6-2009

Adaptive In-Network Processing for Bandwidth and Energy Constrained Mission-Oriented Multi-hop Wireless Networks

Sharanya ESWARAN
Pennsylvania State University

Matthew JOHNSON
City University of New York

Archan MISRA
Singapore Management University, archanm@smu.edu.sg

Thomas LA PORTA
Pennsylvania State University

DOI: https://doi.org/10.1007/978-3-642-02085-8_7

Citation
ESWARAN, Sharanya; JOHNSON, Matthew; MISRA, Archan; and LA PORTA, Thomas. Adaptive In-Network Processing for Bandwidth and Energy Constrained Mission-Oriented Multi-hop Wireless Networks. (2009). *Distributed computing in sensor systems: 5th IEEE International Conference, DCOSS 2009, Marina del Rey, CA, USA, June 8-10, 2009: Proceedings*. 87-102. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/667

Published in Distributed computing in sensor systems: 5th IEEE International Conference, DCOSS 2009, Marina del Rey, CA, USA, June 8-10, 2009: Proceedings. pp. 87-102. Berlin: Springer. LNCS 5516.
http://doi.org/10.1007/978-3-642-02085-8_7

Adaptive In-Network Processing for Bandwidth and Energy Constrained Mission-Oriented Multi-hop Wireless Networks*

Sharanya Eswaran[1], Matthew Johnson[2], Archan Misra[3], and Thomas La Porta[1]

[1] Networking and Security Research Center, Pennsylvania State University
[2] The Graduate Center, City University of New York
[3] Advanced Technology Solutions, Telcordia Technologies

Abstract. In-network processing, involving operations such as filtering, compression and fusion, is widely used in sensor networks to reduce the communication overhead. In many tactical and stream-oriented wireless network applications, both link bandwidth and node energy are critically constrained resources and in-network processing itself imposes non-negligible computing cost. In this work, we have developed a unified and distributed closed-loop control framework that computes both a) the optimal level of sensor stream compression performed by a forwarding node, and b) the best set of nodes where the stream processing operators should be deployed. Our framework extends the Network Utility Maximization (NUM) paradigm, where resource sharing among competing applications is modeled as a form of distributed utility maximization. We also show how our model can be adapted to more realistic cases, where in-network compression may be varied only discretely, and where a fusion operation cannot be fractionally distributed across multiple nodes.

1 Introduction

Many wireless sensor network (WSN) scenarios involve a set of long-running applications, operating over relatively low rates of discrete-event data, and are thus principally *energy-constrained*. Given that communication costs dominate computing costs [13] for relatively simple event-processing operations (such as averaging or finding the maximum of periodic temperature readings), in-network processing has been proposed as a means to increase the network operational

* This research was sponsored by US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

lifetime by reducing the volume of data transmitted to the sink (e.g., [9]). In this approach, an application is modeled as a graph of stream operators, overlaid on the physical wireless network topology.

Our focus is on a slightly different *stream-oriented wireless networking* scenario, where several of these implicit assumptions do not hold. In particular, many military applications involve the use of a multi-hop wireless network (comprising non-sensor nodes) for transporting relatively *high-data rate streams* from a set of sophisticated sensor sources (e.g., video cameras, acoustic arrays and short-range radar feeds) for use by *relatively shorter-duration tactical applications* (often called *missions*). For such environments, bandwidth is a critical shared resource, and congestion control algorithms (e.g., [17]) must be employed to effectively share the wireless link bandwidth among the competing missions. Moreover, the in-network operators for such stream-oriented data typically comprise more sophisticated DSP-based operations (e.g., MPEG compression or wavelet coefficient computation), for which the computational cost cannot be ignored [5]. Accordingly, the application of in-network processing to such sensor-based streaming applications must consider *both bandwidth and energy constraints* and recognize that the energy cost consists of *both communication and computing* overheads.

In the generalized model that we consider here, in-network processing may be viewed as a *tuning knob*, with higher levels of in-network processing (e.g., higher compression or coarser quantization) resulting in higher information loss for (or lower utility to) the application, but providing the benefit of reduced network bandwidth consumption. This introduces a non-linear tradeoff in the energy costs – in general, higher-levels of processing (e.g., more sophisticated compression techniques) lead to reduced transmission energy overheads, but a not-necessarily proportional increase in the computational energy [14].

