Autonomous Traffic Engineering for Boosting Application Fidelity in Wireless Sensor Networks

Md. Abdur RAZZAQUE†, Nonmember, Choong Seon HONG†(a), Member, and Sungwon LEE†, Nonmember

SUMMARY This paper presents an autonomous traffic engineering framework, named ATE, a highly efficient data dissemination mechanism for multipath data forwarding in Wireless Sensor Networks (WSNs). The proposed ATE has several salient features. First, ATE utilizes three coordinating schemes: an incipient congestion inference scheme, an accurate link quality estimation scheme and a dynamic traffic diversion scheme. It significantly minimizes packet drops due to congestion by dynamically and adaptively controlling the data traffic over congested nodes and/or poorer quality links, and by opportunistically exploiting under-utilized nodes for traffic diversion, while minimizing the estimation and measurement overhead. Second, ATE can provide high application fidelity of the network even for increasing values of bit error rates and node failures. The proposed link quality estimation and congestion inference schemes are light weight and distributed, improving the energy efficiency of the network. Autonomous Traffic Engineering has been evaluated extensively via NS-2 simulations, and the results have shown that ATE provides a better performance with minimum overhead than those of existing approaches.

key words: wireless sensor networks, congestion avoidance, application fidelity, traffic engineering, energy efficiency

1. Introduction

Wireless Sensor Networks (WSNs) are comprised of a variable number of autonomous electronic devices, and mechanical components, that have the capability of remote sensing, signal processing and communication in an ad hoc fashion. Sensed data packets are forwarded to the base station (usually known as the sink node) in multihop way and the intermediate nodes act as router for the data packets. The robust and autonomous nature of WSNs make them indispensable aids to mission-critical networking applications, such as battlefield surveillance, radioactive leakage detection, disaster response, and emergency medical care.

The fidelity of an application may be defined as how well its requirements are guaranteed by the data dissemination framework of the network. A typical WSNs application may include a number of performance metrics including the end-to-end data delivery throughput, packet latency, and packet loss ratio. However, conformation to these metrics may be hampered if severe congestion occurs in the network [1]–[3]. Unlike in traditional networks, the problem of congestion avoidance or control in WSNs may be attributed to the fact that, data generated during a crisis state are of utmost importance, and loss of crisis data negates the purpose of deploying an unattended sensor network. In addition, controlling congestion by throttling the source data generation rate is undesirable since it may significantly violate the fidelity requirements.

Congestion detection and control problems in sensor networks usually involve detecting node/link-level congestion in the network and thereby controlling the source data packet generation rates to diminish packet losses [3]–[5]. Although traffic rate control strategies are effective in alleviating congestion, they are unsuitable for special applications for the following two reasons. First, some emerging mission-critical applications, such as imaging and battlefield monitoring demand high-rate data deliveries, i.e., they must be able to transport large volumes of loss-intolerant data concurrently from several sensor nodes. Avoiding congestion by throttling the traffic rate will violate the fidelity requirements. Therefore, it would be better to increase the network capacity by utilizing more resources in order to accommodate excessive incoming traffic during the crisis state. Fortunately, it is common practice in WSNs to densely deploy nodes and to allow unused nodes to go into sleep states so that energy is saved during dormant periods, providing elastic resource to the sensor networks. This advantage enables WSNs to employ adaptive capacity planning schemes to simultaneously avoid possible congestion and satisfy fidelity requirements. Second, it is highly likely that the congestion caused by burst traffic is transient in nature. For example, the sensor nodes will generate a transient burst of traffic when abnormal events occur. It would be inefficient to cope with this transient congestion using feedback-based traffic control. Rather, it may be alleviated through rapid adjustment of network resource provisioning.

However, congestion avoidance based on resource control has received little attention so far. Topology-aware resource adaptation, TARA [1], activates appropriate sensor nodes whose radio is off to form a new topology that has just enough capacity to handle the increased traffic. But, TARA requires end-to-end topology information and an estimate of the capacity using a graph-theoretic approach, which is not suitable for energy-constraint sensor nodes. SIPHON [2] introduces a small number of additional multi-radio virtual sinks that are capable of siphoning off excess data packets from the hot spot region. However, SIPHON’s distributed algorithms, required for virtual sink discovery and selection, not only add overheads to the energy-constraint sensor nodes but also increase the packet delay.

The main objective of traffic engineering is to enhance
performance of the network, on both the traffic and the resource levels. In this paper, we propose an autonomous traffic engineering framework, ATE, to route packets around the hot spot areas, and in response to congestion, to scatter excess packets along multiple paths in which there are idle or under-loaded nodes. The cornerstone of the ATE framework is its method for estimating the downstream link qualities of different forwarding nodes and its ability to infer the incipient congestive state. Autonomous traffic engineering opportunistically exploits higher quality links and/or under-utilized nodes for traffic diversion. Here, in fact, the sleeping nodes are used on demand as safety valves to divert a portion of data packets from areas of high traffic load, alleviating the funneling effect; that is, the many-to-one multipath traffic flows (i.e., from many sensors to the sink) converge somewhere near the sink node, causing congestion in the area, that often characterizes sensor network communications [6]. This is done in order to maintain the fidelity of applications at the sink. For inferring a congestive state, ATE exploits both the buffer occupancy and the degree of media contention of a node. It also exploits the successful packet delivery rates of the sensor nodes during congestion sharing/controlling. Autonomous Traffic Engineering is energy efficient and can boost the application fidelity index by upto 18.3% compared to that of TARA [1].

The rest of this paper is organized as follows. In Sect. 2, we describe the motivation for the ATE framework and introduce key limitations of the existing congestion control schemes. Subsequently, we present a baseline model for the multipath routing protocol used by our ATE in Sect. 3. Section 4 presents the ATE architecture in detail, followed by the performance evaluations using Network Simulator-2 [28] in Sect. 5. Finally, we conclude the paper in Sect. 6.

2. Motivation

In this section, we first explain the necessity of traffic engineering in wireless sensor networks. Then, we identify and explain the limitations of existing techniques.

