $f > f_{\max}$ and $\eta \leq 1$ (congestion, low reliability);
- OOR: $f < f_{\max}$ and $1 - \varepsilon \leq \eta \leq 1 + \varepsilon$ (optimal operating region).

The reliability and energy consumption characteristic of each working region is discussed in detail in Reference [6]. It concludes that the OOR state fulfills the reliability requirement and consumes the least energy. Simulation results presented in Reference [6] showed that the ESRT converges to OOR from any of the four non-OOR states in only a few decision intervals. It thus successfully serves the two fold purpose of fulfilling event reliability and reducing energy consumption.

However, as we discuss in the next section, the ESRT cannot converge to OOR state when the network is in OR condition, i.e., the desired event reliability (R) is higher than what the network can support.

3. ESRT Performance for Over-Demanding Event Reliability (OR) Case

In order to evaluate the ESRT performance in OR case, we have used the same network set up and simulation parameters as in ESRT paper [6]. However, we tried to achieve a higher event reliability (i.e., 4000–4500 packets per 10 s interval when the network can only handle around 3500 packets per 10 s interval).

Our simulation results for the ESRT scheme, shown in Figure 2, reveal that the network cannot converge to the OOR state in OR case. The simulation results also show that the ESRT scheme cannot detect OR situation by itself. In fact, the network either goes to (C, LR) state or operates at a very low frequency in (NC, LR) state as shown in Figure 3, thus wasting most of the bandwidth. As a result, the achieved event reliability is far below the desired reliability ($\eta < 1$). This indicates that the network is trying to achieve a reliability value (r) far beyond its capability, which leads to more congestion, more collisions, lower throughput, and longer delay.

4. Proposed E^2SRT Scheme

The 'over-demanding event reliability' (R_{od}) denotes a situation where the desired event reliability R is sufficiently larger than $R_{\max}$, so that $R_{\max}/R < 1 - \varepsilon$. In this case, we consider that the network is in OR (over-demanding reliability) situation as shown in Figure 3. We first give analytical results (in Section 4.1) to show that ESRT cannot converge to OOR in the OR case. We then discuss in Section 4.2 how our proposed E^2RST scheme overcomes this limitation, including the protocol operation (in Section 4.3) and analytical results (in Section 4.4).

4.1. Analytical Results of State Transitions in ESRT

We use mathematical analysis in this section to show that ESRT cannot converge to OOR state in OR case,

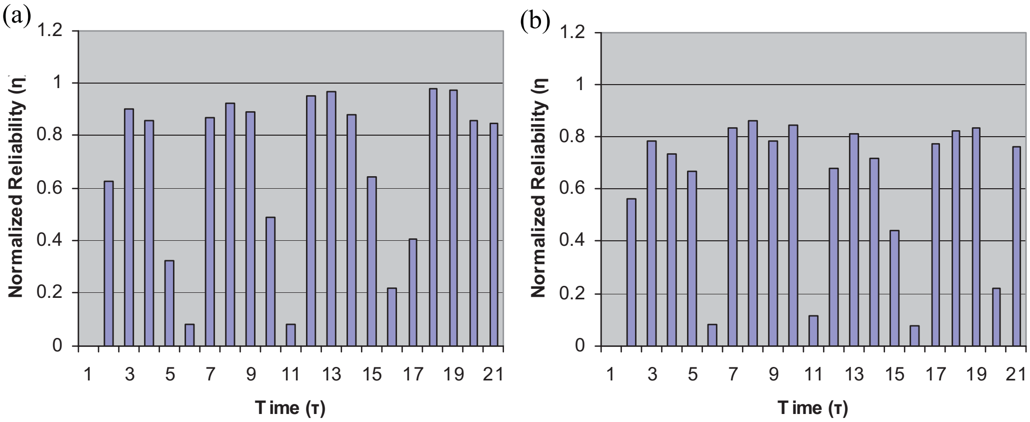


Fig. 2. Variation of normalized reliability (η) with decision time intervals for ESRT in OR condition. Here, the Reliability request is 4000 and 4500 packets per 10 s interval, respectively, for 2(*a*) and 2(*b*). Severe fluctuation in η is evident in both cases.

DOI: 10.1002/wcm

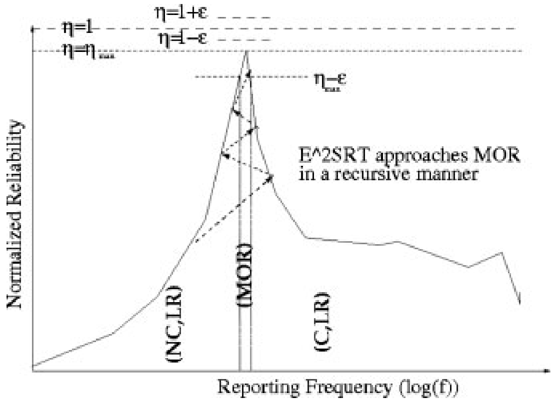


Fig. 3. A representative curve to show the effect of reporting rate (f) on normalized event reliability (η) in OR case. The ESRT fluctuates between the (NC, LR) and (C, LR) states whereas the E^{2}SRT scheme converges to MOR in a few decision intervals.

and fluctuates between two low reliability states (NC, LR) and (C, LR).

Lemma 1. *In OR case, normalized reliability, $\eta = r/R$, will never fall into the region of $[1-\varepsilon, \infty)$.*

Proof. Since $R_{\max}$ is the maximum reliability that the network can achieve with current network setting, it follows that observed event reliability $r_i \le R_{\max}$. Then, $\eta_i = r_i/R \le (R_{\max}/R) < 1-\varepsilon$. We conclude that $\eta_i \in (0, 1-\varepsilon)$.

