An Optimal Resource Control Scheme under Fidelity and Energy Constraints in Sensor Networks

Jaewon Kang and Badri Nath
DATAMAN Mobile Computing Lab.
Computer Science, Rutgers University
110 Frelinghuysen Road, Piscataway, NJ 08854-8019
Tel: 732-445-2706 Fax: 732-445-6722
{*jwkang,badri*}*@cs.rutgers.edu*

Yanyong Zhang
WINLAB
Electrical & Computer Engineering, Rutgers University
94 Brett Road, Piscataway, NJ 08854-8058
Tel: 732-445-0608 Fax: 732-445-0593
yyzhang@winlab.rutgers.edu

***Abstract*— To control congestion, either the traffic from sources should be "reduced" (traffic controlling) or the available resources should be "increased" (resource controlling). Compared to the wired and other wireless counterparts, wireless sensor networks usually have elastic resource availability, and the applications require a certain level of throughput called *fidelity*. As a result, resource control strategies can not only alleviate congestion but also ensure the required fidelity level during congestion by accommodating higher incoming traffic.**

In this paper, we first attempt to formally define the resource control framework that adjusts the resource provisioning at the hotspot nodes during congestion. In an effort to find the optimal resource control under the fidelity and energy constraints, we present a resource increase and decrease algorithm called *Early Increase/Early Decrease* (EIED) that tries to adjust the effective channel capacity quickly to suit the incoming traffic volume in an energy-efficient manner, thereby increasing the fidelity level observed by the application. Under the energy-constrained optimization, we prove this algorithm incurs the lowest overhead of energy consumption for the given fidelity level that is required by the application. We also prove that the EIED algorithm performed in a distributed manner also lowers the energy consumption per packet at an end-to-end level. The effectiveness of the EIED algorithm is verified by simulations based on realistic sensor network configurations.

***Index Terms*— Congestion control, Sensor networks, Resource control, Fidelity, Energy efficiency**

I. Introduction

As the technologies in MEMS, processor design, and wireless communication advance, a wide range of remote monitoring applications using networks of sensor nodes receive much attention recently. These networks will usually be left unattended, with each individual node sensing the physical environment such as the behavior of endangered species, fires in a forest, or traffic conditions on a busy highway. The sensed events are processed locally if necessary and reported to the sinks through multihop wireless communication. These networks operate with a low reporting rate before target events (e.g., fires, abnormal phenomenon, etc) occur. As soon as these events are sensed, however, a high reporting rate is necessary so that the events are accurately portrayed by the reported data at the sinks. As a result, sensor networks alternate between periods with a very low traffic volume (referred to as *dormant state*) and periods with a high traffic volume (referred to as *crisis state*). In most of the sensor applications, dormant state dominates the network lifetime, during which the network usually only maintains a small amount of available resources, such as battery power in active nodes, in order to extend the network lifetime.

During a crisis state, the traffic volume often exceeds the amount of available resources such as outgoing link capacity [20], [23], thus resulting in congestion at various nodes. As a result, the network may enter an unstable state in which packets are randomly dropped, a severe case of which is called *congestion collapse*. Intuitively, we have two options to alleviate congestion: throttling the incoming traffic volume (referred to as *traffic controlling*) and increasing the available resources (referred to as *resource controlling*).

Although traffic control strategies are effective in traditional wired networks and are also suggested in some sensor network scenarios [20], [23], [30], they are unsuitable for our purpose for two main reasons. Firstly, reducing the traffic during a crisis state is unacceptable. The data during a crisis state are of great value, often critical, to the applications such as earthquake or fire monitoring systems. Failure to ship some of these data to the sink can hurt the applications' accurate monitoring, called *fidelity*, and defeat the very reason why the sensor network was deployed in the first place. Secondly, increasing resource provisioning during congestion is easier in sensor networks due to its elastic availability of resources. There is usually an abundance of resources in sensor networks (unlike its wired or other wireless counterparts) because these networks are usually densely deployed in order to achieve a reasonable network lifetime [1], [26], [27]. Therefore, a natural way of managing these limited resources (such as channel capacity, remaining energy, transmission power, etc.) is to conserve as much resources as possible during a dormant state while exploiting them wisely during crises.

Even though congestion control is an important issue for sensor networks, it has received little attention until recently and most of recent studies focus on controlling traffic to alleviate congestion [20], [23], [30]. The question, "*How should network resources such as wireless channel capacity be systematically controlled in conjunction with the congestion level under fidelity and energy constraints?*", has not been studied rigorously even though resource control strategies

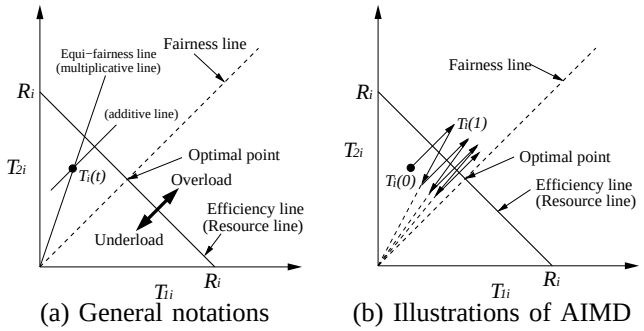
have the potential of minimizing energy consumption during a dormant state as well as accommodating the traffic surge during a crisis state, thus avoiding congestion and satisfying the fidelity requirement. In this paper, we set out to answer this challenging question.

The remainder of this paper is organized as follows. In the next section, we explain the congestion avoidance performed by the famous traffic increase and decrease algorithm, AIMD. We present the related work in Section III. In Section IV, we formulate the goals of resource controlling along with the performance metrics for the potential resource controls. In Section V, we delve into finding the systematic ways of adjusting resources in the unconstrained sensor network environment. In Section VI, we propose a resource increase and decrease algorithm called *Early Increase/Early Decrease* (EIED) that incurs the lowest overhead under fidelity and energy constraints. In Section VII, the EIED algorithm is verified by simulation. Section VIII describes the limitations of the EIED algorithm and concludes this paper.

II. Traffic Increase and Decrease Algorithm for Congestion Avoidance

Controlling traffic in the context of congestion avoidance has been extensively studied mostly through the active queue management (AQM) in wired networks. Two main goals of traffic control strategy are *high resource utilization* and *fairness*. To achieve these goals, each node in the network measures its resource utilization state and feeds this information back to either its previous hop nodes (hop-by-hop strategy) or its sources of traffic (end-to-end strategy) who then adjust their outgoing traffic volume. If the incoming traffic is not throttled especially during congestion, the attainable throughput is not simply saturated but degraded, resulting in random packet drops, low resource utilization, and unfair resource allocation.

Traffic controlling in a two-flow case has been formally depicted as a vector representation by Chiu and Jain [4] shown in Figure 1(a). The horizontal and vertical axes represent the flow 1 and 2's offered traffic at node i, T_{1i} and T_{2i}, respectively. The point $T_i(t)$, whose vector representation is $< T_{1i}(t), T_{2i}(t) >$ and whose aggregate traffic volume is defined to be $(T_{1i}(t) + T_{2i}(t))$, indicates the incoming traffic volumes of the flow 1 and 2 into node i at time t. The efficiency line (referred to as *resource line* in this paper), whose effective resource amount is fixed to R_i, denotes those resource allocations that can most efficiently utilize the given resources, i.e. $T_{1i}(t) + T_{2i}(t) = R_i$. The traffic vector $< T_{1i}(t), T_{2i}(t) >$, where $T_{1i}(t) + T_{2i}(t) > R_i$ (the corresponding point $T_i(t)$ is above the resource line), denotes an *overloaded* state, where congestion occurs. At the same time, the traffic vector $< T_{1i}(t), T_{2i}(t) >$, where $T_{1i}(t) + T_{2i}(t) < R_i$ (the corresponding point $T_i(t)$ is below the resource line), denotes an *underloaded* state which we should avoid as well, when the two flows can potentially increase their traffic volumes. The dotted line, where $T_{1i}(t) = T_{2i}(t)$, includes all the fair resource allocations that give an equal amount of resource to both flows. The resource line and



(a) General notations (b) Illustrations of AIMD

Fig. 1. Additive Increase/Multiplicative Decrease (AIMD) in traffic control

fairness line intersect at the location $< \frac{R_i}{2}, \frac{R_i}{2} >$, which we call the optimal allocation, wherein the effective resource is efficiently and fairly distributed between two flows. Therefore, the goal of traffic controlling is to tune the offered traffic of all flows to approach the optimal point. An n-flow case can be easily realized by extending the 2-dimensional space to the n-dimensional space. In the n-flow case, the total offered traffic at node i at time t, denoted as $T_i(t)$, is represented as $\sum_{j=1}^{n} T_{ji}(t)$. Therefore, the total offered traffic at node i during the event period of D is $\int_0^D T_i(t)dt$, which is in turn $\int_0^D (\sum_{j=1}^{n} T_{ji}(t))dt$.