In this paper, we first introduce and develop a distributed, closed-loop control framework that computes the optimal level of compression performed by a forwarding node on sensor streams, taking into account both energy and bandwidth constraints. In particular, we extend the Network Utility Maximization (NUM) paradigm, pioneered in [1,2], to model resource sharing among competing sensor-based applications as a form of distributed utility maximization. Initially, the physical location of the stream operators is assumed to be pre-specified. Subsequently, the physical location of the operator graph components is treated as another decision variable, i.e., we enhance our optimization model to additionally determine *the nodes where various in-network operations are performed.* We shall show how our technique can capture more realistic scenarios where the quality of in-network processing may be varied only in discrete steps, and where an operator may be instantiated only on a single node. Simulation-based studies, using a packet-level protocol implementation of our algorithms, are then used to demonstrate how "adaptive operator placement" and "variable-quality in-network compression" can together result in a significant improvement (as much as 39%) in overall mission utilities.

The rest of this paper is organized as follows. In Section 2, we explain the unique aspects of our problem; Section 3 briefly summarizes related work. Section 4 presents the mathematical model and protocol for the case where the location of the operators are specified a priori. Subsequently, Section 5 extends the solution to consider the problem of optimal placement of operators; Section 6 describes extensions to the the base algorithms to incorporate the real-life integral constraints. In Section 7, we present our simulation results. Finally, Section 8 concludes the paper.

2 The General Framework of Variable-Quality In-Network Processing and Dynamic Operator Placement

We consider two logically distinct in-network processing operations, compression and fusion:

Compression: The downstream transmission rate of most stream-oriented data can be reduced by the application of appropriate compression algorithms, both lossless and lossy. For example, an MPEG-4 (or higher standards, such as MPEG-21) video stream can be compressed to varying data rates. Compression may be performed independently at *every* forwarding node; conceptually, compression changes the quality (rate) of the output data, but not the *data type*.

Fusion: In contrast to compression, fusion may be viewed as a process of either combining or correlating data from multiple separate streams (e.g., superimposition of audio and video feeds) and/or altering the 'type' of a single data stream. An example of 'type' alteration involves the processing of an audio stream to extract only the 'talk spurts' from the signal.

We thus define an *operator graph* as a set of *fusion* operators. An operator placement algorithm maps each of the nodes of the operator graph to a subset of the forwarding nodes in the network; compression may then be viewed as an implicit data reduction operator permitted at any of the physical nodes lying between two consecutive components of the 'logical' operator graph.

The problem of resource-aware in-network processing was studied in [12], where each individual operator was assumed to be *immutable* (each operator being characterized by a fixed ratio between its output and input data rates) and mapped to a pre-defined forwarding node. Separately, [5] considered the communication+computing cost constraint in the absence of in-network processing. An obvious extension of these frameworks is to allow the placement of the stream processing operators to be a decision variable as well. Prior work on fusion operator placement (such as [6,7,8,9,10,11,12]) treats it as a stand-alone problem, where the objective is to place more selective operators closer to the data sources, without considering the interaction with variable data compression performed at intermediate nodes.

Prior work (such as [5]) also assumes a relatively simple scalar relationship between both computational and communication energy overheads and the incoming stream data rate. Many compression algorithms are, however, characterized by a non-linear energy-vs-compressibility curve, with the energy required for compression increasing dramatically when the ratio of output to input data rates falls below a certain threshold [18].

Based on the above discussion, the key new aspects of our problem formulation can be summarized as follows:

1. We consider the impact of variable quality compression of sensor streams, potentially performed by all forwarding nodes, on the capacity constraints and factor in the non-linear relationship between computational and communication energy overheads.
2. We also explicitly factor in the effect of such variable quality compression on the operator placement problem, and develop a solution that jointly selects both the location of fusion operators and the degree of compression that maximize cumulative system utility.

To solve this problem, we shall develop a NUM-based optimization framework and a fully-distributed protocol that seeks to jointly optimize the following free variables: *i) Source Rate, x:* the rate at which each sensor source transmits data, *ii) Compression Factor, l:* the level of compression, i.e., ratio of output rate to incoming rate, taking place at each forwarding node, and *iii) Operator placement:* the optimal node locations at which fusion operations take place.