Lemma 2. *In OR case, the network has only two possible states, namely (NC, LR) and (C, LR).*

Lemma 2 is a straight-forward extension of Lemma 1. However it reveals the most distinct characteristic of OR case, which is the basis for E^{2}SRT.

Lemma 3. *In and only in OR case, starting from the current network state $S_i = (NC, LR)$ and with linear reliability behavior when the network is not congested, the network state will transit to $S_{i+1} = (C, LR)$.*

Proof. From $S_i = (\mathrm{NC}, \mathrm{LR})$, ESRT aggressively increments f_i as follows:

$$f_{i+1} = \frac{f_i}{\eta_i}.$$

Since $f_{\max} = f_i \cdot \dfrac{R_{\max}}{r_i}$ and $R_{\max}/R < 1-\varepsilon$,

$$f_{i+1} = \frac{f_i}{\eta_i} = \frac{f_i}{\frac{r_i}{R} \cdot \frac{R_{\max}}{R_{\max}}} = f_{\max} \cdot \frac{R}{R_{\max}} > f_{\max} \cdot \frac{1}{1-\varepsilon}$$

Lemma 4. *Starting from $S_i = (C, LR)$, the network state will transit to $S_{i+1} = (C, LR)$ or (NC, LR).*

Proof. From $S_i = (\mathrm{C}, \mathrm{LR})$, ESRT aggressively decrements f_i as follows

$$f_{i+1} = f_i^{\frac{\eta_i}{k}}$$
$$f_{i+1} = f_i^{\frac{\eta_i}{k\log_{f_i} f_{\max}} \log_{f_i} f_{\max}} = (f_i^{\log_{f_i} f_{\max}})^{\frac{\eta_i}{k\log_{f_i} f_{\max}}}$$

where, k denotes the number of successive decision intervals for which the network has remained in (C, LR) state, including the current decision interval, i.e., $k \ge 1$. Here f is decreased more aggressively if a state transition is not detected.

Since $(f_{i+1}^{\log_{f_i} f_{\max}}) = f_{\max}$, it follows that

$$f_{i+1} = f_{\max}^{\frac{\eta_i}{k\log_{f_i} f_{\max}}} = f_{\max}^{\frac{\eta_i \lg f_i}{k \lg f_{\max}}} = f_{\max}^{\frac{\lg f_i^{\eta_i}}{\lg f_{\max}^k}}$$

If $f_i^{\eta_i} \ge f_{\max}^k$, we get $f_{i+1} \ge f_{\max}$;

or, if $f_i^{\eta_i} f_{\max}^k$, we get $f_{i+1} \le f_{\max}$.

Hence, in OR case, the network will either remain in (C, LR) state or transit to (NC, LR) state.

Based on the above analytical results, we redraw the ESRT state model for the OR case as shown in Figure 4.

4.2. The E^{2}SRT Solution

We address the following two issues in our proposed E^{2}SRT scheme:

(a) How to detect an over-demanding desired event reliability (R_{od}) situation, and
(b) If R_{od} problem exists, how to quickly converge to a maximum achievable reliability without requiring the full knowledge of network conditions.

For (a), our simulations and analytical results show that the network will have a direct transition between (NC, LR) and (C, LR) states only when R_{od} exists. Otherwise it will follow the standard ESRT state model. Therefore, our aim is to push the network to

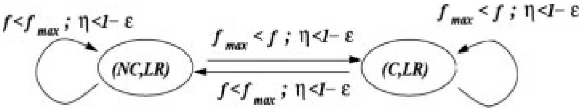


Fig. 4. ESRT protocol state model and transitions for over demanding desired event reliability (OR).

DOI: 10.1002/wcm

approach the maximum reliability point (MRP), ($f_{\max}$, $\eta_{\max}$), for a given network setting. For practical reasons, we allow a tolerance zone of width ε around MRP as illustrated in Figure 3, where ε is a user defined protocol parameter. If at the end of a decision interval i, the normalized reliability η_i is within $[\eta_{\max} - \varepsilon, \eta_{\max}]$ and no congestion is detected in the network, the network is in MOR. A smaller ε will generally give greater proximity to MRP but may need longer convergence time.

If the MRP is known, the sink can reduce the desired event reliability (R) such that the network can converge to OOR in E^2SRT scheme. However, it is difficult to calculate the exact value of MRP ($f_{\max}$, $\eta_{\max}$) due to the following reasons: (i) randomness in initial deployment; (ii) node movements, death or other reasons that change the network topology; (iii) relocation (or movement) of events; (iv) radio interference, and (v) deliberate over-demanding to maximize the network throughput.

A sophisticated algorithm should therefore adapt to the changing network environment, and determine the MRP in a recursive manner based on the feedback from network. As we mentioned in Section 1, the ESRT is the sink based transport protocol that serves to enforce reliability while conserving energy. The E^2SRT scheme adds several new components to ESRT in order to eliminate the R_{od} problem. However, the modification is essentially at the algorithm level. Thus, the proposed E^2SRT scheme inherits all the major features of ESRT such as the communication model and network mode definitions. As an enhanced version, E^2SRT is more resilient to abrupt network changes and resource constraints.

In Figure 3, we showed a typical convergence process of E^2SRT. It highlights the recursive property of E^2SRT algorithm. This recursive process will end when the network state fulfills the following two conditions: (*a*) the network is in (NC, LR), and (*b*) difference of normalized reliabilities of the last two consecutive states (η_{i-1} and η_i) is smaller than $\varepsilon/2$. The E^2SRT operation in each of the three available states is discussed below in the next section.

4.3. E^2SRT Protocol Operation

At the end of each decision interval (i), sink calculates the normalized reliability (η_i), and the current network state (S_i) is determined based on the congestion reports. Using values of S_i, f_i, η_i and the decision boundaries defined in ESRT, the E^2SRT scheme computes the value of sensor reporting frequency (f_{i+1}) for the next decision interval and broadcasts it to the sensor node(s). The corresponding sensor nodes report their event packets to the sink according to this updated frequency in the next decision interval. Here the congestion is estimated as in ESRT scheme. This process is repeated until the MOR state is reached. The state transition model is shown in Figure 5.