2.1 Why Autonomous Traffic Engineering?

Wireless sensor networks demand autonomic solutions because they are inaccessible and unattended in nature. Autonomous traffic engineering in WSNs can be defined as an independent and always available tool that addresses the data traffic routing problems in order to guarantee compliance with user requirements. Without any human interventions, it would dynamically analyze, predict and regulate the behavior of data traffic transmitted over the network. More specifically, it answers the question “How can the data traffic be spread over multiple paths such that it ensures high data delivery throughput and minimizes resource utilization, and all the traffic is delivered without unreasonable delay?” Such a mechanism would be very effective for improving performance in multihop WSNs, where unattended nodes have limited buffer spaces and wireless link quality fluctuates significantly. The following use-cases describe the importance of traffic engineering in WSNs.

Congestion Avoidance. In wireless sensor networks, data traffic generated by event-detecting sensor nodes moves quickly toward the sink node and this many-to-one traffic pattern creates a funneling effect [6], leading to increased transit traffic intensity at the intermediary nodes. Congestion is very likely to occur there due to excessive media contention and/or buffer overflows. To alleviate the congestive state, autonomous traffic engineering may be employed to divert traffic from highly-loaded nodes to under-utilized nodes.

Enhancing Application Fidelity. End-to-end path-based data delivery mechanisms are problematic for dynamic sensor networks since service disruption during path discovery is not acceptable in mission-critical applications. Path reservation-based approaches are not suitable due to their huge overhead for path discovery and recovery in large scale sensor networks. Therefore, hop-by-hop based energy-aware data delivery with dynamic traffic engineering is desirable for fidelity enhancement in large scale WSNs.

Uniform Resource Utilization. The uniform utilization of available resources (e.g., energy) in WSN nodes is critical since it can increase network lifetime. A traffic engineering framework may assign more of the total traffic load to under-utilized nodes and less to over-committed nodes improving the uniform resource utilization of available nodes in the network.

Figure 1 shows a small part of a wireless sensor network illustrating the advantage of traffic engineering. Suppose that if nodes A, B, C and D send or relay data packets on their respective shortest paths, nodes F and E will quickly become congested. If node B blindly scatters its excessive packets on a random detour path in an attempt to alleviate the congestion on its shortest path, for example, forwarding packets to node E, the congestion at other hot spots may further increase. If the forwarding node takes into account the load status on its immediate neighbors when distributing the excess packets, an appropriate detour path consisting of idle or under-loaded nodes (such as $B \rightarrow G \rightarrow H \rightarrow I \rightarrow Sink$) can be used intentionally, but not blindly. Motivated by this understanding, in this paper, we follow a dynamic capacity planning philosophy to alleviate congestion and to adhere to

![Fig. 1 Illustration of traffic engineering.](image-url)
the fidelity requirement by designing an autonomous traffic engineering scheme. Considering the features of large-scale sensor networks and the practicality of the protocol implementation, our framework needs low computational overhead (compared to graph-theoretic approaches in [1]) to allow timely execution on slow processors, maintains a minimal amount of state information, and dispenses with global knowledge and extra overhead, such as end-to-end topology information.

2.2 Limitations of Existing Approaches

Existing solutions in the literature can broadly be classified into two categories: 1) traffic rate control based and 2) network resource control based. The former is employed by most studies. Congestion detection and avoidance, CODA [3], presents the first detailed investigation of congestion control in sensor networks, where congestion is detected by sampling the traffic loading of the medium and monitoring the queue occupancy. When a node detects congestion, it broadcasts a backpressure message upstream, and the upstream nodes throttle the traffic volume to alleviate the congestion. In event-to-sink reliable transport, ESRT [4], the sink node is required to periodically configure the source-sending rates in order to avoid congestion. In priority-based sink node is required to periodically configure the source-sending rates in order to avoid congestion. In priority-based congestion control protocol, PCCP [5], the incipient congestion is detected using the ratio of packet service time to the inter-arrival time at a node. When this ratio becomes greater than 1, congestion is detected. PCCP does not care about the current state of the node buffer; hence, false congestion may be frequently identified due to the temporary unfairness of the underlying medium access control protocol. Although traffic control can effectively alleviate congestion, it could impose a negative impact on the application fidelity.

The second category of congestion control mechanisms uses a resource-control strategy rather than one of traffic control. In [7], Kang et al. first proposed resource provisioning to alleviate congestion and to improve throughput. This system incipiently determined the influence of multiple paths on the end-to-end channel capacity and provided some guidelines for the design of resource control algorithms. Topology-aware resource adaptation, (TARA) [1], activates appropriate sensor nodes whose radio is off to form a new topology with just enough capacity to handle the increased traffic. To efficiently estimate the capacity using a graph-theoretic approach, TARA requires not only local knowledge, but also knowledge about the end-to-end topology. This overhead is too high to allow the network to scale up to a large number of nodes. Another scheme in [2] introduces a small number of multi-radio virtual sinks, randomly or selectively distributed across the sensor field, that are capable of removing excess data packets from the hotspot region. However, the distributed algorithms, required for virtual sink discovery and selection, not only increase the overheads of the energy-constraint sensor nodes but also increase the packet delivery delay.

Among existing systems, TARA provides the most effective solution and is the most similar to our approach. However, there are a few fundamental differences between the two. First, TARA operates on a single path routing network, whereas ATE uses multiple forwarder nodes to manage the traffic loads. Moreover, unlike in TARA, the ATE nodes forward packets to neighbor nodes located closer to the sink. No prior path setup or path recovery mechanisms are required. Second, traffic rerouting in TARA is only instigated by the detection of congestion, and it always creates a new path to divert the additional traffic. TARA does not react to link quality fluctuations, whereas in ATE, both the downstream link quality and the incipient congestive states may provoke a node to reroute traffic loads toward under-utilized or idle nodes enabling ATE to achieve higher fidelity levels. Third, ATE has its own congestion inference mechanism, which is more accurate and energy-efficient than that of TARA, which borrows its congestion-detection mechanism from other works [3], [8] and TARA does not explicitly mention the condition of congestion detection.

3. Preliminaries

In this section, we present the network model used in this paper and describe the basic multipath routing algorithm that determines the set of available next-hop nodes for each sensor node.

3.1 Network Model

We consider a WSN consisting of a sink node $S$, an array of homogeneous wireless sensor nodes $N$ with a set of interconnecting links $L$ between them. We assume that each sensor node $i \in N$ knows the set of its next hop neighbors $N_i$ (i.e., downstream nodes) through whom it will forward data packets to the sink. A typical multipath routing algorithm that could allow a node to learn this is described in Sect. 3.2. Let $L^i_0$ and $L^i_1$ denote the set of upstream and downstream links, respectively, connected to node $i \in N$. The set of all neighbor nodes is $N_i$. Furthermore, given node $i \in N$, $r^i_{l,0}$ denotes the data packet rate arriving to node $i$ through link $l \in L^i_0$. Similarly, $r^i_{l,1}$ denotes the data packet rate departing node $i$ through link $l \in L^i_1$. The above notations are shown in Fig. 2.