3 Other Related Work

The classical NUM framework [1,2] was recently extended in [17] to a more general WSN environment, where individual missions derive their utility from a composite set of sensors, and intermediate nodes use link-layer multicast to forward sensor data downstream to multiple subscribing missions. In this WSN-centric model (referred to as WSN-NUM), the optimization problem is formulated as:

$$\textbf{maximize} \sum_{m \in M} U_m(X_m) \ \textbf{subject to} \sum_{\forall (k,s) \in q} \frac{x_s}{c_{k,s}} \leq 1, \ \forall q \in Q,$$

where q is one of the set (Q) of all maximal cliques in the conflict graph; $U_m(X_m)$ represents the utility function of mission m (M being the set of all missions) as a function of the vector of rates associated with the set of sensors S, and $c_{k,s}$ is the transmission rate used by node k during the link-layer broadcast of the data from sensor s. Based on this new model, a sensor (source) s adapts its rate as:

$$\frac{d}{dt} x_s(t) = \kappa\Big(\sum_{m \in Miss(s)} w_{ms}(t) - x_s(t) \sum_{\forall q \in Path(s)} \sum_{\forall (k,s) \in q} \frac{\mu_q(t)}{c_{k,s}}\Big) \tag{1}$$

where $\mu_q(t)$ (the 'cost' per bit charged by each forwarding clique) is given as $\mu_q(t) = (\sum_{\forall (k,s) \in q} \frac{x_s(t)}{c_{k,s}} - 1 + \varepsilon)^+ / \varepsilon^2$. Each mission (we assume that all the streams for a single mission are destined to a single 'sink' node) adapts its 'willingness to pay' term w_{ms} for sensor s based on the source rates and its

Table 1. Most Common Mathematical Symbols

M	Set of all missions	P^k_{max}	Max power budget at node k.
S	Total number of sources	P^k_{recv}	Power consumed at node k by data reception
$set(m)$	Set of sources used by mission m		
$Miss(s)$	Set of missions using flow s (directly or fused)	P^k_{trans}	Power consumed at node k by data transmission
$Path(s)$	Multicast route for flow s (raw or fused) from its source to Miss(s)	P^k_{comp}	Power consumed at node k by data processing
c_{ks}	Transmission rate at node k for flow s	P^k_{tot}	$P^k_{recv} + P^k_{trans} + P^k_{comp}$
x^{rec}_s	Received rate for flow s at a mission	α^k_{recv},	Power consumed per bit of received,
(k, s)	The transmission of flow s at node k	α^k_{trans},	transmitted and
$x_{in}(s, k)$	Incoming rate at node k for flow s	α^k_{comp}	compressed data at node k
$x_{out}(s, k)$	Outgoing rate for flow s from node k	$l_{k,s}$	Compression factor at node k for flow s

own utility function $U_m(.)$, according to $w_{ms}(t) = x_s(t)\frac{\partial U_m}{\partial x_s}$. The cost at each clique is cumulatively added along the forwarding nodes and piggy-backed with the data. The missions send this cost and willingness-to-pay as feedback to their sources. Each source uses this information to determine its rate for the next iteration, according to Eq. (1).

This notion of utility-based adaptation under in-network stream processing was first explored in [15] for wired networks, where each sensor flow is assumed to pass through an arbitrary processing graph, with each operator on the graph performing a *fixed* fractional reduction (or increase) in the output rate. In the absence of any constraints on the total power consumption at a node, the problem of optimal in-network processing and rate adaptation decomposes into the multi-rate multicasting problem. This problem was studied for multi-hop wireless networks in [16], where a back-pressure based solution was developed. A data gathering algorithm with tunable compression for communication and computation efficiency was developed in [18], but it did not consider the aspects of joint utilities, congestion control and operator placement.

4 The Network Model and the Optimization Problem

We first explain the process by which nodes select the optimal level of stream compression, assuming that the positions of the components of the operator graphs are pre-specified.

4.1 Assumptions

Our formulation and solution makes the following assumptions: (i) Each sensor's data flows over a pre-defined multicast tree to its set of subscribing sink nodes (each sink representing a mission). (ii) A fused stream cannot be subsequently disaggregated; accordingly fusion of two streams at a node is possible only if all downstream subscribers (for each of the two sensors) require the same fused information. (iii) Each sensor's flow is completely elastic, i.e., each node can adjust its transmission rate x_s by any arbitrary amount, as long as $x_s > 0$. (iv) The computational power required for compression increases with decrease in the compression factor (i.e., ratio of transmitted rate to incoming rate). (v) A fusion or compression operation performed by an intermediate node is applied identically to the flow on each of the outgoing links.

4.2 The Model

Each mission's utility is modeled as a joint function of the rate that it receives from multiple sensors. The utility of a mission m is a function of the rate at which it receives data, denoted as $U_m(\{x_s^{rec}\}_{s \in set(m)})$, where x_s^{rec} is the received rate of flow s and $set(m)$ is the set of sensors that are sources for m. $U(.)$ is assumed to be a jointly-concave function of the rates of all incoming flows. Table (1) lists the common mathematical symbols used in this paper.