The E^2SRT introduces a recursive algorithm that converges to MOR in a few rounds of estimation of MRP. As observed from Figures 1 and 3, the network shows some linear and symmetry properties around MOR region in the normalized reliability curve as a function of reporting frequency ($\log f$). Furthermore, as we previously discussed, the network fluctuates between the (NC, LR) and (C, LR) states. The MRP is somewhere in between these two states. We denote the reporting frequencies of the last (C, LR) and (NC, LR) states as $f_{(\text{c,lr})}$ and $f_{(\text{nc,lr})}$, respectively. We estimate the updated reporting frequency as follows:

$$f_{i+1} = 10^{\dfrac{\log f_{(\text{nc,lr})} + \log f_{(\text{c,lr})}}{2}} \tag{1}$$

According to our simulation results, the network may stay in either (NC, LR) or (C, LR) state for more than one consecutive decision intervals, if f is too far

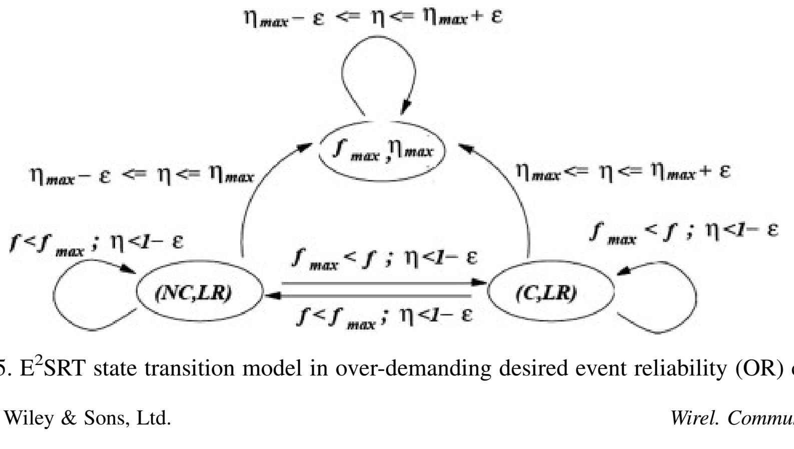


Fig. 5. E^2SRT state transition model in over-demanding desired event reliability (OR) case.

DOI: 10.1002/wcm

away from the state transition point. To improve the convergence rate, we give more weight to the last recorded frequency of opposite state in calculating the updated reporting frequency as discussed below.

(1) *(NC, LR) (No congestion, low reliability):* Since the OOR state is not feasible, goal of the frequency update policy is to drive the network to MOR. As pointed out by Lemma 3, using ESRT algorithm, the network would inevitably jump into the most undesirable (C, LR) state. Here, we already know that the network is in OR state, as it has at least once jumped to the (C, LR) state and then fell back to the (NC, R) state. The reporting frequency is updated as

$$f_{i+1} = 10^{\frac{1}{k+1}\log f_{(\mathrm{nc,lr})} + \frac{k}{k+1}\log f_{(\mathrm{c,lr})}} \qquad (2)$$

(2) *(C, LR) (Congestion, low reliability):* In this state, we either detect a transition from (NC, LR) state, so we know the network is now in OR state, or, the network remains in the (C, LR) states itself which means the frequency has to be further reduced. We count the time intervals (k) for which the network has successively remained in the (C, LR) state. As k increases, it generally means $f_{(\mathrm{nc,lr})}$ is closer to MOR than $f_{(\mathrm{c,lr})}$. We therefore assign it a higher weight than $f_{(\mathrm{c,lr})}$. Using these considerations, we update the reporting frequency as

$$f_{i+1} = 10^{\frac{k}{k+1}\log f_{(\mathrm{nc,lr})} + \frac{1}{k+1}\log f_{(\mathrm{c,lr})}} \qquad (3)$$

(3) MOR (Maximum operating region): In this state, the reporting frequency remains unchanged for the next decision interval as,

$$f_{i+1} = f_i$$

Figure 5 shows the state transition model of the E^{2}SRT scheme. The E^{2}SRT algorithm is summarized in the pseudo-code in Figure 6.

4.4. Analytical Results for E^{2}SRT

We use mathematical analysis in this section to show that E^{2}SRT converges to MOR state in OR case.

Lemma 5. *The network cannot stay in* (NC, LR) *or* (C, LR) *state for infinite number of iterations.*

Proof. Assume E^{2}SRT stays in (NC, LR) state for an infinite number of iterations, *i.e.*, k approaches infinity. According to the frequency update policy in Equation (2), f_{i+1} asymptotically approaches the last recorded $f_{(\mathrm{c,lr})}$, and drives E^{2}SRT to (C, LR) when k is sufficiently large. Therefore, the network cannot stay in (NC, LR) for infinite number of iterations. Similarly, we can show that E^{2}SRT cannot stay in (C, LR) for infinite number of iterations. This completes the proof.

Let us denote the logarithm of the sequence of frequencies generated by E^{2}SRT as $\{\log f^i_{(\mathrm{c,lr})} \,|\, i = 1, 2, 3, \ldots\}$ and $\{\log f^j_{(\mathrm{nc,lr})} \,|\, j = 1, 2, 3, \ldots\}$. We have the following lemma for these two sequences.