Now, given any node $i \in N$ and any link $l \in L_i$, let $d(l, i)$ and $u(l, i)$ denote the downstream and upstream nodes
of \( i \), respectively. Note that for any two neighbor nodes, we can write \( r^{out}_{il} = r^{in}_{li} \) or \( r^{out}_{ul} = r^{in}_{lu} \) if there is no link loss. Therefore, the aggregated traffic load of any node \( i \in \mathcal{N} \) may be calculated as, \( r^{agg}_{i} = \sum_{k \in \mathcal{N}} (r^{out}_{ik}) + r_{i} \), where \( r_{i} \) is the data packet generation rate of node \( i \). Therefore, the aggregated traffic load of a source only node is simply its data packet generation rate. We assume that each node splits its aggregated traffic load over the active downstream links by striving to conserve the traffic flow (i.e., aims at no losses) and to increase the throughput. With respect to Fig. 2, for example, node \( i \) attempts to satisfy the following flow conservation constraint

\[
r^{agg}_{i} = r^{out}_{j,i} + e^{out}_{g,j} + r^{out}_{h,i} + r_{i} = r^{out}_{i,j} + e^{out}_{j,i}.
\]

(1)

3.2 Multipath Routing Algorithm

End-to-end (disjoint or partially disjoint) path discovery based multipath routing algorithms [9], [10] are not suitable for large-scale dynamic wireless sensor networks mainly due to two reasons: (i) increased overheads of discovery and maintenance of long routes and (ii) unacceptable path discovery latency for periodic urgent packets [11]–[13]. Rather, autonomic and self-organizing sensing and fusion characteristics of large-scale WSNs necessitate a lightweight routing mechanism that does not require nodes to maintain global state information and provide high scalability and robustness. Geographic routing, that implements hop-by-hop routing, has been proven to be more suitable in meeting up the above requirements. MMSPEED [12] uses multi-speed paths exploiting nodes’ geographic locations, but it does not consider their residual energy levels. In [11], location-aware nodes take urgency factor of real-time packets and nodes’ residual energy levels into account for creating paths. In the following, we design an energy-balanced multipath geographic forwarding (EBMGF) algorithm exploiting normalized geographic progress and nodes’ residual energy levels.

We assume that the sink and sensor nodes are aware of their geographic location information either via GPS (Global Positioning System) or other location determination techniques such as [14], [15]. In the case, each sensor node broadcasts periodic BEACON messages to its single-hop neighbors that carries the nodes’ \((x, y)\) position and residual energy \(e_{res}\) information. The sink node broadcasts its location update message to all nodes either by single hop [4], [6] or multihop fashion [16]. Thus, each sensor node learns the geographic locations of the sink and all of its neighbor nodes as well as each neighbor’s residual energy levels. A node may use the neighbors that are closer to the sink than it is and that have sufficient residual energy to be the next-hop nodes. More explicitly, an energy-balanced multipath geographic forwarding algorithm (EBMGF) may be employed that calculates an aggregate weight value \( W_j \) for each candidate next-hop nodes \( j \in \mathcal{N}_j \) as of Eq. (2) and thereby chooses the nodes with higher weight values for multipath data forwarding.

\[
W_j = w_p \cdot \frac{dist(i, S) - dist(j, S)}{dist(i, S)} + w_e \cdot \frac{e_{res}^{j}}{e_{init}^{j}}.
\]

(2)

Here, \( dist(i, S) \) represents the Euclidian distance between node \( i \) and the sink \( S \), and \( e_{init}^{j} \) is the initial energy of node \( j \); \( w_p \) and \( w_e \) are nonnegative weight factors conditioning that \( w_p = C \cdot \frac{dist(S)}{T_{life}} \) and \( w_e = (100 - w_p) \), with \( C \) being a constant multiplier and \( T_{life} \) the packet lifetime. The philosophy behind the above equation is as follows. We try to assign maximum priority to the progress factor for packets with high emergency \( (dist(i, S)/T_{life}) \) and lesser priority to the energy factor. It makes perfect sense, because for time critical packets with aggressive deadlines, our major concern should be delivering the packets in time without having to worry about uniform energy utilization of neighbor nodes.

Note also that, in Eq. (2), only the next-hop nodes that satisfy the condition \( dist(i, S) > dist(j, S) \) are considered. More elaborately, the weight value \( W_j \) is a linear combination of two parameters. The first parameter is the normalized geographic progress, which indicates how much geographic progress a packet can make toward the destination sink \( S \). If several candidate downstream nodes have equal residual energies, maximizing Eq. (2) decreases the number of hops a packet has to travel before it reaches the destination sink, which in turn may decrease the energy consumption. The second parameter, the ratio of residual energy \( e_{res}^{j} \) to initial energy \( e_{init}^{j} \), represents the fraction of energy available at the downstream node \( j \in \mathcal{N}_j \). This ensures a balanced energy consumption among the nodes.

The energy-balanced multipath geographic forwarding algorithm (EBMGF), which is used to identify a list of candidate downstream nodes sorted in descending order of their weight values, is presented in Algorithm 1 of Fig. 3.

4. The ATE Architecture

While our EBMGF algorithm is effective in balancing the energy consumption of the sensor nodes, it deliberately fails...
to capture the effects of link quality fluctuations and traffic bursts, resulting in its inability to guarantee the application fidelity. In the next subsections, we describe the congestion control components of the proposed ATE framework that significantly minimize the packet drops due to congestion by (i) dynamically and adaptively decreasing the data traffic over inferior quality links and/or congested nodes, and (ii) opportunistically exploiting superior quality links and/or under-utilized (sleeping) nodes for traffic diversion, while minimizing the estimation and measurement overheads.