Depending on the relative position of the traffic vector point $T_i(t)$ with respect to the resource line, the traffic control component of node i sends a binary feedback to the source of each flow as shown below.

$$b_i(t) = \begin{cases} 0 & if\ T_i(t) < R_i \quad (underloaded) \\ 1 & if\ T_i(t) \geq R_i \quad (overloaded) \end{cases} \tag{1}$$

The feedback of 1 means that the network is overloaded and the source needs to reduce its traffic (the rate at which it pumps data); 0 means that the flow can increase its traffic. The source has several options in systematically decreasing/increasing its traffic after receiving the feedback. In [4], Chiu and Jain have proved that a simple additive increase and multiplicative decrease (AIMD) algorithm as shown in Equation 2 satisfies the sufficient conditions for convergence to a high utilization and fair state in a distributed manner, regardless of the starting state of the network:

$$T_i(t+1) = \begin{cases} T_i(t) + a_I & if\ b_i(t) = 0 \\ m_D T_i(t) & if\ b_i(t) = 1, \end{cases} \tag{2}$$

where a_I and m_D are constants and $a_I > 0$ and $0 < m_D < 1$. This algorithm has been extensively studied in the literature and adopted by some versions of TCP congestion window control algorithms [7], [8]. Figure 1(b) illustrates the temporal progression of the traffic vector point $T_i(t)$ that approaches the optimal point by AIMD.

III. Related Work

As mentioned earlier, most of prior work have been focused on the efficient usage of traffic control during congestion in sensor networks. A guideline of congestion control in sensor networks was first given by [21]. They suggest that congestion controlling must not only be based on the network capacity, but also on the fidelity level dictated by the applications. In

other words, the total data received from M reporting sensors, represented as $\sum_{i=1}^{M} b(S_i)$ where $b(S_i)$ is the bit rate of sensor *i*, should not exceed a certain fraction of the available channel capacity C_{total}, but also be high enough to satisfy the desired fidelity as shown in Equation 3:

$$C_{application} <= \sum_{i=1}^{M} b(S_i) <= \alpha C_{total}. \quad (3)$$

$C_{application}$ is the application-specific fidelity level. We can infer from Equation 3 that if applications require a high fidelity *temporarily* during a crisis state (i.e. if $C_{application}$ exceeds $\sum_{i=1}^{M} b(S_i)$) and $\sum_{i=1}^{M} b(S_i)$ cannot be increased due to the upper bound imposed by αC_{total}, then the available channel capacity C_{total} should be temporarily increased to allow an increased $\sum_{i=1}^{M} b(S_i)$, therefore to meet the fidelity requirement in Equation 3.

CODA [23] presents the first detailed study on congestion detection and avoidance in sensor networks. In CODA, as soon as a node detects congestion, it broadcasts a backpressure message upstream. An upstream node that receives the backpressure throttles its outgoing traffic by dropping packets or by forwarding the backpressure message upstream again. CODA also provides a method of controlling traffic in an end-to-end manner by requiring constant feedback (ACKs) from the sinks. Sankarasubramaniam, Akan and Akyildiz [20] have proposed an event-to-sink reliable transport (ESRT) protocol, which can serve as a congestion control protocol as well since congestion is considered to be the primary cause for unreliable delivery. In ESRT, the sink reduces the data rates of all sources (i) when the number of received packets within a window is below the desired level; or (ii) when an intermediate node reports possible buffer overflow by marking 1-bit CN field in the header of the delivered packet(s). Hull, Jamieson, and Balakrishnan [11] have studied three congestion control techniques: hop-by-hop traffic control, limiting source rate, and a prioritized medium access control (MAC). They show network congestion can be greatly alleviated when these three traffic control techniques operate in concert rather than in isolation. Ee and Bajcsy [6] proposed a distributed congestion control scheme based on hop-by-hop automatic repeat request (ARQ) in many-to-one routing scenario. To avoid congestion, motes are allowed to generate data at a rate network can handle. For this, each mote individually determines its local maximum transmission rate and divides the transmission rate by total number of upstream motes (children) to give a data generation rate of each upstream mote. The computed data generation rate is compared with the data generation rate propagated from its downstream node (parent). The smaller rate is propagated to its upstream nodes. Woo and Culler [24] proposed an adaptive traffic control scheme in which locally generated traffic and route-through traffic are assigned bandwidth proportionally to provide fairness among the flows with different path lengths, which also avoids congestion. Yi and Shakkottai [30] proposed a hop-by-hop congestion control scheme that allocates bandwidth to various users in a fair manner. They show a hop-by-hop traffic control scheme push-backs and spreads congestion over space, called spatial spreading, leading to scattered small peak loads. This ensures that the required buffer size are spatially spread.

Several multipath routing protocols [5], [9], [15], [22], [29] can increase the end-to-end channel capacity during congestion. They, however, are developed in the context of reliability, load balancing, and failure recovery rather than congestion control, and are not energy-aware.

IV. Problem Formulation

In this section, we attempt to formally define the framework of the resource control strategy. The framework of the resource control strategy differs from that of the traffic control strategy in several ways. Firstly, node i's effective resource capacity (the resource line in Figure 1(a)), which has been fixed to R_i in the traffic control framework, is now represented as a time-varying function, $R_i(t)$, due to its elastic availability. Secondly, fairness among competing flows is not realized by the resource control operation alone unless per-flow resource provisioning is implemented, which, however, incurs too much overhead. Thirdly, due to its centralized view of the incoming traffic load, the resource control operation can be performed locally without the need of feedback to other nodes. Fourthly, in a resource control strategy an increase operation is applied during the overloaded state and a decrease operation is applied during the underloaded state, which is opposite to the traffic control scheme. Finally, it is more costly to increase/decrease resource provisioning than to increase/decrease the outgoing traffic volume. For example, the cost of adjusting TCP's congestion window is much lower than the cost of adjusting channel capacity by changing transmission power or the number of active nodes in the sensor field.

Figure 2 shows two examples of the resource control policies. At time $t+1$, the aggregate incoming traffic volume $T_i(t+1)$ exceeds the effective resource capacity $R_i(t+1)$ at node i. The effective resource capacity is thus increased to $R_i(t+2)$ at time $t+2$ to accommodate the traffic surge, thereby alleviating congestion. The notations of $T_i(t)$ and $R_i(t)$ are used throughout this paper indicating the aggregate incoming traffic and the effective resource capacity of node i at time t, respectively. Therefore, the outgoing traffic volume of node i at time t is $min(T_i(t), R_i(t))$ if we ignore the queueing delay and processing time. In Figure 2(a), the effective resource capacity is over-provisioned at time $t+2$ to allow the potential traffic increase of flow 2. While this policy lowers the resource utilization and wastes more energy, it is resilient to the traffic fluctuation compared to the policy in Figure 2(b) which has $R_i(t+2) = T_i(t+1)$, and thus more stable.

Figure 2 illustrates stability and resource utilization for a resource control policy. The criteria for a resource control policy are listed below (*fidelity* and *energy efficiency* are more important than others):

- *Fidelity:* The main motivation for resource controlling is to ship as much incoming traffic as possible, at least above the required fidelity level F (in bits per unit of time), to one or more sinks during a crisis state, so that the delivered data can produce a meaningful view of the sensed event and subsequently entail necessary actions by

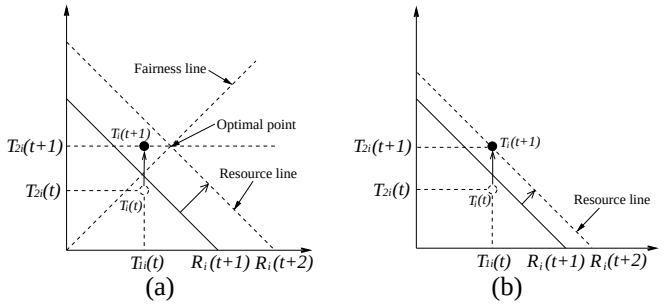


Fig. 2. Examples of resource increase during congestion

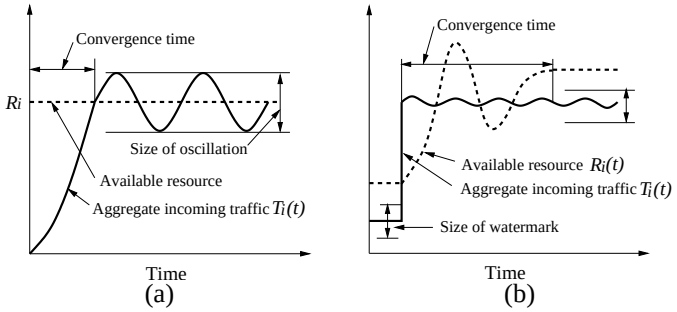


Fig. 3. Convergence time and oscillation of (a) traffic control and (b) resource control

the application that extracts the delivered data from the sinks. Therefore, the observed fidelity level of an event, denoted as F^{obs}, is formally defined to be the average volume of data received by m sinks from n reporting sources during the event period D and represented as follows:

$$F^{obs} = \frac{\int_0^D (\sum_{j=1}^{m} \sum_{i=1}^{n} F_{ij}(t))dt}{D}, \tag{4}$$

where $F_{ij}(t)$ indicates the data volume (in bits) delivered to sink j from source i at time t. F^{obs} should be greater than the required fidelity level F. Especially, we term the total delivered data to the sinks during the period of D, which is $\int_0^D (\sum_{j=1}^{m} \sum_{i=1}^{n} F_{ij}(t))dt$ in the above equation, *fidelity amount*.