Lemma 6. *Both* $\{\log f^i_{(c,lr)} \,|\, i = 1, 2, 3, \ldots\}$ *and* $\{\log f^j_{(nc,lr)} \,|\, j = 1, 2, 3, \ldots\}$ *converges.*

Proof. From the frequency update policy of E^{2}SRT, it is easy to verify that

$$\log f^1_{(\mathrm{c,lr})} > \log f^2_{(\mathrm{c,lr})} > \cdots > \log f^i_{(\mathrm{c,lr})} > \log f^{i+1}_{(\mathrm{c,lr})} > \cdots,$$

and

$$\log f^1_{(\mathrm{nc,lr})} < \log f^2_{(\mathrm{nc,lr})} < \cdots < \log f^j_{(\mathrm{nc,lr})} < \log f^{j+1}_{(\mathrm{nc,lr})} < \cdots.$$

Moreover, we have $\{\log f^i_{(\mathrm{c,lr})} \,|\, i = 1, 2, 3, \ldots\}$ lower-bounded by $\log f_{\max}$, and $\{\log f^j_{(\mathrm{nc,lr})} \,|\, j = 1, 2, 3, \ldots\}$ upper-bounded by $\log f_{\max}$.

Since the sequence $\{\log f^i_{(\mathrm{c,lr})} \,|\, j = 1, 2, 3, \ldots\}$is monotonically decreasing and lower-bounded, it converges. Similarly, we can show that the sequence $\{\log f^j_{(\mathrm{nc,lr})} \,|\, j = 1, 2, 3, \ldots\}$ converges.

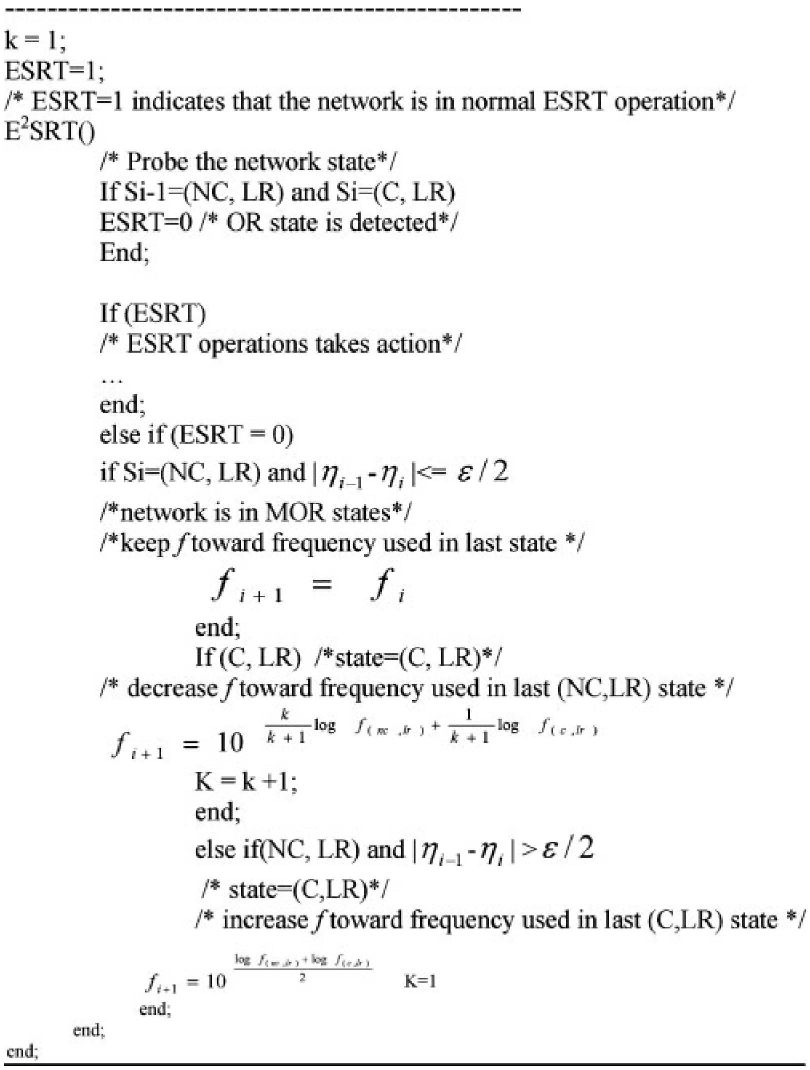
Theorem 1. *E^{2}SRT converges to MOR state in a finite number of iterations.*

The proof is provided in Appendix.

5. E^{2}SRT Performance Evaluation

In this section, we present simulation results for evaluating the performance of E^{2}SRT scheme. We used a simulation scenario with the 64 senders, tolerance $\varepsilon = 5\%$, and event radius of 40 m. Other simulation parameters were kept the same as those listed in Table I in ESRT [6]. Our results show that E^{2}SRT converges to a maximum operating point (MOR) when the network is in OR state.

DOI: 10.1002/wcm



```
-------------------------------------------------
k = 1;
ESRT=1;
/* ESRT=1 indicates that the network is in normal ESRT operation*/
E^2SRT()
        /* Probe the network state*/
        If Si-1=(NC, LR) and Si=(C, LR)
        ESRT=0 /* OR state is detected*/
        End;

        If (ESRT)
        /* ESRT operations takes action*/
        ...
        end;
        else if (ESRT = 0)
        if Si=(NC, LR) and |$\eta_{i-1}$ - $\eta_i$ |<= $\varepsilon / 2$
        /*network is in MOR states*/
        /*keep f toward frequency used in last state */
                $f_{i+1} = f_i$
                end;
                If (C, LR)  /*state=(C, LR)*/
        /* decrease f toward frequency used in last (NC,LR) state */
        $f_{i+1} = 10^{\frac{k}{k+1}\log f_{(nc,lr)} + \frac{1}{k+1}\log f_{(c,lr)}}$
                K = k +1;
                end;
                else if(NC, LR) and |$\eta_{i-1}$ - $\eta_i$ | > $\varepsilon / 2$
                 /* state=(C,LR)*/
                /* increase f toward frequency used in last (C,LR) state */
          $f_{i+1} = 10^{\frac{\log f_{(nc,lr)} + \log f_{(c,lr)}}{2}}$        K=1
          end;
     end;
end;
```

Fig. 6. Pseudo-code of the E^{2}SRT algorithm.