4.1 Choosing Multiple Forwarders and Traffic Splitting

Splitting the generated data traffic from a source sensor node and transmitting over multiple paths towards destination node has been proven to achieve higher data delivery performance [10], [11]. In our ATE design, we allow each ATE source sensor node to transmit its outgoing traffic over two forwarder nodes concurrently, namely F1 and F2 while the other intermediate nodes just forward the traffic without employing any splitting policy. An ATE source sensor node may switch on to other forwarder node(s) F3 only when it fails to preserve the flow conservation constraint Eq. (1) either due to congestion or degradation of forwarding link qualities. The rationale for using two paths for traffic splitting (i.e., load balancing) is that there is little or no gain in aggregate throughput from using more than two paths [17]. Initially, a source node i ∈ N starts data packet transfer toward the destination sink S using the top two next-hop nodes selected from the sorted list of neighbors N i

As long as the network operates with a light traffic load and the link qualities do not vary significantly, our traffic splitting policy works as follows. An ATE source node first transmits a packet on F1. Four packet transmission time intervals later, the source node transmits the next packet on F2, followed by four packet transmission intervals and another packet transmission on F1. Therefore, the source loads each forwarding link with one packet for every eight packet transmission intervals, resulting in a packet loading rate of 1/8 link data rates per path, as shown in Fig. 4. The intermediate source nodes, that also act as routers of data traffic from other source node(s), reduce the number of packet transmission intervals between two consecutive transmissions (e.g., 1/6 or 1/4 link data rates per path) according to their aggregated traffic loads. We believe that the above packet loading rate is quite reasonable for many moderate to high rate sensor applications (e.g., medical imaging, battlefield monitoring, moving object tracking, etc).

Since the above traffic splitting policy neither considers the forwarding link quality nor the network congestive states, it may not guarantee any conservation constraint on the senders’ flow. To cope with this, in the following two subsections, we first discuss how each ATE node (being source or not) infers network congestion and estimates the forwarding link qualities in order to activate the appropriate traffic engineering action.

4.2 Incipient Congestion Inference

Many-to-one traffic flows in WSNs might converge somewhere near the sink node [6] and nodes in that area may be overloaded. Subsequently, the nodes surrounding in the overloaded node would quickly be overloaded as well. This may be caused by the fact that many of the neighbors of the overloaded node also have data packets to send, competing for the medium access with the overloaded node and thus increasing the media contention quickly, which in turn increases the packet collisions and the number of retransmissions for each packet. Thus the system forces the surrounding node buffers to grow as well, leading to a congestive state. In addition to that, the communications among the nodes, 2/3-hops away from the overloaded node, might be affected due to the hidden/exposed terminal problem and severe situation will occur if the number of such transmitting nodes rises to high in the surrounding area, i.e., the probability of existing hidden/exposed terminal increases [18], [19].

Accurately inferring a future congestive state is an important challenge in designing ATEs. In the literature on wireless and sensor networks, a number of approaches have been proposed for measuring the level of congestion at a node, including, detections based on buffer occupancy [20], [21], channel utilization [3] and the ratio of packet service time to inter-arrival time [5]. In buffer-based schemes, a sensor sends a packet to its downstream neighbor only when the latter has free buffer space in which to hold the packet. In the Light Weight Buffer (LWB [20]) management system, every sensor advertises only one-sixth of its remaining buffers so as to avoid congestion and to solve the hidden terminal problem. Although this method can guarantee the packet is not dropped during forwarding, the buffer utilization is very low and the associated communication overhead significantly decreases the throughput of the network. Moreover, LWB can’t infer a congestive state due to the excessive media contention often caused by traffic bursts in sensor networks. In [3], a sensor periodically probes the neighborhood wireless media whether it is idle or busy and thus measures the channel loading conditions. This procedure is not only energy-consuming but also fails to precisely measure the degree of congestion.

In this paper, ATE measures the congestion level of a node by exploiting both the buffer occupancy and the degree of media contention of a node. The buffer occupancy refers to the number of data packets in the buffer, and the degree of media contention corresponds to the channel activity level that can be measured by exploiting the retry field of MAC frame control.
4.2.1 Congestion Measurement Based on Buffer Occupancy

A buffer-based congestion level can be measured with almost no cost to the system. When the data packet arrives (either a transitive or a locally generated packet), each sensor node monitors its current buffer size \((BS_{cur})\) and calculates a running average value \((BS_{avg})\) using the Exponential Weighted Moving Average (EWMA) formula as follows,

\[
BS_{avg} = (1 - w_q) \times BS_{avg} + w_q \times BS_{cur}, 
\]

where, \(w_q\) is the moving average coefficient associated with older values, used to avoid random fluctuations; here, we set its value to 0.1. Note that \(BS_{avg}\) is increased when a node receives more packets than it can forward. More specifically, the buffer occupancy of a node is built up when the time gap between the packet service time and the packet inter-arrival time increases repeatedly. Hence, buffer-based congestion detection is capable of capturing long-term congestion in the network. However, it fails to detect instantaneous congestion due to media contention. Note here that our measurement method avoids periodic sampling of the transmitted frames, the node can quantify the degree of contention in the channel. Let, \(X_k (k = 0, 1)\) denote the numbers of frames whose retry field is \(k\). Then, the degree of wireless media contention \(deg(MC)\) can be expressed as the fraction of retransmitted frames, which is calculated as follows,

\[
deg(MC) = \frac{X_1}{X_0 + X_1}, 
\]

Note that \(deg(MC)\) may also be referred to as the probability that the first transmission attempt will be unsuccessful. Since unsuccessful transmission attempts may be induced by collisions as well as channel errors, the above metric can more accurately capture the congestive state of the network. Moreover, it can be measured online and requires no additional overhead except some computations. Finally, we take recent measurement of \(deg(MC)_{cur}\) values for every \(P\) packets and compute the runtime adaptive estimation reflecting the network dynamics. Again, the estimation is performed using a moving average as was done in Eq. (3):

\[
deg(MC)_{avg} = (1 - w_c) \times deg(MC)_{avg} + w_c \times deg(MC)_{cur}, 
\]

where, \(w_c\) is the moving average coefficient with value 0.1.

4.2.2 Congestion Measurement Based on Degree of Media Contention

Wireless media are shared by neighborhood contendong nodes. As the number of (event-detecting) source or transmitting nodes in the surrounding media (including hidden/exposed nodes) increases, the degree of media contention also increases. A traditional method for measuring this is to allow sensor nodes to periodically determine whether the media is busy or idle during the carrier sensing period of CSMA/CA MAC. For example, CODA and TARA nodes perform periodic sampling and calculates the exponential average of several samples to determine the channel loading status.