- *Energy Efficiency:* Increasing the effective resource capacity can improve the quality of service observed by the application, but it may also increase the total energy consumption due to the higher data rate and the maintenance overhead of the increased resource provisioning. In addition, if the increased resource capacity still does not accommodate the incoming traffic, some of the traffic will still be discarded due to congestion, thereby nullifying the energy expended for receiving the traffic from the previous hop nodes. Therefore, the total energy consumption at node i during the event period of D, denoted as $\int_0^D E_i(t)dt$ where $E_i(t)$ is the energy consumed by node i at time t, includes (i) the energy E_{idle} to maintain the resource of $\int_0^D R_i(t)dt$ when node i is idle, (ii) the energy $E_{receive}$ to receive the incoming traffic volume of $\int_0^D T_i(t)dt$ from the neighbor nodes, and (iii) the energy $E_{transmit}$ to transmit the outgoing traffic volume of $\int_0^D min(T_i(t), R_i(t))dt$ to the next hops. $E_i(t)$ depends on the quantitative relation between $T_i(t)$ and $R_i(t)$. Let's assume the energy consumed to receive 1 bit is the same as the energy to forward 1 bit[1]. When node i is underloaded (i.e., $T_i(t) < R_i(t)$), $\int_0^D E_i(t)dt$ is the sum of E_{idle}, $E_{receive}$, and $E_{transmit}$, where $E_{receive}$ is equivalent to $E_{transmit}$ because all the incoming traffic are forwarded to the next hop nodes. When node i is optimally utilized (i.e., $T_i(t) = R_i(t)$), $\int_0^D E_i(t)dt$ is the sum of only $E_{receive}$ and $E_{transmit}$ because node i is not idle during the event period. When node i is overloaded (i.e., $T_i(t) > R_i(t)$), $\int_0^D E_i(t)dt$ is the sum of $E_{receive}$ and $E_{transmit}$, where $E_{receive}$ is greater than $E_{transmit}$ because some of the incoming traffic are dropped and not transmitted due to the lack of the available resources. During the periods when $T_i(t) > R_i(t)$ holds, node i not only waste energy receiving the incoming traffic that is ultimately dropped due to the lack of $R_i(t)$, but also nullifies the energy consumed for the dropped traffic to forward from the sources to node i by the network. In Section VI, the energy efficiency in conjunction with an arbitrary resource control is analyzed in detail.
- *Packet Energy Efficiency:* Packet energy efficiency, denoted as PE, indicates the average amount of energy consumed by the network to successfully forward a packet from a source to a sink. If n sources send the sensed events towards m sinks and l nodes are involved in delivering the packets during the event period of D, PE is calculated by dividing the total energy consumption in the network by the fidelity amount observed by m sinks as follows:

$$PE = \frac{\int_0^D (\sum_{i=1}^{l} E_i(t))dt}{\int_0^D (\sum_{j=1}^{m} \sum_{i=1}^{n} F_{ij}(t))dt} P, \tag{5}$$

where P is the average packet size (in bits).
- *Convergence Speed:* The convergence time in the traffic control strategy is defined as the time taken for the competing flows to reach an "equilibrium" in which their aggregate traffic volume oscillates around the effective resource capacity. In the case of traffic control, the effective resource capacity at node i is fixed to R_i as shown in Figure 3(a). In the resource control strategy, the effective resource capacity is elastic and it reaches an equilibrium when the aggregate incoming traffic oscillates between the upper and lower watermarks whose values are determined relative to $R_i(t)$ as shown in Figure 3(b). The convergence time is defined as the interval between the time when the aggregate incoming traffic hits the upper or lower watermark and the time when the equilibrium is reached. Therefore, the convergence speed depends on not only the resource control policy but also the size of watermark (i.e. gap between the upper and lower watermarks).
- *Stability*: While the stability of a traffic control policy

[1]Generally, the energy consumed during transmitting is greater than the energy consumed during receiving, which is assumed in our scheme shown in Section VI.

is governed by the oscillation size of the aggregate incoming traffic, the stability of a resource control strategy is measured by the frequency of the increase/decrease operations as a result of fluctuating incoming traffic. This is because the effective resource capacity remains constant without requiring any more increase/decrease operations after it reaches an equilibrium as shown in Figure 3(b). The degree of stability can be enhanced (i) by smoothing the incoming traffic using a low-pass filter that returns an exponential weighted moving average T_i^{avg} at node i as shown below:

$$T_i^{avg}(t+\delta) = (1-\alpha) * T_i^{avg}(t) + \alpha * T_i(t), \quad (6)$$

where δ and α are constants and $\delta > 0$ and $0 < \alpha < 1$, or (ii) by over-provisioning the effective resource capacity along with the larger watermark size to allow the rapid fluctuation of incoming traffic. Larger δ or smaller α in Equation 6 will increase the stability of a resource control policy. Increasing the stability of a resource control scheme, however, likely hurts the resource efficiency (or utilization) of the scheme.

- *Resource Efficiency*: The resource efficiency of node i at time t is measured by $\frac{T_i(t)}{R_i(t)}$. Please note that measuring resource efficiency is only meaningful when $T_i(t) < R_i(t)$. Overload ($T_i(t) > R_i(t)$) or underload ($T_i(t) < R_i(t)$) are generally not desirable, but some level of underload is needed to accommodate fluctuating incoming traffic, thereby improving stability as shown in Figure 2(a).

The goal of the resource control strategy is to systematically adjust the effective resource capacity available to a node, based on the incoming traffic volume, while satisfying the above criteria. In the next section, various resource control strategies are proposed and evaluated in terms of above criteria. The optimal resource control strategy under fidelity and energy efficiency requirements is explained in Section VI.

V. General Resource Increase and Decrease Algorithms

While the traffic control strategy employs a binary congestion feedback shown in Equation 1, a trinary congestion feedback is needed by the resource control strategy as follows:

$$b_i(t) = \begin{cases} 0 & if\ T_i(t) < R_i(t) - w_l \\ -1 & if\ R_i(t) - w_l \le T_i(t) \le R_i(t) - w_u \\ 1 & if\ T_i(t) > R_i(t) - w_u, \end{cases} \quad (7)$$

where w_u and w_l are positive constants with $w_l \ge w_u$. The upper and lower watermarks of node i at time t are expressed as $(R_i(t) - w_u)$ and $(R_i(t) - w_l)$, respectively. The term $(T_i(t) - R_i(t))$ denotes the congestion degree or the resource deficiency of node i at time t.

When node i's aggregate incoming traffic volume $T_i(t)$ falls below the lower watermark, the congestion feedback is set to 0 indicating the node's current resource capacity can be reduced. When the node's aggregate incoming traffic volume lies between the upper and lower watermarks, the congestion feedback function returns -1 indicating no resource control is needed. When the node's aggregate incoming traffic volume exceeds the upper watermark, the congestion feedback function returns 1 indicating the node's effective resource capacity should be increased. The upper and lower watermarks are already illustrated in Figure 3(b). When $R_i(t)$ is not accurately known at the time of a resource control operation, the trinary congestion feedback function $b_i(t)$ in Equation 7 that compares $T_i(t)$ with $R_i(t)$ cannot be performed. Therefore, $b_i(t)$ is inferred indirectly from other measurements such as packet queue occupancy, packet drop rate, and wireless channel loading, which, however, lowers the accuracy of $b_i(t)$. In this section, several feasible resource controls are discussed.

A. Feasible Resource Controls

Equation 8 shows a linear resource control that can be deployed at node i:

$$R_i(t+1) = \begin{cases} m_D R_i(t) + a_D & if\ \ b_i(t) = 0 \\ R_i(t) & if\ \ b_i(t) = -1 \\ m_I R_i(t) + a_I & if\ \ b_i(t) = 1, \end{cases} \quad (8)$$

where m_I, a_I, m_D, and a_D are constants and $m_I > 1$, $a_I > 0$, $0 < m_D < 1$, and $a_D < 0$. Neither increase nor decrease operation is performed when $b_i(t) = -1$. In this paper, we focus on the *additive* linear resource control, where the current effective resource capacity is increased or decreased by adding or subtracting some amount as shown in Equation 9. The multiplicative control has been used by the traffic control scheme to achieve fairness, which, however, is not the goal of the resource control. Also, depending on the resource type, the effective capacity of some resources fluctuate rapidly even when their nominal resource capacity is constant, making it difficulty to quantify the effective resource capacity. As a result, it is unrealistic to multiply the resource capacity. For example, the wireless channel capacity of a node changes rapidly due to noise, fading, and high path loss when it is deployed in a harsh environment. When the current effective resource capacity is difficult to quantify, the multiplicative increase/decrease operations cannot be performed. On the other hand, $T_i(t)$ can be easily quantified by measuring the incoming traffic volume (not the queue length). The incoming traffic, however, might be discarded due to the queue overflow when they are enqueued or due to high channel contention when they are sent out[2].