As shown in Figure 2, the normalized event reliability (η) achieved by ESRT scheme varies from as high as 0.95 to as low as 0.1. As we discussed in Section 3, this is because the network first uses a much higher value of reporting frequency, which leads to congestion. Then it reduces the frequency to a very low value to pull itself out of the congestion. This process repeats and causes the network to fluctuate. Unfortunately, this fluctuation cannot be eliminated by ESRT itself.

As shown in Figure 7, the normalized reliability is stabilized after about seven rounds of E^{2}SRT operation in OR case. The desired event reliability request is 4000 and 4500 packets, respectively, for Figure 7(a)

Table I. Throughput, latency, and loss rate achieved by the ESRT and E^{2}SRT schemes for R = 4000 and 4500 packets (values shown in bracket).

	Value in the first 20 decision intervals (ESRT)	Value in the first 20 decision intervals (E^{2}SRT)	Stable value in MOR state (E^{2}SRT)
Mean throughput (Mbps)	0.235 (0.206)	0.289 (0.273)	0.301 (0.301)
Mean latency (s)	0.313 (0.413)	0.147 (0.183)	0.047 (0.047)
Mean packet loss rate (%)	0.103 (0.148)	0.047 (0.042)	0.001 (0.001)
Standard deviation of normalized reliability σ	0.301(0.264)	0.082 (0.072)	0.0038 (0.0037)

DOI: 10.1002/wcm

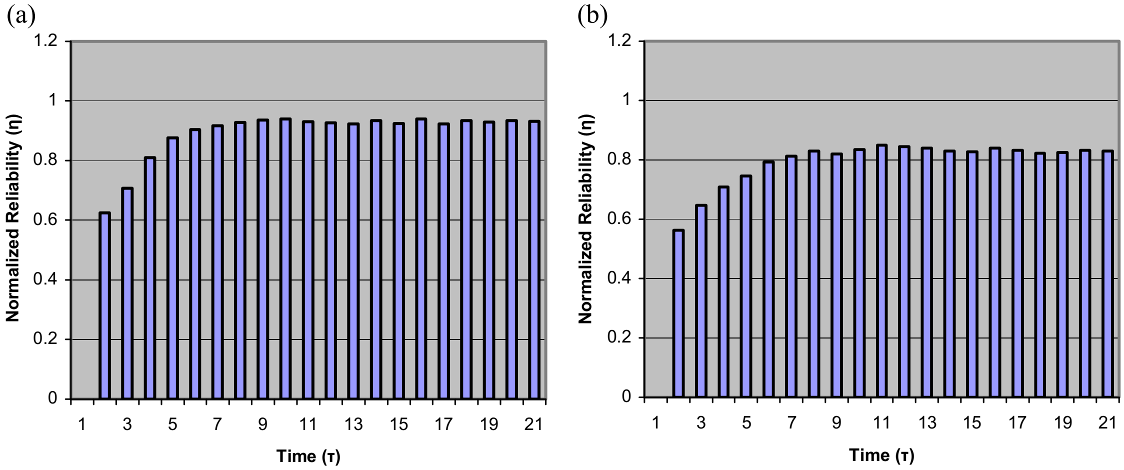


Fig. 7. Normalized event reliability achieved using the E^2SRT scheme in OR case. The desired event reliability is 4000 (left graph) and 4500 (right graph) packets for the 10 s decision interval. Network settings are the same as they are for the ESRT simulation. The maximum reachable reliability ($R_{\max}$) is around 3500 packet for the 10 s decision interval.

and (b). The maximum reachable reliability ($R_{\max}$) is around 3500 packets. The mean normalized reliability reached by the E^2SRT in the first 20 decision intervals is about 0.89 and 0.79, respectively, as compared to 0.68 and 0.61 for ESRT scheme. The proposed E^2SRT scheme thus improves the transport reliability performance by about 30% for the two different network settings compared to the ESRT scheme.

In Table I, we show the mean throughput, latency, loss rate, and standard deviation for ESRT and E^2SRT schemes for the above-mentioned simulation set up. Here, the throughput is measured as the total amount of data that reaches the sink per unit time. Note that throughput is proportional to the average attained reliability. The latency is measured as the average time delay experienced by data packets from the sensor to sink. To compute the latency, we set the timestamps for all data packets generated at the sensors and calculate the average time taken for the packets to reach the sink (not including the lost packets). The packet loss rate is calculated as $(1 - r_{\text{success}})$, where r_{success} is defined as the ratio of total number of packets successfully received by the sink to the total number of packets that have been generated by all the sensors during the time of measurement. In each run of the simulation, we sample the normalized reliability and use the formula $\sigma = \sqrt{\frac{1}{n}\sum_1^n (x - \bar{x})^2}$ to calculate the standard deviation. For each reliability request η, we run 10 independent simulations and take their average (arithmetic mean) as the final value.

As seen from Table I, the ESRT scheme provides a lower throughput, much higher latency and loss rate as compared to E^2SRT for our simulation set up. The proposed E^2SRT scheme successfully avoids the fluctuations and recursively converges to it best achievable reliability (i.e., MOR) values in a few decision intervals. The fluctuation of ESRT is obvious from the standard deviation of its normalized reliability, which is much higher than the proposed E^2SRT scheme. As expected, the E^2SRT scheme outperforms ESRT in all four performance measures.