Our measurement method avoids periodic sampling of wireless media in order to improve energy-efficiency. It exploits the retry field of a CSMA/CA (e.g., IEEE 802.11 DCF or IEEE 802.15.4) MAC frame control to measure the congestion due to media contention. Note here that our measurement method does not depend on any specific MAC protocol, provided that the MAC employs a retransmission mechanism. We observe that there is a correlation between the frequency of retransmitting frames and the collision probability. As the network congestion level increases, the number of retransmissions is also likely to increase. Figure 5 shows the format of the two-byte frame control, where the single-bit retry field is used to indicate whether a data or control frame is transmitted for the first time or is a retransmission (0 or 1). While transmitting a frame, an ATE node \(i \in N\) inspects the value of retry field in the MAC header. By counting the frequencies of 0 and 1 values in the retry field
level. Note that this is the desired state where the nodes are adequately loaded and maintain a congestion-free data delivery environment. In this case, an ALERT message is sent to all the single-hop upstream nodes so that they do not further increase their packet loading rates to this downstream node. Our traffic engineering algorithm will aim to maintain the system in this state. The last and most undesirable state is the HIGH congestive state, which is detected whenever either one of the running average values crosses its upper threshold. At this stage, the congestive node sends a congestion notification message CNGST to the upstream node that is creating the highest proportion of the nodes’ aggregated traffic load. If that upstream node is also congested, it repeats the process until the CNGST message reaches a node with a MEDIUM or LOW congestion level. Traffic engineering actions (to be presented soon) will then be activated at that node.

One important point to note here is that the above multi-level congestion status detection and notifications allow ATE nodes to act in the most suitable way to alleviate congestion. Another unexpected situation may arise even if the congestion is not detected by a node or no notification message is received, i.e., link quality degradation. This may occur due to link asymmetry, or the presence of obstacles, to which the congestion detection mechanism may not be quickly responsive. In order to combat this problem, each ATE node estimates forwarding link quality.

4.3 Link Quality Estimation

A number of link quality metrics are discussed in the literature: expected transmission count (ETX) [22], expected transmission time (ETT) [23], small and large sized packet pair [24], etc. However, none of them is suitable for sensor networks since they use either special probe packets or periodic broadcast packets to quantify the metric values, which are not energy efficient. Since our goal is to determine whether a link can sustain the traffic load assigned to it (i.e., whether or not the flow conservation constraint Eq. (1) is maintained), we are interested in expressing the link quality in terms of its average successful packet delivery rate, namely the packet success rate \( PSR = \frac{\text{SP}}{\text{PP}} \), where \( \text{SP} \) is the average duration of successful packet deliveries. It is reasonable to estimate the metric value online during packet transfer over the downstream links, without considering the overhead of probe packets. Each ATE node \( i \in N \) measures the \( \text{SP}_l \), \( l \in L_i^d \) values using the Weighted Average Success Period (WASP) method\(^\text{††}\) that works as follows. WASP measures the instantaneous duration of each successful packet transfer, \( SP(n) \), which is the time duration from the \( n \)th packet transmission time to the time of successful transfer of last bit of the \((n+1)\)th packet and calculates the average value for the last \( P \) packets using Eq. (6),

\[
\text{SP}_l = \frac{\sum_{n=1}^{P} SP(n) \times w_n}{\sum_{n=1}^{P} w_n},
\]

where,

\[ w_n = 1 - \frac{n - P/2}{P/2 + 1}, \quad P/2 < n \leq P. \]

For, \( P = 8 \), this gives weights of 1, 1, 1, 1, 0.8, 0.6, 0.4 and 0.2 for \( w_1 \) through \( w_8 \), respectively.

Note that the sensitivity of the calculated success rate \( PSR_l = \frac{1}{\text{SP}_l} \) of link \( l \in L_i^d \) depends on the value of \( P \). In practice, a value of \( P = 8 \), with the most recent four samples equally weighted, appears to be a lower bound that still achieves a reasonable balance between resilience to link variations and responding quickly to real changes in the network conditions.

4.4 Autonomous Traffic Engineering Algorithm

Recall that the EBMGF algorithm chooses the next-hop nodes having higher residual energies and providing with maximal progress of data packets toward the sink. As long as the network operates with light traffic load, EBMGF (or its variant) alone may be expected to provide energy-balanced data delivery to the sink. However, it deliberately fails to mitigate the effects caused by bursts of traffic, link-quality fluctuations, link asymmetry, etc. At this stage, the traffic engineering algorithm comes into action and diverts traffic toward under-utilized sleeping nodes or better quality links. More precisely, the ATE algorithm may alter the next-hop addresses (of packets) chosen by EBMGF.

As shown in Fig. 6, the ATE algorithm uses the estimated \( PSR, BS_{avg} \) and \( deg(MC)_{avg} \) values and control message (if there is any) from the neighbor nodes as input and takes the appropriate control action(s). If it detects congestion, the algorithm sends the CNGST notification message to the upstream node that injects the highest traffic (lines 4–6 in Algorithm 2 of Fig. 7); or if it receives the CNGST message from any of its downstreams \( j \in N_i^d \), it repeats the

\( \text{††} \)Nodes can keep track of the incoming traffic volume from each upstream node and store this data in a separate field of the neighbor table, a data structure that is already maintained by most sensor networks.

\( \text{†††} \)WASP is very similar to the WLAI method [25], which measures packet loss intervals for TCP congestion control.
above procedure if it is also congested (lines 7–11), otherwise, it becomes the traffic distributor. It then diverts traffic from \( j \in N_i^d \) to an idle downstream node \( k \in N_i^d \) (lines 12–16). It is highly expected that, the above traffic diversion would decrease media contention in the congested area and thus the congestive state would soon be disappeared. If the congestion still remains, data traffic from other source nodes will be diverted to a detour path following the same procedure as stated above. Moreover, when a node receives an ALERT message, it remains silent and defers any further increase in the packet forwarding rate to that downstream node (lines 17–18). That is, the system wants a node and its associated links to carry moderate and tolerable traffic loads, trying to avoid congestive states proactively. Lastly, if the estimated \( PS_R_l \) value of any link \( l \in L_i^d \) falls below the tolerable level (e.g., 90% success rate), the ATE solves the problem locally, i.e., node \( i \) initiates a local traffic diversion (lines 19–22), incurring less overhead to the system.