1) Additive Resource Controls: In additive resource controls, the effective resource capacity is adjusted by repeatedly adding or subtracting an amount of $a(t)$ to or from the existing $R_i(t)$ until $b_i(t)$ returns -1, as is shown in Equation 9. $a(t)$ is a time-varying positive real function that is not correlated to $R_i(t)$.

$$R_i(t+1) = \begin{cases} R_i(t) - a(t) & if\ \ b_i(t) = 0 \\ R_i(t) & if\ \ b_i(t) = -1 \\ R_i(t) + a(t) & if\ \ b_i(t) = 1 \end{cases} \quad (9)$$

The additive control in Equation 9 is different from the additive increase/decrease policy applied in the traffic control scheme

[2]In most MAC protocols, packets are dropped after several unsuccessful retransmissions.

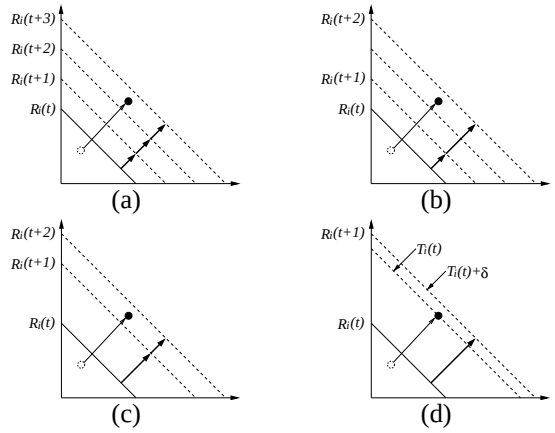


Fig. 4. Feasible resource control policies in a two-flow case as in Figure 2

in that the additive amount $a(t)$ is not constant but a time-varying function.

In this paper, we propose four variations of additive resource control policies based on how $a(t)$ is calculated. In each policy, we discuss the following aspects: (i) the amount of the adjusted resource capacity on each resource increase operation, (ii) the resulting resource capacity after n consecutive resource increase operations, (iii) the convergence time, and (iv) the average over-provisioned resource capacity after n consecutive increases. Since the convergence time of a resource control also depends on the watermark size, here it is expressed in the number of the consecutive resource increase operations needed until the resulting resource capacity exceeds the aggregate incoming traffic. We assume that the aggregate incoming traffic volume is rather stable during the resource control operation and R^{req} is the exact amount of the required resource capacity to alleviate congestion. Above four values can be calculated in the same way for the decrease operations.

- Constant Increase/Decrease: In this policy, the amount of resource change in each step is the same, i.e. $a(t+1) = a(t)$. If $a(0) = r$, $a(t+1)$ is expressed as follows:

$$a(t+1) = r. \tag{10}$$

  In this case, the amount of resource increase/decrease in each step is r and the total amount of resource increase after n consecutive steps is nr. Given that nr is greater than R^{req} to alleviate congestion, the convergence time is $\lceil \frac{R^{req}}{r} \rceil$. This policy is illustrated in Figure 4(a). In the figure, the average over-provisioned resource amount is $\frac{r}{2}$.
- Exponential Increase/Decrease: In this policy, $a(t)$ increases/decreases additively during the consecutive resource increase/decrease operations. However, $a(t)$ is reset to $a(0)$ when the aggregate incoming traffic, $T_i(t)$ falls again between the upper and lower watermarks, i.e. when $b_i(t) = -1$. $a(t+1)$ is shown as follows:

$$a(t+1) = \begin{cases} a(t) + r & if \;\; b_i(t+1) = b_i(t) \\ r & if \;\; b_i(t+1) \neq b_i(t), \end{cases} \tag{11}$$

  where $a(0) = r$. The amount of resource increase/decrease in the n_{th} step is nr and the total amount of resource increase after n consecutive steps is $\frac{n(n+1)}{2}r$. The convergence time is $\lceil \sqrt{\frac{2R^{req}}{r} - \frac{1}{4}} - \frac{1}{2} \rceil$. The average over-provisioned resource amount is $\frac{kr}{2}$, where k is the above convergence time. This policy is shown in Figure 4(b).
- Constant Increase/Decrease after Arbitrary Increase/Decrease: In this policy, a node initially adjusts the effective resource capacity by the amount of β and then fine-tunes the resource change using the Constant Increase/Decrease policy. The motivation of this policy is to reduce the convergence time. We have

$$a(t+1) = \begin{cases} r & if \;\; b_i(t+1) = b_i(t) \\ \beta & if \;\; b_i(t+1) \neq b_i(t), \end{cases} \tag{12}$$

  where $\beta \gg r$ and $a(0) = r$. Since we have $\beta \gg r$, the effective resource capacity is quickly increased or decreased at the beginning. The amount of resource increase in the n_{th} step is either r when $n > 1$ or β when $n = 1$. The amount of resource increase after n consecutive steps is $\beta + (n-1)r$. Therefore, the convergence time is either $\lceil \frac{R^{req}-\beta}{r} \rceil$ when $R^{req} > \beta$ or 1 when $R^{req} \leq \beta$. The average over-provisioned resource amount is either $\frac{r}{2}$ when $R^{req} > \beta$ or $\frac{\beta}{2}$ when $R^{req} \leq \beta$. This policy is shown in Figure 4(c).
- Multiplicative Increase/Decrease of $a(t)$: This policy employs

$$a(t+1) = \begin{cases} ma(t) & if \;\; b_i(t+1) = b_i(t) \\ r & if \;\; b_i(t+1) \neq b_i(t), \end{cases} \tag{13}$$

  where $a(0) = r$. In this policy, the amount of resource increase/decrease in the n_{th} resource control step is $m^{n-1}r$ and the total amount of resource increase after n consecutive steps is $\frac{m^n-1}{m-1}r$. The convergence time is $\lceil log_m(\frac{R^{req}}{r}(m-1)+1) \rceil$. The average over-provisioned resource amount is $\frac{m^{k-1}}{2}r$ where k is the convergence time.

2) Multiplicative Resource Control: As discussed above, the multiplicative resource control is possible only when the current resource capacity $R_i(t)$ is stable and easy to quantify because the amount of resource change cannot be calculated without knowing the current resource capacity. Specifically, we have

$$R_i(t+1) = \begin{cases} m_D R_i(t) & if \;\; b_i(t) = 0 \\ R_i(t) & if \;\; b_i(t) = -1 \\ m_I R_i(t) & if \;\; b_i(t) = 1, \end{cases} \tag{14}$$

where $m_I > 1$, $0 < m_D < 1$. The amount of resource increase in the n_{th} consecutive resource control step is $(m_I - 1)m_I^{n-1}R_i(0)$, where $R_i(0)$ is the effective resource capacity of node i when the increase operation is first performed. The total amount of resource increase after n consecutive steps is $(m_I^n - 1)R_i(0)$. The convergence time is $\lceil log_{m_I}(\frac{R^{req}}{R_i(0)} + 1) \rceil$. The average over-provisioned resource amount is $\frac{(m_I-1)m_I^{k-1}R_i(0)}{2}$ where k is the convergence time.

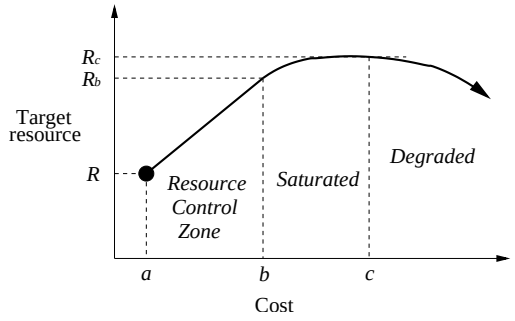


Fig. 5. Target resource graph with respect to cost

3) Fit Resource Control: If the resource capacity can be controlled quickly, then the effective resource capacity can be increased directly to the volume of $T_i(t)$:

$$R_i(t+1) = \begin{cases} T_i(t) + \delta & if \ \ b_i(t) = 0 \\ R_i(t) & if \ \ b_i(t) = -1 \\ T_i(t) + \delta & if \ \ b_i(t) = 1, \end{cases} \quad (15)$$

where δ is used to allow the fluctuation of incoming traffic, thereby improving the stability of this policy. The amount of resource increase/decrease in each resource control step is $|T(t) + \delta - R(t)|$ that is $(R^{req} + \delta)$ under our assumptions. The over-provisioned resource capacity is δ.

B. Discussions

Depending on the resource type, adjusting its capacity may not be achieved instantaneously. Even though a fast resource adjustment is possible, it may lead to instability when the incoming traffic volume is fluctuating. To increase the stability of a resource control strategy, the over-provisioned resource capacity should be large enough to accommodate the varying traffic, which, however, lowers the resource utilization. Low resource utilization leads to the waste of energy. In the next section, we present a resource control strategy that incurs the lowest overhead under fidelity and energy constraints.

VI. Early Increase/Early Decrease

In this section, we first characterize available resources in sensor networks. Next, we present an optimal resource control strategy under fidelity and energy constraints.

A. Resources in Sensor Networks

In sensor networks, various types of resources exist such as remaining energy, transmission power, channel capacity (or bandwidth), packet buffers in the queue, active nodes (i.e. whose radio is on.), and forwarding path(s) towards a sink. Resources in sensor networks are correlated in the sense that changing the provisioning of one type of resource affects the availability of other resources. For example, if the transmission power of a node increases, the remaining energy of that node is likely drained quickly and the channel capacity is also affected.