6. Related Work and Discussion

In this section we discuss some of the recently proposed WSN transport protocols and their connection with E^2SRT. These schemes are categorized in two major groups: the downstream (from sink to sensor) oriented [7–10] and the upstream (from sensor to sink) oriented [6,11–13].

Among the downstream transport schemes, Wan *et al.* [7] developed the PSFQ (pump slowly and fetch quickly) scheme to support a simple and robust transport to deliver data from the sink to intended receivers. The key idea in PSFQ is to use a relatively slow rate to distribute data from source nodes (‘pump slowly’), but allow nodes experiencing data loss to aggressively recover the missing segments from immediate neighbors (local recovery, ‘fetch quickly’). In Reference [8], Park *et al.* proposed another solution that delivers entire messages with reduced time-delay compared to PSFQ. In Reference [9], DTC (distributed TCP caching) used the segment caching and local retransmissions to avoid expensive end-to-end retransmissions.

DOI: 10.1002/wcm

DTC provides interoperability with external TCP/IP networks, which make it possible to directly connect the sensor network with a wired network infrastructure, without proxies or middle-boxes. GARUDA [10] is a two-tier and two-stage negative acknowledgement (NACK)-based loss recovery scheme. It differentiates between the core (conceptually more important) nodes and non-core nodes by assigning different weights to their hop-count.

Among the upstream transport schemes, reliable multi-segment transport (RMST) [11] is based on directed diffusion [14]. It uses timer-driven selective NACK to enforce hop-by-hop (caching mode) and end-to-end (non-caching mode) reliability. RMST is lacking in adaptive design and a good congestion control mechanism. In order to improve channel utilization and to reduce acknowledgement (ACK)-loss related retransmission, reliable bursty convergecast (RBC) scheme [12] proposed a window-less block acknowledgment scheme for continuous packet forwarding guarantee. RBC is specially enhanced in its ACK handling capability with replication of the acknowledgments for received packets and mechanism to handle varying ACK-delay. Sensor TCP (STCP) [13] is a base station based generic transport protocol that can be reconfigured to support different applications and reliability requirement. STCP versatility also makes it a mediocre choice for specific requirement as well as potential implementation and communication overhead. Another issue with STCP is its heavy use of source (sensor node) caching, which is impractical for many WSN applications.

The ESRT and E^{2}SRT schemes fall in the category of upstream reliability oriented transport protocols. As mentioned earlier, the reliability and the success of sensor network applications are determined by the quantity of collective information instead of packet level reliability. However, a fundamental difference of ESRT and E^{2}SRT compared to other schemes in this group is their event-to-sink reliability assumption. By shifting away from the packet-level end-to-end or hop-by-hop reliability, ESRT and E^{2}SRT highlight the mission-specific characteristic of WSN and focus on information collection.

7. Conclusion

To overcome the OR problem defined in Section 3, we introduced the E^{2}SRT protocol featuring an adaptive algorithm that can detect the OR condition and recursively drive the network to work in its MOR, and prevent the network from unwanted fluctuations. If the desired reliability can be satisfied by the network, E^{2}SRT behaves the same as the ESRT scheme. After detection of OR state, E^{2}SRT launches a new set of operations to force the network to converge to MOR and prevent the fluctuation between two low reliability states. E^{2}SRT is more robust and more adaptive to the changing event and network environment. Our simulation results show that E^{2}SRT performs very well in the presence of OR. It stabilizes the network and improves the network performance by around 20–25% in terms of throughput with lower latency and loss rate compared to the ESRT scheme.

Appendix

Theorem 1. *E^{2}SRT converges to* $\log f_{max}$

Proof. The goals is to prove that for any given ε, we can find m that $|\log f^{i}_{(\mathrm{c,lr})} - \log f^{\max}| < \varepsilon$ and $|\log f^{j}_{(\mathrm{nc,lr})} - \log f^{\max}| < \varepsilon$ for any $i, j > m$.

Let $\log f^{*}_{(\mathrm{c,lr})}$ denote the converging frequency of sequence $\{\log f^{i}_{(\mathrm{c,lr})}\,|\, i = 1, 2, 3, \ldots\}$. Similarly, let $\log f^{*}_{(\mathrm{nc,lr})}$ denote the converging frequency of sequence $\{\log f^{j}_{(\mathrm{nc,lr})}\,|\, j = 1, 2, 3, \ldots\}$.

According to *Lemma* 6, for any given ϵ, we can find m_1 that $|\log f^{i}_{(\mathrm{c,lr})} - \log f^{*}_{(\mathrm{c,lr})}| < \alpha\varepsilon$ for any $i > m_1$. Here α is a positive scalar and can be made arbitrarily small. Similarly we can find m_2 that $|\log f^{j}_{(\mathrm{nc,lr})} - \log f^{*}_{(\mathrm{nc,lr})}| < \alpha\varepsilon$, for any $j > m_2$. Let $m > m_1$ and $m > m_2$, it is easy to verify that $|\log f^{i}_{(\mathrm{c,lr})} - \log f^{*}_{(\mathrm{c,lr})}| < \alpha\varepsilon$ and $|\log f^{j}_{(\mathrm{nc,lr})} - \log f^{*}_{(\mathrm{nc,lr})}| < \alpha\varepsilon$, for any $i, j > m$.