The selection of a new downstream node \( k \in N_i^d \) for traffic diversion in ATE is a great design challenge, since a random selection may not be able to divert the excessive traffic from the hot spot. Thanks to the location-aware system, the traffic distributor node can easily locate a new neighbor \( k \), which is geographically separated from the congested node \( j \). Note also that ATE uses a passive wake-up mechanism for nodes in order to achieve an ultra-low energy overhead (see Sect. 4.5).

4.5 Node Wakeup Mechanism

The advent of a remotely activated switch (RAS) [26] allows sensor radios to be active only when they are needed to receive or transmit data packets, remaining sleeping at all other times. By sending an RF signal with the designated paging sequence (e.g., corresponding to the node ID) to the intended RAS-equipped receiver node, the latter can be woken up\(^\ddagger\). This completely eliminates unnecessary energy consumption due to listening. We named this Selective Wake-up (SW) energy consumption scheme, and its state transition diagram is shown in Fig. 8. Another advantage of using this scheme is that the maintenance of the RAS and the RF transmitter consumes very little energy.

4.6 ATE Operational Overhead

The excellent performance of ATE can not be achieved free of cost. The system incurs computation costs for running the EBMGF and ATE algorithms. The EBMGF algorithm is executed once for each packet, but the frequency of the ATE algorithm dynamically varies based on the congestive states of the network. However, the computations are not complex, and the comparisons and algebraic calculations may consume only a trivial amount of energy. The dominant source of energy consumption is the sensor radio module. ATE requires to send congestion notification messages towards upstream direction. However, ATE control messages travel just adequate number of hops to find a traffic distributor, without over-utilizing resources. Furthermore, our consideration of location-aware sensor nodes (either via GPS or self-localization techniques) has some extra costs. Thanks to the recent advancements in the design of extremely low power GPS receivers for sensor motes [27] and light-weight distributed self-localization algorithms [15], and many others therein, which greatly increase the applicability of location-aware systems at reduced cost. Moreover, since ATE uses hop-by-hop data forwarding, end-to-end path discovery and maintenance packets are not required. As a consequence, ATE incurs slightly less energy overhead than that utilized by TARA.

5. Performance Evaluation

5.1 Simulation Model and Method

We have evaluated the performance of ATE using simulation experiments conducted on NS-2 [28] and compared the

\(^\ddagger\)The RF signal may be sent by a transmitter wishing to transmit data to a receiver node or by local circuitry when local sensors detect some phenomena or event.
Table 2  Configuration of parameters.

<table>
<thead>
<tr>
<th>Deployment</th>
<th>Area size</th>
<th>2000 m × 2000 m</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Deployment type</td>
<td>Random</td>
</tr>
<tr>
<td></td>
<td>Network architecture</td>
<td>Homogeneous Flat</td>
</tr>
<tr>
<td></td>
<td>Number of nodes</td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td>Sink</td>
<td>(1000, 1000)</td>
</tr>
<tr>
<td></td>
<td>Initial node energy</td>
<td>100 Joule</td>
</tr>
<tr>
<td></td>
<td>Buffer size</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Sources in one event</td>
<td>15 Nodes</td>
</tr>
<tr>
<td></td>
<td>Radio range</td>
<td>100 m</td>
</tr>
<tr>
<td></td>
<td>Link layer trans. rate</td>
<td>512 kbps</td>
</tr>
<tr>
<td></td>
<td>Transmit power</td>
<td>$7.214 \times 10^{-3}$ Watt</td>
</tr>
<tr>
<td></td>
<td>Rev. signal threshold</td>
<td>$3.65209 \times 10^{-10}$ Watt</td>
</tr>
<tr>
<td></td>
<td>PHY error model</td>
<td>Uniform random</td>
</tr>
<tr>
<td>Task</td>
<td>Application Type</td>
<td>Event-driven</td>
</tr>
<tr>
<td></td>
<td>Packet size</td>
<td>64 Byte</td>
</tr>
<tr>
<td>MAC</td>
<td>Traffic type</td>
<td>CBR</td>
</tr>
<tr>
<td></td>
<td>IEEE 802.11 DCF</td>
<td></td>
</tr>
<tr>
<td>ATE</td>
<td>$BS_{low}, BS_{high}$</td>
<td>6, 24</td>
</tr>
<tr>
<td></td>
<td>$wp, wc$</td>
<td>70, 30</td>
</tr>
<tr>
<td></td>
<td>$deg(MC)_{low}$</td>
<td>0.18</td>
</tr>
<tr>
<td></td>
<td>$deg(MC)_{high}$</td>
<td>0.50</td>
</tr>
<tr>
<td>Simulation</td>
<td>Time</td>
<td>200 seconds</td>
</tr>
</tbody>
</table>

Table 3  Events and bursts description.

<table>
<thead>
<tr>
<th>Event 1</th>
<th>Event 2</th>
<th>Event 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>burst 1</td>
<td>10–40 s</td>
<td>20–50 s</td>
</tr>
<tr>
<td>burst 2</td>
<td>90–120 s</td>
<td>100–130 s</td>
</tr>
</tbody>
</table>

results with those of the CODA [3] and TARA[1] systems. The configuration of the simulation environment parameters is listed in Table 2. The bursts of data traffic from three randomly chosen events, listed in Table 3, are considered in the performance studies. For each data point in the graphs, we average the results of ten simulation runs. Moreover, for the sake of fair comparison, we have used SW energy consumption scheme in all the three approaches.

5.2 Simulation Results

5.2.1 Fidelity Index

The fidelity index is defined as the ratio of the number of packets received by the sink per unit time to the total number of packets generated per unit time by all source nodes in the network. Therefore, a data delivery scheme, that provides a higher end-to-end throughput, and reduces the packet latency and loss ratio, will have a higher fidelity index value. As shown in Fig. 9(a), ATE provides a much higher fidelity index than those of TARA and CODA. An improvement of as much as 18.3% over that of TARA is observed at high error rates. This is mainly due to ATE’s multi-level congestion detection and timely control actions. Figure 10(a) shows that ATE is more robust to node failures while still maintaining the fidelity level. This is mainly because of ATE’s localized and low overhead operation nature. On the other hand, TARA’s shortcomings stem from the fact that failure
5.2.2 Energy Efficiency ($\eta$)

Energy efficiency is measured as the ratio of the total amount of energy dissipated by all source and forwarder nodes to the number of packets received by the sink. Therefore, in this paper, the energy efficiency is represented by the average amount of energy expended for each successful packet reception. Note also that the higher is the $\eta$ value, the lower is the efficiency, and vice-versa.