If other resources consumed as a result of adjusting a target resource are collectively termed *cost*, the availability graph of the target resource with respect to the cost is shown in Figure 5. We believe that a minimum resource capacity is required for target resource at the cost of a as shown in the figure. This minimum resource, denoted as R, is called the *default resource capacity*. For example, when a node's radio is turned on with some level of initial energy cost, its transmission power is set to the default level that provides a minimum coverage for the connectivity with neighbor nodes and a minimum channel capacity. This is also true when the target resource is the end-to-end capacity between a source and a sink which may be connected through multiple paths. The default resource capacity in this case is the end-to-end channel capacity when a single path is established.

As the cost increases, the capacity of a target resource is expected to first quickly increase until the cost reaches b where the target resource is increased to R_b. After this point, the target resource capacity is saturated and may even degrade later on. For example, when a node's transmission power level is too high, it will increase the interference degree among its neighbors and thus reduce the effective channel capacity. Also, if a source creates too many paths to a sink, the end-to-end channel capacity will be lowered due to the interference and contention among the paths. Therefore, the resource control scheme should adjust the target resource capacity between R and R_b, which is termed *resource control zone* as shown in Figure 5, to effectively make use of the resource. Depending on the resource type, the resource graph might look different. In Section VII, we will learn that the end-to-end channel capacity of a flow with respect to the energy consumption for the different number of multiple paths resembles the resource graph as shown in Figure 5.

B. Resource Control under Fidelity and Energy Constraints

In this section, we present the resource control strategy that achieves the highest fidelity level with the lowest energy consumption, thereby maximizing the packet energy efficiency, which combines both key criteria as shown in Equation 5. The problem of maximizing the packet energy efficiency, which is measured from the perspective of a whole network, can be reduced to the problem of minimizing the *bit energy* consumed by each individual node to successfully forward 1 bit to the next hop if the packet size is constant. The bit energy consumed by a node is calculated by dividing the total energy consumption by the total outgoing traffic that are successfully received by the next hop node. Therefore, the goal of the resource control strategy under fidelity and energy constraints during the event period of D is represented as follows:

$$minimize \quad \frac{\int_0^D E_i(t)dt}{\int_0^D min(T_i(t), R_i(t))dt}, \quad (16)$$

where $E_i(t)$ is the total energy consumed by node i at time t as already defined in Section IV and $min(T_i(t), R_i(t))$ indicates the outgoing traffic from node i at time t. Under the fidelity and energy constraints, the *target resource* is the effective channel capacity of a node and the *cost* for the target resource adjustment is measured by the energy consumed by the node.

We first exemplify the bit energy consumed by the Constant Increase/Decrease policy explained in Equation 10. Figure 6 illustrates the evolvement of the effective channel capacity

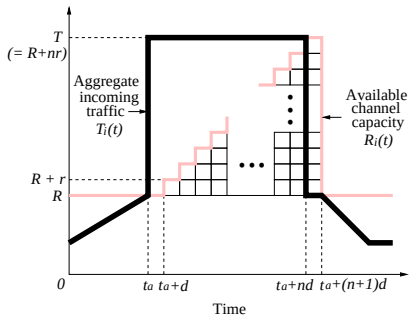


Fig. 6. Temporal progression of Constant Increase/Decrease policy

by the Constant Increase/Decrease policy when the aggregate incoming traffic becomes T at time t_a, exceeding the default channel capacity R. If the upper watermark is lower than T, the trinary congestion feedback function of node i returns 1 at time t_a, indicating the channel capacity should be increased. Let's assume it takes d time to adjust $R_i(t)$ irrespective of the amount of adjustment. Then at time $t_a + d$, $R_i(t_a + d)$ becomes $R + r$. In the figure, the aggregate incoming traffic later drops to the default channel capacity R at time $t_a + nd$ and the effective channel capacity falls back to the default channel capacity R at time $t_a + (n + 1)d$. To calculate the energy consumed between t_a and $t_a + (n + 1)d$, we define three energy constants as follows:

- e_l: average energy consumed by 1 unit of channel capacity of a node per unit of time when the node is idle. 1 unit of channel capacity can accommodate 1 bit of incoming traffic. Therefore, if a channel capacity of R is idle for the period of d, the total energy consumption is $e_l Rd$.
- e_r: average energy consumption by a node to successfully receive 1 bit from its neighbor node. e_r is calculated by dividing the total energy expended by the receiving activities of a node by the total amount of traffic (in bits) received from the previous hop nodes. The successfully received traffic can be later dropped due to queue overflow.
- e_t: average energy consumption by a node to successfully transmit 1 bit to the next hop node. e_t is calculated by dividing the total energy expended by the transmitting activities of a node by the total amount of traffic (in bits) successfully forwarded to the next hop nodes.

Generally, $e_t > e_r \geq e_l$ holds [27]. The event period is defined to be a time interval from the time when the incoming traffic volume hits the upper watermark until the time when the available channel capacity shrinks back to the default resource capacity R. In Figure 6, the event period of node i is between t_a and $t_a + (n + 1)d$. The total energy consumed by node i during an event period is the sum of the energy spent when the node is receiving, transmitting, or idle. The area of $T_i(t)$ between t_a and $t_a + (n + 1)d$, denoted as A_r, indicates the total traffic volume during the event period. The overlapping area of $T_i(t)$ and $R_i(t)$ between t_a and $t_a+(n+1)d$, denoted as A_t, corresponds to the total traffic volume forwarded during the event period. The area of $R_i(t)$ not covered by $T_i(t)$ between t_a and $t_a+(n+1)d$, denoted as A_l, indicates the over-provisioned channel capacity during the event period, which quantifies the degree of the node's idleness. Therefore, the total energy consumed by node i during the event period is represented as follows:

$$\begin{aligned} E_i &= A_r e_r + A_t e_t + A_l e_l \\ &= (ndT + dR)e_r + ((n+1)dR + \tfrac{n(n-1)}{2}dr)e_t + nrde_l, \end{aligned} \tag{17}$$

where $n = \frac{T-R}{r}$. The transmitted traffic volume during the event period is the area of A_t, which is $(n+1)dR+\frac{n(n-1)}{2}dr$. Therefore, the bit energy consumed during the event period under the scenario shown in Figure 6 is

$$\frac{(ndT + dR)e_r + ((n+1)dR + \frac{n(n-1)}{2}dr)e_t + nrde_l}{(n+1)dR + \frac{n(n-1)}{2}dr}. \tag{18}$$

Since $T = R + nr$, Equation 18 is reduced to

$$\frac{(n+1)R + n^2 r}{(n+1)R + \frac{n(n-1)}{2}r}e_r + e_t + \frac{nr}{(n+1)R + \frac{n(n-1)}{2}r}e_l. \tag{19}$$

Since the coefficient of e_r is greater than 1 and the last term is nonnegative in the above equation, the calculated bit energy consumption is greater than $(e_r + e_t)$, which is the optimal bit energy consumption as will be shown later. Therefore, the Constant Increase/Decrease policy is not optimal.

Now, we shall derive the resource control strategy that minimizes the bit energy consumption under an arbitrary incoming traffic function $T_i(t)$ to find the optimal resource control policy.

Theorem 1: The bit energy consumed by node i is minimized when the node's effective channel capacity is equal to the aggregate incoming traffic, i.e. $R_i(t) = T_i(t)$.

Proof: To find a resource control that minimizes the bit energy consumption of node i, we first depict arbitrary $R_i(t)$ and $T_i(t)$ in Figure 7(a). Each bit received but discarded later (possibly due to queue overflow or high channel contention) by node i consumes e_r energy. Each bit received and forwarded to next hop node by node i consumes $(e_r + e_t)$ energy. Therefore, the total energy consumed by node i between t_a and $t_a+(n+1)d$ is

$$\int_{t_a}^{t_a+(n+1)d} E_i(t)dt, \tag{20}$$

where

$$\begin{aligned} E_i(t) &= \begin{cases} (e_r + e_t)T_i(t) + e_l(R_i(t) - T_i(t)) & if\ T_i(t) < R_i(t) \\ (e_r + e_t)T_i(t) & if\ T_i(t) = R_i(t) \\ e_r T_i(t) + e_t R_i(t) & if\ T_i(t) > R_i(t) \end{cases} \\ &= \begin{cases} (e_r + e_t - e_l)T_i(t) + e_l R_i(t) & if\ T_i(t) < R_i(t) \\ (e_r + e_t)T_i(t) & if\ T_i(t) = R_i(t) \\ e_r T_i(t) + e_t R_i(t) & if\ T_i(t) > R_i(t). \end{cases} \end{aligned}$$

The successfully forwarded traffic volume from node i at time t is $min(T_i(t), R_i(t))$ and represented as follows in each case:

$$min(T_i(t), R_i(t)) = \begin{cases} T_i(t) & if\ \ T_i(t) < R_i(t) \\ T_i(t) & if\ \ T_i(t) = R_i(t) \\ R_i(t) & if\ \ T_i(t) > R_i(t). \end{cases} \tag{21}$$