According to *Lemma* 5, E^{2}SRT cannot stay in (NC, LR) or (C, LR) for infinite number of iterations. Without loss of generality, we assume that at the $(m+1)$th iteration, E^{2}SRT goes form (NC, LR) to (C, LR). We have the following inequalities:

$$\begin{aligned}
&|\log f^{m}_{(\mathrm{c,lr})} - \log f^{*}_{(\mathrm{c,lr})}| < \alpha\varepsilon,\\
&|\log f^{m}_{(\mathrm{nc,lr})} - \log f^{*}_{(\mathrm{nc,lr})}| < \alpha\varepsilon,\\
&|\log f^{m+1}_{(\mathrm{nc,lr})} - \log f^{*}_{(\mathrm{nc,lr})}| < \alpha\varepsilon
\end{aligned}$$

and

$$\log f^{m}_{(\mathrm{nc,lr})} < \log f^{m+1}_{(\mathrm{nc,lr})}.$$

Also from *Lemma* 6, we have

$$\log f^{m}_{(\mathrm{nc,lr})} < \log f^{m+1}_{(\mathrm{nc,lr})} < \log f^{*}_{(\mathrm{nc,lr})} \leq \log f^{\max}$$

and

$$\log f^{m}_{(\mathrm{c,lr})} > \log f^{m+1}_{(\mathrm{c,lr})} > \log f^{*}_{(\mathrm{c,lr})} \geq \log f^{\max}.$$

DOI: 10.1002/wcm

Note that there are only four possible cases of convergence for any $i, j > m$.

Case 1: $|\log f^{j}_{(\mathrm{nc,lr})} - \log f^{\max}| \geq \varepsilon$ and $|\log f^{i}_{(\mathrm{c,lr})} - \log f^{\max}| \geq \varepsilon$ for any $i, j > m$
Case 2: $|\log f^{j}_{(\mathrm{nc,lr})} - \log f^{\max}| < \varepsilon$ but $|\log f^{i}_{(\mathrm{c,lr})} - \log f^{\max}| \geq \varepsilon$ for any $i, j > m$
Case 3: $|\log f^{i}_{(\mathrm{c,lr})} - \log f^{\max}| < \varepsilon$ but $|\log f^{j}_{(\mathrm{nc,lr})} - \log f^{\max}| \geq \varepsilon$ for any $i, j > m$
Case 4: $|\log f^{j}_{(\mathrm{nc,lr})} - \log f^{\max}| < \varepsilon$ and $|\log f^{i}_{(\mathrm{c,lr})} - \log f^{\max}| < \varepsilon$ for any $i, j > m$

If Case 4 is the only feasible condition, we are done. To do this, we prove that Cases 1, 2, and 3 are infeasible at iteration $m + 1$.

Assume Case 1 is true, i.e., $|\log f^{*}_{(\mathrm{c,lr})} - \log f^{\max}| \geq \varepsilon$ and $|\log f^{*}_{(\mathrm{nc,lr})} - \log f^{\max}| \geq \varepsilon$. We have

$$
\begin{aligned}
&|\log f^{*}_{(\mathrm{c,lr})} - \log f^{m+1}_{(\mathrm{c,lr})}| \\
&= \log f^{*}_{(\mathrm{c,lr})} - \log f^{m+1}_{(\mathrm{c,lr})} \\
&= \log f^{*}_{(\mathrm{c,lr})} - \left(\frac{1}{k+1} \log f^{m}_{(\mathrm{nc,lr})} + \frac{k}{k+1} \log f^{m}_{(\mathrm{c,lr})} \right) \\
&= \left(\frac{1}{k+1} \log f^{*}_{(\mathrm{c,lr})} - \frac{1}{k+1} \log f^{m}_{(\mathrm{nc,lr})} \right) + \left(\frac{k}{k+1} \log f^{*}_{(\mathrm{c,lr})} - \frac{k}{k+1} \log f^{m}_{(\mathrm{c,lr})} \right) \\
&> \left(\frac{1}{k+1} \log f^{*}_{(\mathrm{c,lr})} - \frac{1}{k+1} \log f^{*}_{(\mathrm{nc,lr})} \right) + \left(\frac{k}{k+1} \log f^{*}_{(\mathrm{c,lr})} - \frac{k}{k+1} \log f^{m}_{(\mathrm{c,lr})} \right) \\
&= \left(\frac{1}{k+1} \log f^{*}_{(\mathrm{c,lr})} - \frac{1}{k+1} \log f^{\max} \right) + \left(\frac{1}{k+1} \log f^{\max} - \frac{1}{k+1} \log f^{*}_{(\mathrm{nc,lr})} \right) \\
&= \left(\frac{1}{k+1} \log f^{*}_{(\mathrm{c,lr})} - \frac{1}{k+1} \log f^{\max} \right) + \left(\frac{1}{k+1} \log f^{\max} - \frac{1}{k+1} \log f^{*}_{(\mathrm{nc,lr})} \right) \\
&+ \left(\frac{k}{k+1} \log f^{*}_{(\mathrm{c,lr})} - \frac{k}{k+1} \log f^{m}_{(\mathrm{c,lr})} \right) \\
&> \frac{1}{k+1} \varepsilon + \frac{1}{k+1} \varepsilon - \alpha \frac{k}{k+1} \varepsilon \\
&> \frac{2 + \alpha k}{k+1} \varepsilon \\
&> \alpha \varepsilon
\end{aligned}
$$

The last inequality is true because we can make α arbitrarily small, e.g., for any $\alpha < \frac{2}{2k+1}$, we have $|\log f^{m+1}_{(\mathrm{c,lr})} - \log f^{*}_{(\mathrm{c,lr})}| > \alpha\varepsilon$.

This contradicts the condition that $|\log f^{i}_{(\mathrm{c,lr})} - \log f^{*}_{(\mathrm{c,lr})}| < \alpha\varepsilon$ for any $i >$ m.

Put it together, we conclude that Case 1 is invalid and should be removed from consideration.

Similarly, we can prove that Case 2 and Case 3 are invalid.

Given E^2SRT's frequency update policy, at least one of the four cases should be true. Therefore, Case 4 is the only valid case.

This completes the proof.

Acknowledgment

This work was partially supported by a grant from Cisco University Grant Program (UGP) under a subcontract to Dr S. Kumar.