5.2.3 Standard Deviation of Energy ($\sigma$)

The standard deviation of energy defines the average variance between the residual energy levels on all nodes and is measured by Eq. (7),

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (e_i^{res} - \mu_{res})^2},$$

where, $e_i^{res}$ and $\mu_{res}$ are, respectively, the residual energy of node $i$ and the mean residual energy for all nodes. Therefore, the value of $\sigma$ indicates how well the energy consumption is distributed among the sensor nodes. The smaller is the value, the better is the capability of ATE to balance the energy consumption.

As shown in Fig. 9(b) and Fig. 10(b), ATE is energy efficient compared with other methods. This is due to its efficient avoidance of congestive areas and thereby reducing the number of packet drops as well as the number of retransmissions at each intermediary hops, thus decreasing the energy cost of each received packet. Furthermore, ATE reduces the standard deviation of energy consumption by using the residual energy of forwarding nodes as one part of the routing metric and by distributing the total traffic load over spatially-separated nodes, thus increasing the network lifetime.

5.2.4 Operational Energy Overhead

Considering that computation costs (in terms of energy) are trivial compared to those of radio communications, we measure the energy consumption overheads of BEACON and congestion notification messages for ATE, as well as back-pressure and route construction messages for TARA and CODA. Even though these control packets are very small in size, the corresponding total workload and energy consumption increase with the number of such control packets in the network.

As ATE does not need to broadcast the route request messages network wide, each ATE node periodically broadcasts the BEACON messages to its single-hop neighbors. This process incurs less overhead than that of the other systems, as shown in Fig. 9(c). However, Fig. 10(c) shows that the gap between the overheads of ATE and TARA increases rapidly with node failure rate due to TARA’s overly frequent route initiations.
We measure the average end-to-end throughput (i.e., bytes received by the sink per unit time), packet delay and loss ratio during the data dissemination period and find that ATE can achieve performance improvements of 21.34%, 14.72% and 19.47%, respectively, over those of TARA, as shown in Table 4. Simulation runs are carried out for BER = 10^{-3} with node failure rate = 2.

### 5.3 Performance Study of ATE Components

In this subsection, we separately evaluate the performances of ATE components grouped into three different categories—ATE with no congestion control actions (ATE-NoCC) that includes only EBMGF algorithm (Sect. 3.2) and multipath traffic splitting policy (Sect. 4.1); ATE without link quality consideration (ATE-NoLQ), i.e., excepting only the estimation of PSR (Sect. 4.3) and corresponding actions in ATE algorithm; and, ATE with all components having full congestion control facilities (ATE). For this evaluation, we compare the above component groups in terms of only the fidelity index value achieved by them for increasing traffic loads, bit error rates and node failure rates as shown in Figs. 11(a), 11(b) and 11(c), respectively. This study facilitates us to clarify the effectiveness of various components of ATE, particularly in different network conditions.

To get the graph point values of Fig. 11(a), we execute simulation runs for source data traffic from increasing number of events (since network traffic load increases with the number of events). The durations of the events are time-overlapped in the same way as in burst 1 of Table 3; the event locations are randomly generated and the average result of 15 simulation runs are taken for the graph points. Therefore, variations among the obtained individual results occur mainly due to randomness of topology.

From the graphs of Fig. 11, we observe that ATE-NoCC, using only multipath data forwarding (with no congestion control actions) toward the sink node, can not achieve an acceptable level of fidelity index value. As expected theoretically, the performance of ATE-NoCC is greatly decreased at higher traffic loads and node failure rates. Through careful observation to the details of our simulation results, we notice that, in ATE-NoCC, excessive packet collisions occur in the bottleneck areas developing more intense funneling effect, which in turn quickly raises up the node queue sizes and thereby causes many packet drops. In addition to that, the number of retransmissions required for a packet at each hop increases with the collisions, prolonging the per-hop packet delay and thereby the end-to-end packet delivery delays. Thus, the packet reception rate at the sink drops much below an acceptance level. We also observe that the consideration of link qualities in traffic engineering actions in ATE has noticeable performance improvements over ATE-NoLQ, particularly for higher bit error rates and node failure rates. For instance, 0.72% to 7.4% improvement in Fig. 11(b) and 0.13% to 12.09% improve-
ment in Fig. 11(c) have been observed in terms of fidelity index value, for increasing bit error rates and node failure rates, respectively. The above results are realized by the fact that end-to-end data delivery performance deteriorates when sensor nodes fail suddenly or link qualities degrades, i.e., PSR value decreases. Since ATE redirects traffic toward a new path upon falling of PSR value below a certain threshold, it can keep the achieved fidelity index value at a high level; whereas, ATE-NoLQ can not, offering reduced performance.

5.4 Discussion

As our autonomous traffic engineering framework learns the neighbor’s (incipient) congestive state and residual energy levels in advance of time, its routing and traffic engineering functions effectively distributes the traffic loads over less-congestive and energy-rich forwarder nodes, implementing a load and energy-balanced data dissemination framework. Results reveal that ATE’s dynamic traffic redirection efficiently deals with the new types of congestion that are an artifact of the funneling effect and a product of increasing workload. Therefore, the network nodes are not overshooted with high traffic as well as the standard deviation of node’s residual energy levels found to be minimum.

Furthermore, simulation results show that ATE can efficiently cater for the application requirements with various network conditions, i.e., different combinations of traffic loads, bit error rates and sensor node failure rates. As a result, ATE can significantly improve the effective capacity of a sensor network in terms of number of flows meeting the fidelity requirement with reduced overhead, under versatile network environments. The developed ATE framework is expected to work with good performance in a large number of emerging WSN applications, particularly in which comparatively high-rate data delivery is demanded. For example, medical applications, object tracking applications, industrial process control and so on. However, it may not be suitable for applications that emphasize only on very low rate traffic flows, for instance, agricultural applications.