Therefore, the bit energy consumed by node i at time t is represented as follows:

$$\begin{cases} (e_r + e_t - e_l) + \frac{R_i(t)}{T_i(t)}e_l & if\ \ T_i(t) < R_i(t) \\ e_r + e_t & if\ \ T_i(t) = R_i(t) \\ \frac{T_i(t)}{R_i(t)}e_r + e_t & if\ \ T_i(t) > R_i(t). \end{cases} \tag{22}$$

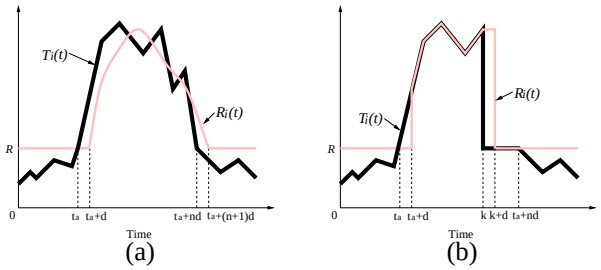


Fig. 7. (a) Arbitrary resource control and (b) EIED combined with traffic controlling

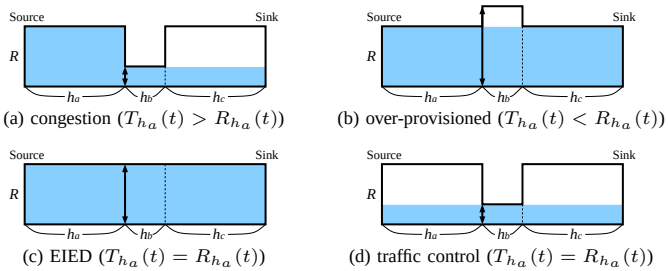


Fig. 8. Snapshots of effective channel capacity and traffic volume of a flow

In the resource control scheme, only $R_i(t)$ is adjustable. Therefore, the bit energy consumption is minimized to $(e_r + e_t)$ when $R_i(t)$ is equal to $T_i(t)$. □

Based on Theorem 1, a new resource control policy can be defined as follows:

$$R_i(t+1) = \begin{cases} T_i(t) & if \quad b_i(t) = 0 \\ R_i(t) & if \quad b_i(t) = -1 \\ T_i(t) & if \quad b_i(t) = 1. \end{cases} \tag{23}$$

The constants, w_u and w_l used in the trinary congestion feedback (Equation 7) are 0 in this equation. This control keeps adjusting its effective channel capacity to the aggregate incoming volume, spending or conserving its limited energy as *early* as possible. We call this policy *Early Increase/Early Decrease* (EIED). The EIED policy corresponds to the Fit Resource Control illustrated in Equation 15 when δ is 0. The above policy assumes that each node has an infinite energy budget to accommodate the incoming traffic, which, however, is unrealistic. Figure 7(b) illustrates the EIED policy combined with traffic controlling. Node i stops the resource control operation at time k and starts the traffic control operation by asking its previous hop node(s)[3] to reduce the incoming traffic volume to the default resource capacity R at time k, i.e. $T_i(k) = R$. The effective channel capacity is also reduced to the default resource capacity R at time $k+d$, i.e. $R_i(k+d) = R$. The traffic controlling is triggered (i) when the cumulative energy consumption after triggering the resource control scheme exceeds the energy budget B per event or (ii) when the cumulative outgoing traffic exceeds the required fidelity amount. A large k will increase the fidelity amount, but consume more energy, thereby exceeding the energy budget B.

[3]This approach is known as the hop-by-hop traffic control. Node i can take the end-to-end approach by sending a feedback explicitly or implicitly to the source(s) of traffic.

Now, we expand the EIED policy to the end-to-end level. Figure 8(a) shows a snapshot of the effective capacity and traffic volume of a flow from the source to the sink. In the figure, the traffic is forwarded from the leftmost source (node 0) to the rightmost sink (node $h(= h_a+h_b+h_c)$). The number of hops of the flow is $(h_a + h_b + h_c)$. The height of the shaded area indicates the traffic volume of each node on the routing path from the source to the sink. On the other hand, the effective channel capacity of each node is measured vertically inside the area bounded by the solid line. Therefore, the white area inside the area bounded by the solid line indicates the idle resource provisioning on the routing path. The nodes that are between h_a and (h_a+h_b) hops away from the source have the channel capacity of less than R, illustrating a congested scenario. The reporting rate of the source during congestion is as high as the default channel capacity R. Therefore, a hotspot forms around node h_a since node h_a's incoming traffic volume $T_{h_a}(t)$ exceeds the effective channel capacity $R_{h_a}(t)$ that is shown by the vertical arrow in the figure. Figure 8(b) shows that the effective channel capacities of the bottleneck nodes are now over-provisioned. Figure 8(c) shows that the effective channel capacities of the bottleneck nodes are adjusted to the incoming traffic volume at node h_a, T_{h_a}. Figure 8(d) shows that the reporting rate of the source is reduced to R_{h_a} by a traffic control scheme after congestion, thereby eliminating congestion at the hotspot.

During 1 unit of time, each bit in the shaded area consumes $(e_r + e_t)$ to be forwarded to the next hop while each unit of channel capacity in the white area consumes e_l. The delivered traffic volume to the sink at time t is R_{h_a} for Figure 8(a), (c), and (d) or R for Figure 8(b). Therefore, the bit energy consumption of the flow by the network[4] at time t is:

- if $R = T_{h_a}(t) > R_{h_a}(t)$ (i.e. Figure 8 (a)), $((h_b + h_c)(e_r + e_t) - h_c e_l + (h_a(e_r + e_t) + h_c e_l)\frac{R}{R_{h_a}(t)})$
- if $R = T_{h_a}(t) < R_{h_a}(t)$ (i.e. Figure 8 (b)), $((h_a + h_b + h_c)(e_r + e_t) - h_b e_l + h_b e_l \frac{R_{h_a}(t)}{R})$
- if $R = T_{h_a}(t) = R_{h_a}(t)$ (i.e. Figure 8 (c)), $(h_a + h_b + h_c)(e_r + e_t)$
- if $R > T_{h_a} = R_{h_a}$ (i.e. Figure 8 (d)), $((h_a+h_b+h_c)(e_r+e_t)-(h_a+h_c)e_l+(h_a+h_c)e_l\frac{R}{R_{h_a}(t)})$

The above analysis indicates that the bit energy consumption is minimized to $(h_a + h_b + h_c)(e_r + e_t)$ in an end-to-end level when the effective channel capacity at the hotspot is equivalent to the incoming traffic volume $T_{h_a}(t)$. Surprisingly, it is shown that traffic controlling (shown in Figure 8(d)) consumes higher bit energy than the EIED policy (shown in Figure 8(c)), which we will verify in the next section. Therefore, we introduce the following theorem.

Theorem 2: The EIED policy enforced distributedly by the nodes in the hotspot minimizes the bit energy consumption at the end-to-end level.

The EIED policy can be implemented in two ways: on the per-node basis or at an end-to-end level. In the per-node way,

[4]We assume the nodes that don't participate in the packet forwarding are inactive, i.e. their radio is off, consuming little energy.

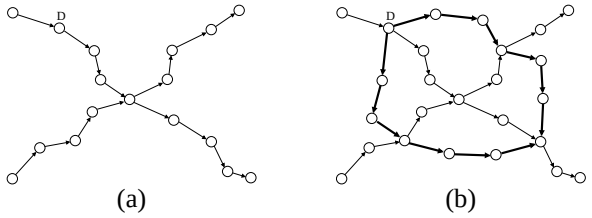


Fig. 9. Congestion at intermediate nodes and resource controlling by creating multiple paths.

each node individually adjusts its effective channel capacity by controlling duty cycle [27], [28] or transmission power [3], [18], [19], [31]. In the end-to-end way, nodes collaborate to provide a different end-to-end channel capacity [13]. For example, the end-to-end channel capacity will be increased if multiple paths between the source and the sink are created. In the next section, the EIED policy is verified by changing the number of the routing paths between the source and the sink.

VII. Verifying EIED using Simulations

In this section, the effectiveness of the EIED policy is verified under the fidelity and energy constraints and compared with the *ideal traffic control scheme*. The ideal traffic control scheme adjusts the reporting rates of source nodes in such a way it can achieve the highest fidelity level for the given channel capacity.

A. Simulation Environments

Since the effective channel capacity of a node is decided by many factors such as deployed MAC protocol, raw bit rate, transmission power, contention level, etc, the EIED policy can be implemented in many ways, for example by controlling transmission power, in sensor networks. In our simulation, a node's effective channel capacity is adjusted by dynamically creating and tearing down additional routing paths.

Dense sensor deployment is required for the flexible path creation and teardown. While *network density* from a higher level describes the total number of deployed nodes in the sensor field, the *neighbor degree*, i.e. the number of nodes within the radio range of a node, more accurately represents the density of a sensor network in terms of connectivity. While the network density δ is represented as $\frac{N}{A}$, where N is the number of active sensors (i.e. whose radio is on) and A is the size of a sensor field, the neighbor degree is defined as $\delta \pi R^2$, where R is the radio range of a node. Therefore, for a given sensor field, the average neighbor degree is decided either by radio range or by the average number of active nodes. To make a dense network with strong connectivity, 225 nodes with a communication radio range[5] of 50m are randomly deployed in the 530m by 530m sensor field. Therefore, the maximum neighbor degree is around 6 when all the nodes have their radios on.