References

1. Xu N, Rangwala S, Chintalapudi K, Ganesan D, Broad A, Govindan R, Deborah E. A wireless sensor network for structural monitoring. *ACM Sensys*, November 2004.
2. Mainwaring A, Polastre J, Szewczyk R, Culler D, Anderson J. Wireless sensor networks for habitat monitoring. *WSNA*, September 2002.
3. Lundquist J, Cayan D, Dettinger M. Meteorology and hydrology in Yosemite National Park: a sensor network application. *Information Processing in Sensor Networks* (IPSN), April 2003.
4. He T, Krishnamurthy S, Stankovic JA, Abdelzaher T, Luo L, Stoleru R, Yan T, Gu L, Hui J, Krogh B. Energy-efficient surveillance system using wireless sensor networks. *Proceedings of 2nd International Conference on Mobile Systems, Applications and Services*, 6–9 June, 2004, Boston, MA, USA.

DOI: 10.1002/wcm

5. *RFC 793*, Transmission Control Protocol, http://www.ibiblio.org/pub/docs/rfc/rfc793.txt
6. Akan OB, Akyildiz IF. Event-to-sink reliable transport in wireless sensor networks. *IEEE/ACM Transactions on Networking* 2005; **13**(5): 1003–1016.
7. Wan CY, Campbell AT, Krishnamurthy L. PSFQ: a reliable transport protocol for wireless sensor networks. *WSNA*, Atlanta, September 2002.
8. Park S-J, Sivakumar R. Sink-to-sensors reliability in sensor networks. *MobiHoc*, Annapolis, MD, USA, June 2003.
9. Dunkels A, Alonso J, Voigt T, Ritter H. Distributed TCP caching for wireless sensor networks. *Proceedings of the 3rd Annual Mediterranean Ad Hoc Networking Workshop*, Bodrum, Turkey, 27–30 June, 2004.
10. Park S-J, Vedantham R, Sivakumar R, Akyildiz IF. A scalable approach for reliable downstream data delivery in wireless sensor networks. *Proceedings of the ACM MobiHoc*, Roppongi, Japan, 24–26 May, 2004.
11. Stann F, Heidemann J. RMST: reliable data transport in sensor networks. *Proceedings of 1st International Workshop Sensor Net Protocols and Applications*, Anchorage, Alaska, USA, April 2003; 102–112.
12. Zhang H, Arora A, Choi YR, Gouda MG. Reliable bursty convergecast in wireless sensor networks. *Proceedings of the ACM Mobihoc*, Urbana-Champain, IL, 25–28 May, 2005.
13. Iyer YG, Gandham S, Venkatesan S. STCP: a generic transport layer protocol for wireless sensor networks. *Proceedings of IEEE ICCCN*, San Diego, CA, 17–19 October, 2005.
14. Intanagonwiwat C, Govindan R, Estrin D. Directed diffusion: a scalable and robust communication paradigm for sensor networks. *Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom 2000)*, Boston, MA, USA, August 2000; 56–67.

Authors' Biographies

Sunil Kumar is an Associate Professor and Thomas G. Pine Faculty Fellow in the Electrical and Computer Engineering department at San Diego State University, San Diego, California, U.S.A. From August 2002 to July 2006, he was an Assistant Professor in Electrical and Computer Engineering at Clarkson University, Potsdam, NY. He received M.E. and Ph.D. in Electrical and Electronics Engineering from Birla Institute of Technology and Science, Pilani (India) in 1992 and 1997, respectively. His research interests include QoS-aware Cross-layer Protocols for Multimedia Traffic in Wireless Networks, and Robust Multimedia Compression techniques. He has published more than 80 research articles in international journals and conferences, including two book/book chapters. His research has been supported by U.S. National Science Foundation (NSF), U.S. Air Force, DOE, Cisco and Sprint.

Zhenhua Feng received his B.E. degree in Electrical Engineering from Beijing Institute of Technology, Beijing, China, in 2003, and his M.S. degree in Computer Engineering from Clarkson University, Potsdam, NY, U.S.A. in 2006. He is currently a Ph.D. student and research assistant in the Bradley Department of Electrical and Computer Engineering at Virginia Tech. His current research focuses on algorithms and optimization for frequency-agile radio networks, MIMO, and cooperative communication networks.

Fei Hu is an associate professor in the Department of Electrical and Computer Engineering at the University of Alabama, Tuscaloosa, AL, U.S.A. His research interests are wireless networks, wireless security and their applications in Bio-Medicine. His research has been supported by NSF, Cisco, Sprint, and other sources. He obtained his first Ph.D. degree at Shanghai Tongji University, China in 1999, and second Ph.D. degree at Clarkson University, U.S.A. in 2002, all in the field of Electrical and Computer Engineering. He has published over 100 journal/conference papers and book (chapters). He is also the editor for over five international journals.

Yang Xiao worked in industry as a MAC (medium access control) architect involving the IEEE 802.11 standard enhancement work before he joined Department of Computer Science at The University of Memphis in 2002. He is currently with Department of Computer Science at The University of Alabama. He currently serves as Editor-in-Chief for *International Journal of Security and Networks* (IJSN), *International Journal of Sensor Networks* (IJSNet), and *International Journal of Telemedicine and Applications* (IJTA). He also serves as an associate editor for several journals, including *IEEE Transactions on Vehicular Technology.* He serves on TPC for more than 100 conferences such as INFOCOM, ICDCS, MOBIHOC, ICC, GLOBECOM, WCNC, etc. His research areas are security, telemedicine, sensor networks, and wireless networks. He has published more than 280 papers in major journals (more than 60 in various IEEE journals/magazines), refereed conference proceedings, and book chapters. Dr Xiao's research has been supported by the U.S. National Science Foundation (NSF), U.S. Army Research, Fleet and Industrial Supply Center San Diego (FISCSD), and The University of Alabama's Research Grants Committee.

DOI: 10.1002/wcm