The rather pessimistic model of using the (strict) selective traffic splitting policy (Sect. 4.1) for multipath data forwarding in the current work provides with the first step research. Some dynamic selection techniques can be used to increase the robustness of ATE. The main limitation of this paper is related to a lack of sufficient understanding about the dynamics of threshold values. For example, \( \text{deg}(MC)_{\text{low}} \) and \( \text{deg}(MC)_{\text{high}} \) values were determined through numerous simulation experiments. If we could build an analytical model for them, we would be able to dynamically select the optimal values to adapt to different situations. This is a complicated and challenging task that we have left for future work. Also, ATE components are designed for single-sink network model. If we can redesign EBMGF and ATE algorithms suitable for multi-sink network model, a significant performance improvement is expected to be achieved.

6. Conclusions

In this paper, we design and evaluate an online traffic engineering tool for a randomly deployed homogeneous wireless sensor network that can circumvent adverse effects of network congestion and channel conditions. Traffic engineering actions are taken dynamically (on-demand) and they are driven by the inference of multi-level congestive states and the measured packet success rate values of forwarding links. When congestion appears, excessive packets are dynamically scattered to multiple paths consisting of idle or underloaded nodes. Therefore, as depicted in the simulation results, ATE can effectively alleviate congestion through bypassing the hot spots and meet the fidelity requirements by improving the overall throughput, latency and loss ratio.

Acknowledgment

We would like to pay our highest level of gratitude and sincere thanks to the anonymous reviewers for their expert opinions and useful feedbacks, which helped a lot to enrich the quality of this paper. This research was partially supported by the MKE, Korea, under the ITRC support program supervised by the NIPA (NIPA-2010-(C1090-1021-0003)) and by a grant from the Kyung Hee University in 2010 (KHU-20100189). Dr. CS Hong is the corresponding author.

References


Appendix: Threshold Values

Buffer Size Measurement Thresholds. The calculation method of average buffer size using EWMA in Eq. (3) can be imagined as a low-pass filter. Thus, the average queue size does not increase significantly due to short bursts of traffic or temporary congestion in the network. Therefore, a higher value of $BS_{low}$ is desired in order to maintain the link utilization at an acceptably high level [29]. For an 80% link utilization and a node buffer size of $BS = 30$, using an M/M/1/K queuing model in a high traffic burst environment, we find that the optimal value for $BS_{low}$ is equal to 6. On the other hand, the optimal value for $BS_{high}$ depends not only on the node buffer size $BS$, but also on the maximum allowable average delay for a packet from the intermediary nodes. According to the rule-of-thumb used in [29], [30], ATE sets its value to $4 \times BS_{low}$, i.e., equal to 24. Note also that the sensitivity of the measured average buffer size $BS_{avg}$ greatly depends on the moving average weight $w_q$, since it determines the time-constant of the low-pass filter. If $w_q$ is set to a large value, the averaging procedure will not filter out transient congestion in the network. Assume that the queue is initially empty and increases from 0 to $P$ packets over $P$ packet arrivals, at which time $BS_{avg}$ can be calculated as follows [29],

$$BS_{avg} = \sum_{p=1}^{P} pw_q(1-w_q)^{P-p} = P + 1 + \left(\frac{(1-w_q)^{P+1} - 1}{w_q}\right)$$

For the given $BS_{low} = 6$ and $P = 20$ values, it is necessary to choose $w_q \geq 0.08$ to satisfy the condition, $BS_{avg} > BS_{low}$.

Degree of Media Contention Measurement Thresholds. Recall that, in Eq. (4), the degree of wireless media contention $deg(MC)$ is calculated as the ratio of the retransmission frames over the total number of transmission attempts. Choosing the optimal values of the lower and upper thresholds for the measured average value $deg(MC)_{avg}$ is a system performance tuning issue. Since IEEE 802.11 DCF allows a node to retransmit a packet until the retry count reaches $MaxRetryLimit$ (default value = 7), it would be very energy inefficient to set the $deg(MC)_{high}$ value at a much higher
level. It also depends on how much contention a node would tolerate before signaling a congestive state. On the other hand, according to the probability theory, a higher retry limit value increases the probability of successful packet delivery. Similarly, setting much lower value for $\text{deg}(MC)_{\text{low}}$ may decrease network utilization. Note here that analyzing the most optimal values for the above thresholds is out of the scope of this work. We have carried out extensive simulations with all possible values and determined that setting $\text{deg}(MC)_{\text{low}} = 0.18$ and $\text{deg}(MC)_{\text{high}} = 0.50$ gives better results in our simulation environment. However, we did not provide results using other values.

Md. Abdur Razzaque received the B.S. degree in Applied Physics and Electronics and M.S. degree in Computer Science from the University of Dhaka, Bangladesh in 1997 and 1999, respectively. He obtained his Ph.D. degree in Computer Engineering from Kyung Hee University, South Korea in August, 2009. He had worked as an Assistant Professor in the Department of Computer Science and Information Technology of Islamic University of Technology (IUT), Gazipur, Bangladesh. He is an Assistant Professor (now on leave) in the Department of Computer Science and Engineering, University of Dhaka, Bangladesh. He is now working as a research professor, College of Electronics and Information, Kyung Hee University, South Korea. His research interest is in the area of modeling, analysis and optimization of communication protocols and architectures for wireless sensor networks, distributed systems, etc. He has published a good number of papers in international conferences and journals. He is a member of IEEE, IEEE Computer Society and KIPS.

Choong Seon Hong received his B.S. and M.S. degrees in electronic engineering from Kyung Hee University, Seoul, Korea, in 1983, 1985, respectively. In 1988 he joined KT, where he worked on Broadband Networks as a member of the technical staff. From September 1993, he joined Keio University, Japan. He received the Ph.D. degree at Keio University in March 1997. He had worked for the Telecommunications Network Lab, KT as a senior member of technical staff and as a director of the networking research team until August 1999. Since September 1999, he has been working as a professor of the School of Electronics and Information, Kyung Hee University. He has served as a Program Committee Member and an Organizing Committee Member for International conferences such as NOMS, IM, APNOMS, E2EMON, CCNC, ADSN, ICPP, DIM, WISA, BeN and TINA. His research interests include ad hoc networks, network security and network management. He is a member of IEEE, IPSJ, KIPS, KICS and KISS.

Sungwon Lee received his B.S. and Ph.D. degrees from Kyung Hee University, Korea. He is a professor of the Computer Engineering Department at Kyung Hee University, South Korea. Dr. Lee was a senior engineer of Telecommunications and Networks Division at Samsung Electronics Inc. during 1999 to 2008. He is an editor of the Journal of Korean Institute of Information Scientists and Engineers: Computing Practices and Letters.