A congestion scenario by cross traffic is shown in Figure 9(a)[6]. After node D detects congestion, it keeps creating additional paths quickly until congestion subsides; it tears down these additional paths when its channel capacity is later under-utilized. Therefore, the end-to-end channel capacity is always kept suitable for the incoming traffic level. Figure 9(b) shows that the end-to-end channel capacity of a flow is increased by creating 2 additional paths. To create an additional path, existing multiple path routing schemes [5], [9], [14], [15], [16], [17], [22], [25], [29] can be used. In our simulations, directed diffusion [12] is used as routing protocol. Two sinks that are far apart are randomly picked in the sensor field in each instance of simulation. For each sink, 3 sources are randomly picked, resulting in 6 flows in total. Each of the 3 sources for a given sink exclusively selects one of 3 event periods of 10-20, 30-40, and 50-60 seconds during the simulation time of 70 seconds and reports the event at a high rate which we will vary during its event period. The event reporting rate of a source during a dormant state is 1 packet/second. The underlying MAC protocol is IEEE 802.11 DCF with RTS/CTS. The data packet size is 100 bytes and the raw channel capacity is 2 Mbps.

[5]The interference radio range of a node is set to be twice the communication range.

[6]Only active nodes are depicted in the figures.

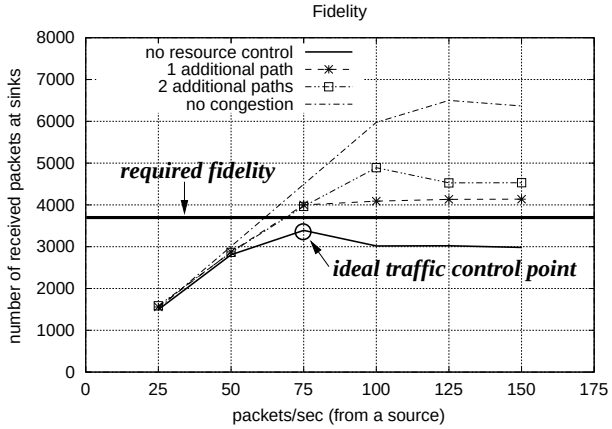


Fig. 10. Fidelity amount observed by sinks

B. Fidelity

Figure 10 shows the fidelity amount, i.e. total amount of delivered data to the sinks, during the event period with respect to different reporting rates at a source. As a baseline scenario, sources and sinks are carefully selected so that no congestion can happen, which is illustrated by the "no congestion" curve. To find an ideal traffic control, no resource controlling is performed, which is drawn on "no resource control" curve. The EIED schemes with the maximum number of additional paths of 1 and 2 are shown on "1 additional path" and "2 additional paths" curves, respectively.

A traffic control scheme will perform the best when it tunes each source's reporting rate to 75 packets/second on "no resource control" curve, where its fidelity amount is the largest. We call the reporting rate that yields the highest fidelity amount an *ideal traffic control point*. However, the fidelity amount at the ideal traffic control point is still below the required fidelity amount in the figure. Compared with this ideal traffic control, a single additional path enhances the fidelity level by at most 22% while two additional paths

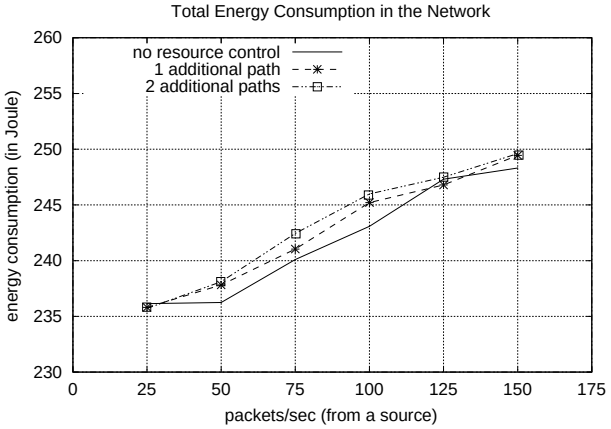


Fig. 11. Total energy consumption in the network

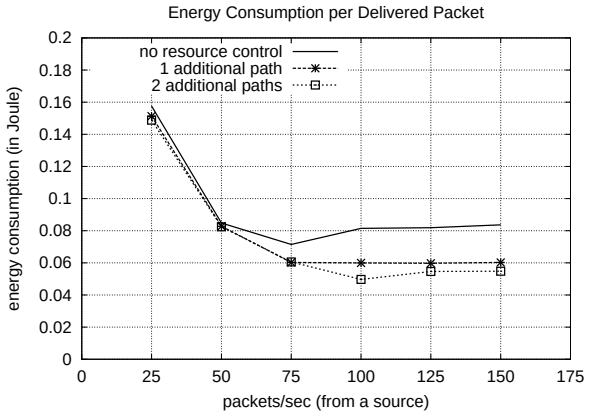


Fig. 12. Energy consumption per delivered packet

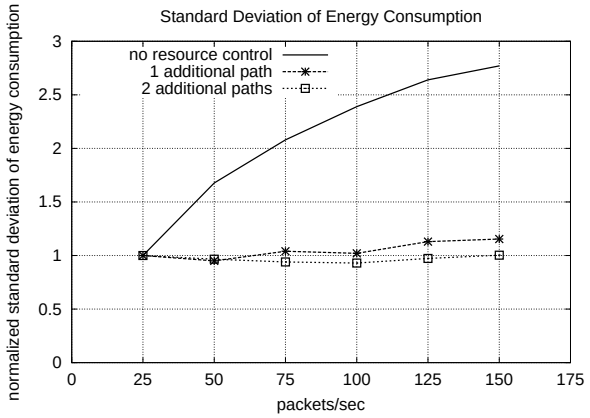


Fig. 13. Distribution of energy consumptions of the nodes in the sensor field

achieve at most 44% fidelity enhancement. The curve with more additional paths accommodates more traffic, thereby alleviating congestion by approaching the "no congestion" curve if the reporting rate of a source is less than or equal to 100 packets/second. When the source's reporting rate exceeds 100 packets/second, "no congestion" curve also shows the saturated fidelity amount, which, however, is caused not by the congestion of the cross traffic but by the reporting rate that is too high. Therefore, we focus on the results when the reporting rate of a source is less than or equal to 100 packets/second.

The above results may look straightforward since we increase the effective channel capacity by having additional paths. Since resource controlling is expected to consume more energy compared to the traffic control scheme during congestion, we need to investigate the packet energy efficiency.

C. Packet Energy Efficiency

Figure 11 shows the total energy consumption in the network for each strategy. Contrary to our intuition, the EIED policy does not consume much more energy compared to the "no resource control" case. In the "no resource control" case, a lot of energy has been wasted by high channel contention (i.e. collisions and subsequent retransmissions) around the hotspot when resource controlling is not performed. In the "no resource control" case, more energy can be save by reducing the source reporting rates, which, however, lowers the fidelity level. Figure 11 indicates that the total energy consumption without resource controlling at the ideal traffic control point is still lower than those with resource controlling.

Figure 12 shows the packet energy, i.e. the average energy consumed by the network for each delivered packet, under the same scenarios. It shows that even the packet energy consumption of the ideal traffic controlling (i.e. the "no resource control" curve when the reporting rate of a source is 75 packets/sec) is still higher than those of EIED schemes. This has been analytically proved earlier in Section VI-B and summarized in **Theorem 2**. These results show the additional energy consumed by the EIED policy offsets the energy wasted by packet drops due to congestion. To explain this, suppose a packet routes through p nodes $(n_1, ..., n_p)$ until it reaches the sink, and each unsuccessful 1 bit transmission requires e_u energy, and each successful 1 bit transmission requires $(e_t + e_r)$ energy, where e_r and e_t are defined in Section VI-B. Suppose that node n_i on the routing path transmits the packet whose size is P bits m_i times possibly due to contention ($m_i < M + 1$ where M is the maximum retransmission threshold) before it reaches node n_{i+1}. If the packet is dropped at the n' hop due to congestion after being transmitted $M+1$ times by that node, then the total amount of energy wasted for this packet is $P(\sum_{i=1}^{n'-1}(e_u(m_i-1)+(e_t+e_r))+e_u(M+1))$. This result shows resource controlling does not incur extra energy consumption as long as it reduces the packet drop rate. This result also shows that successfully delivering one packet in a dormant state (when the reporting rate a source is less than 50 packets/second in the figure) consumes significantly more energy than in a crisis state since nodes spend most of energy idle-listening, which strongly manifests the need of deploying an energy-conserving scheme (for example, dynamically changing a node's duty cycle).

Nodes in the sensor field should each spend a similar amount of energy to prevent an unbalanced energy depletion across network, which will shorten the lifetime of a sensor network by creating holes in the network [2], [10]. Figure 13 shows the standard deviation of the amount of energy consumption of each node in the network. The standard deviations in the three cases are normalized with respect to the standard

deviation at 50 packets/second in the "no congestion" case for the comparison purpose. The results show that resource controlling improves the energy consumption balance across the entire network irrespective of data rates. This is because the EIED algorithm is implemented by creating more paths and distributing the traffic over these paths that take advantage of the remaining energy of idle nodes. The ideal traffic control scheme may be able to alleviate unbalanced energy consumption by reducing the incoming traffic, but it cannot take advantage of idle nodes that are not on the routing path.

VIII. Concluding Remarks and Future Directions

This paper for the first time attempts to formally define the resource control framework that adjusts the effective resource capacity rather than the incoming traffic at the hotspot to alleviate congestion. We describe several potential metrics and intuitions behind the resource control that are applicable especially to the sensor network where the availability of resource is elastic and a certain level of throughput called fidelity is required. In an effort to find the optimal resource control under the fidelity and energy constraints, we present a resource increase and decrease algorithm called EIED that achieves the highest fidelity level with the lowest energy overhead by controlling the effective channel capacity to the level of incoming traffic load. We prove that the EIED algorithm performed in a distributed manner also lowers the packet energy consumption in the end-to-end level. We also discover that the EIED algorithm performs better than the ideal traffic control scheme in terms of fidelity and energy efficiency that are the key metrics in sensor networks.

We plan to pursue the following future directions.

- Our analysis has focused on minimizing the packet energy consumption mainly under the fidelity and energy constraints and has little understanding of convergence speed, stability properties of the EIED algorithm.
- Fairness is not discussed as part of the resource control objectives. Resource control with per-flow resource provisioning should be further explored to achieve fairness among competing flows.
- In the end-to-end model of the EIED algorithm, the bit energy consumption is assumed to be the same irrespective of the congestion level a node experiences. Generally, the bit energy consumption increases when the congestion level becomes high. Therefore, the bit energy consumption depending on the congestion level should be further studied.
- To implement the EIED algorithm more accurately in the real sensor network, quantifying the *extensible* resource with respect to the cost (e.g. energy) is necessary, so that required resource can be quickly increased as soon as congestion is detected.

References

[1] A. Cerpa and D. Estrin. ASCENT: Adaptive Self-Configuring Sensor Networks Topologies. In *Proceedings of IEEE INFOCOM*, June 2002.

[2] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2001)*, July 2001.

[3] M. Chiang. To Layer or Not To Layer: Balancing Transport and Physical Layers in Wireless Multihop Networks. In *Proceedings of IEEE INFOCOM'04*, March 2004.

[4] D.-M. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems 17(1):1-14*, 1989.

[5] S. De, C. Qiao, and H. Wu. Meshed multipath routing with selective forwarding: an efficient strategy in wireless sensor networks. In *Elsevier Computer Networks, Vol. 43, pp. 481-497*, 2003.

[6] C. T. Ee and R. Bajcsy. Congestion control and fairness for many-to-one routing in sensor networks. In *Proceedings of ACM SenSys*, Nov. 2004.

[7] S. Floyd. Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-way Traffic. *ACM Communication Review, 21(5):30-47*, Oct. 1991.

[8] S. Floyd and K. Fall. Promoting the Use of End-to-End Congestion Control in the Internet. *IEEE/ACM Transactions on Networking*, Aug. 1999.

[9] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-Resilent, Energy-Efficient Multipath Routing in Wireless Sensor Networks. In *ACM Mobile Computing and Communications Review, Volume 5, Issue 4, pp. 11-25*, 2001.

[10] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. An Energy-Efficient Surveillance System Using Wireless Sensor Networks. In *Proceedings of the ACM MobiSys 2004*, 2004.

[11] B. Hull, K. Jamieson, and H. Balakrishnan. Mitigating congestion in wireless sensor networks. In *Proceedings of ACM SenSys*, Nov. 2004.

[12] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networks (MobiCOM)*, August 2000.

[13] J. Kang, Y. Zhang, B. Nath, and S. Yu. Adaptive Resource Control Scheme to Alleviate Congestion in Sensor Networks. *Proceedings of IEEE Workshop on Broadband Advanced Sensor Networks*, 2004.

[14] S. J. Lee and M. Gerla. Dynamic Load-Aware Routing in Ad hoc Networks. In *Proceedings of IEEE International Conference on Communications (ICC)*, June 2001.

[15] M. K. Marina and S. R. Das. On-demand Multipath Distance Vector Routing in Ad Hoc Networks. In *Proceedings of International Conference for Network Protocols (ICNP)*, November 2001.

[16] A. Nasipuri, R. Castaneda, and S. R. Das. Performance of Multipath Routing for On-Demand Protocols in Mobile Ad Hoc Networks. In *ACM/Kluwer Mobile Networks and Applications (MONET), vol.6, no.4, pp. 339-349*, 2001.

[17] M. R. Pearlman, Z. J. Haas, P. Sholander, and S. S. Tabrizi. On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Networks. In *Proceedings of ACM MobiHoc*, August 2000.

[18] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *Proceedings of IEEE INFOCOM'00*, pages 404–413, 2000.

[19] R. S. Sreenivas P. R. Kumar S. Narayanaswamy, V. Kawadia. Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW Protocol. In *Proceedings of the European Wireless Conference - Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, February 2002.

[20] Y. Sankarasubramaniam, O. Akan, and I. Akyildiz. ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks. In *proceedings of the ACM MobiHoc Conference*, 2003.

[21] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman. Infrastructure tradeoffs for sensor networks. *ACM Workshop on Wireless Sensor Networks and Applications (WSNA)*, Sep. 2002.

[22] A. Valera, W. K. G. Seah, and SV Rao. Cooperative Packet Caching and Shortest Multipath Routing in Mobile Ad hoc Networks. In *Proceedings of IEEE INFOCOM*, March 2003.

[23] C. Wan, S. Eisenman, and A. Campbell. CODA: Congestion Detection and Avoidance in Sensor Networks. In *proceedings of the ACM SenSys 2003*, 2003.

[24] A. Woo and D. Culler. A Transmission Control Scheme for Media Access in Sensor Networks. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2001)*, July 2001.

[25] K. Wu and J. Harms. Performance Study of a Multipath Routing for Wireless Mobile Ad Hoc Networks. In *Proceedings of IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, February 2001.

[26] T. Yan, T. He, and J. A. Stankovic. Differentiated Surveillance Service for Sensor Networks. In *Proceedings of the ACM SenSys 2003*, 2003.

[27] F. Ye, G. Zhong, S. Lu, and L. Zhang. PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*, May 2003.
[28] W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of IEEE INFOCOM'02*, June 2002.
[29] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi. A Framework for Reliable Routing in Mobile Ad Hoc Networks. In *Proceedings of IEEE INFOCOM*, March 2003.
[30] Y. Yi and S. Shakkottai. A Hop-by-hop Congestion Control over a Wireless Multi-hop Network. In *Proceedings of IEEE INFOCOM'04*, March 2004.
[31] R. Zheng and R. Kravets. On-demand Power Management for Ad Hoc Networks. In *Proceedings of IEEE INFOCOM'03*, March 2003.

IX. Biographies

Jaewon Kang (jwkang@cs.rutgers.edu) is currently a Ph.D. candidate in the Department of Computer Science at Rutgers University. He received his B.S. and M.S. degrees in computer science & engineering from Seoul National University, Korea in 1992 and 1994, respectively. Before entering the Ph.D. program at Rutgers, he joined LG Electronics in 1994, working in the CDMA Smartphone Laboratory for the development of embedded operating system, wireless TCP/IP suite, CDMA data services for CDMA smart phones. He got an award from the IBM PSP Division for the contributions to the enhancement of the IBM Microkernel in 1995. His current research interest includes voice over packet wireless networks, fair packet scheduling in Internet backbone routers, and congestion control in wireless sensor networks.

Yanyong Zhang (yyzhang@winlab.rutgers.edu) received Ph.D. in Computer Science and Engineering from Penn State University in 2002. She is an assistant professor in the Department of Electrical and Computer Engineering at Rutgers University, and she is also a faculty member at Wireless Laboratory (WINLAB). Her current research interests include sensor networks, sensor network security and privacy, and fault-tolerant sensor networks. She has received several NSF grants on these topics, including an NSF CAREER award. She has organized the Workshops on System Management Tools for Large-Scale Parallel Systems (2005, 2006), and the Workshop on Self-Healing, Adaptive and Self-MANaged Systems (SHAMAN) 2002, and she is on the technical committee for several conferences and workshops. She is a member of ACM and IEEE.

Badri Nath (badri@cs.rutgers.edu) is currently a professor in the Department of Computer Science at Rutgers University and an associate director at WINLAB. He is a co-Principal investigator of the DataMan project at Rutgers University. His research interests are in the area of mobile computing and sensor networks. As part of his research, he is developing protocols and services suited for large scale dense networks. He current interest is on developing an information architecture for sensor networks. He is also the recipient of the ten year best paper award at VLDB and the best paper award at HPSR 2003. He received his Ph.D. degree from the University of Massachusetts, Amherst.