The key feature of our model is to permit each intermediate node to perform a *'variable level of compression'*, denoted as $l_{k,s}$ (where $0 < l_{k,s} \leq 1$), that effectively alters the rate of a flow that is transmitted at node k and originated at source s. $l_{k,s}$ determines the ratio of the outgoing flow rate to the incoming flow rate for sensor s at node k, i.e., $l_{ks} = \frac{x_{out}(s,k)}{x_{in}(s,k)}$. The variable compression level l effectively acts as a 'tuning knob', allowing a forwarding node to modify the outgoing data rate in a manner that balances its competing computational and communication energy costs, and satisfies the capacity constraints. Intuitively, a congested network benefits from more aggressive compression. Conversely, a network operating at low link utilization should have little need for compression unless its transmission energy cost is too high.

The centralized model for this problem of utility maximization with adaptive in-network processing can be written as: **NUM-INP(U,C,P):**

$$\textbf{maximize} \sum_{m \in M} U_m(\{x_s^{rec}\}_{s \in set(m)}) - \delta \sum_{\forall nodes, k} P_{tot}^k, \textbf{ subject to} \tag{2}$$

$$\textbf{i) Capacity Constraint: } \sum_{\forall (k,i) \in q} \frac{x_{out}(i,k)}{c_{ki}} \leq 1, \forall q \in set\ of\ cliques,\ Q \tag{3}$$

$$\textbf{ii) Energy Constraint: } P_{tot}^k \leq P_{max}^k, \forall nodes, k \tag{4}$$

$$\textbf{where } P_{tot}^k = P_{rec}^k + P_{trans}^k + P_{comp}^k,\ 0 \leq \delta \leq 1 \textbf{ and } \ x_s \geq 0\ \forall s$$

The objective is to maximize the total utility of all missions, subject to an "energy" penalty function $\delta \sum_{\forall nodes,k} P_{tot}^k$, which ensures a unique solution by creating a convex optimization objective. δ (between 0 and 1) determines the weightage given to power consumption (vs. utility); in general, the penalty function can be the sum of any convex functions of P_{tot}^k. The capacity and energy constraints are explained as follows:

Capacity Constraint: The capacity constraint in Eq. (3) states that the total air-time fractions of all interfering transmissions (i.e., all transmissions in a maximal clique of the conflict graph) must not exceed unity. Please see [17] for further details.

Energy Constraint: The energy constraint in Eq. (4) states that the total power consumed at a node k due to data reception (P_{recv}^k), transmission (P_{trans}^k) and computation including both compression and fusion (if a fusion node) (P^k_{comp}) must not exceed the maximum power budget at node k (P^k_{max}). As is common in literature [3,4,5], we assume a linear energy model as follows:

$$P^k_{recv} = \alpha^k_{recv} \sum_{\forall \text{flows},\ s \text{ at } k} x_{in}(s,k);\ P^k_{trans} = \alpha^k_{trans} \sum_{\forall flows s \text{ at k}} x_{out}(s,k);$$

$$\text{If } k \text{ is not a fusion point: } P^k_{comp} = \alpha^k_{comp} \sum_{\forall \text{flows},\ s \text{ at } k} x_{in}(s,k)(\frac{1}{l_{ks}} - 1);$$

where $0 < l_{ks} \leq 1$. If k is a fusion point, there is an additional computational cost of $\alpha^k_{comp} \sum_{\forall flows, f\ fused\ at\ k} \frac{x_{out}(f,k)}{l_{kf}}$ incurred by the fusion process. Without loss of generality, we assume that this cost is proportional to the rate of the fused flow, and that the cost per bit is the same for compression and fusion.

4.3 Distributed Solution to the Optimization Problem

In order to solve this optimization problem in a distributed manner, we derive an iterative, gradient-based solution for the model shown in Eq. (2)-(4). We first make the problem unconstrained by taking Lagrangian as shown below:

$$\textbf{maximize} \sum_{m \in M} U_m(\{x^{rec}_s\}_{s \in set(m)}) - \delta \sum_{\forall nodes, k} P^k_{tot} -$$

$$\sum_{\forall cliques, q} \mu_q (\sum_{\forall (k,s) \in q} \frac{x_{out}(s,k)}{c_{ks}} - 1) - \sum_{\forall nodes, k} \eta_k (P^k_{tot} - P^k_{max})$$

where μ_q and η_k are Lagrangian multipliers. Using the first-order necessary conditions for gradients with respect to x_s and $l_{k,s}$, we get the following equations:

$$\frac{d}{dt} x_s(t) = \kappa x_s (\sum_{m \in Miss(s)} \frac{\partial U_m}{\partial x_s} - \sum_{\forall q \in Path(s)} \mu_q \sum_{\forall (k,s) \in q} \frac{\partial x_{out}(s,k)}{\partial x_s C_{ks}} - \sum_{\forall k \in Path(s)} (\eta_k + \delta) \frac{\partial P^k_{tot}}{\partial x_s}) \quad (5)$$

$$\frac{d}{dt} l_{k,i}(t) = \kappa l_{k,i} (\sum_{m \in Miss(i)} \frac{\partial U_m}{\partial l_{k,i}} - \sum_{\forall q \in Path(i)} \mu_q \sum_{\forall (v,i) \in q} \frac{\partial x_{out}(i,v)}{\partial l_{k,i} C_{vi}} - \sum_{\forall v \in Path(i)} (\eta_v + \delta) \frac{\partial P^v_{tot}}{\partial l_{k,i}}) \quad (6)$$

where, μ_q is defined as the shadow cost of congestion charged at each clique q and is given by $\mu_q(t) = (\sum_{\forall (k,s) \in q} \frac{x_s(t)}{c_{k,s}} - 1 + \epsilon)^+ / \epsilon^{\delta_1}$

Similarly, η_k is the shadow cost of energy charged at each node k and is given by $\eta_k(t) = (\frac{P^k_{tot}(t)}{P^k_{max}(t)} - 1 + \epsilon')^+ / \epsilon'^{\delta_2}$ where δ_1 and δ_2 are constants greater than 0. ϵ and ϵ' ($0 \leq \epsilon, \epsilon' \leq 1$) determine the tolerance margin [17].

Eq. (5) provides the algorithm by which the source sensors adjust their rates at each iteration; Eq. (6) shows how at each node, the degree of compression for each flow that the node forwards is varied in each iteration. We observe the following: (i) Source rate x_s depends on the rates at which the downstream nodes forward either this source's flow directly (when there is no fusion), or any flow derived from this source's flow (when there is fusion). Similarly, it also depends on the power consumed at all downstream nodes that forward either the source's direct flow or a flow derived (via fusion) from this source. (ii) The compression levels at the forwarding nodes depend on the forwarding rates and power consumption at all downstream nodes that receive this flow (either raw or fused).

When the source and forwarding rates are independently adjusted according to Eq. (5) and (6), the network converges at the optimal global utility, with penalties paid for congestion and power consumption. Please see [19] for proof.

4.4 Protocol-Level Implementation of the NUM Algorithm

The biggest challenge in building a fully-distributed and localized protocol for this model arises from the presence of fusion operators at specific intermediate WSN nodes. The stream that a mission receives is now obtained by fusing one or more flows from $set(m)$ according to a series of operators, as defined by the operator graph. An individual operator f can be viewed as a function that takes as input the rates of the flows to be fused, and gives as output the rate of the resulting fused flow.

Hence, the utility of a mission m is a joint function of rates x_i^{rec}, $\forall i \in$ *set of flows received at* m, with some of these flows being 'raw' flows (potentially compressed) from the corresponding sensor, and other flows being 'derived', through the application of a fusion operator at intermediate nodes (which act as the 'source' for the derived flow). While Eq. (5) refers only to rate adjustment at the 'raw' sources (i.e., sensors), the flow i in Eq. (6) may refer to either a raw or derived flow. Hence the distributed formulation in Eq. (5) and (6) is sufficient for deriving the optimal rates for both 'raw' and 'derived' flows.

From a protocol-perspective, however, the end-to-end feedback mechanism used in [17], whereby the sinks simply convey their willingness to pay directly to the source sensors, needs to be modified to reflect the inability of a sink to directly compute its 'willingness to pay' for a source that has passed through intermediate fusion points. For example, if a stream from source s is transformed

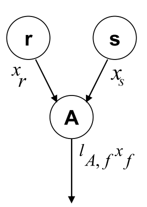

Fig. 1. Node A fuses flows r, s; transmits fused flow f

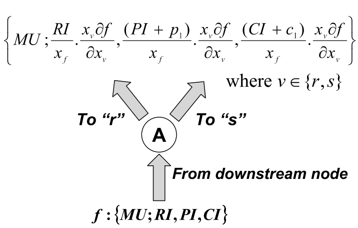

Fig. 2. Feedback messages received and propagated by A

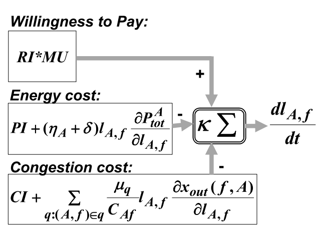

Fig. 3. Computation of $l_{A,f}$ according to Eq. (6)

twice by operators f and g before reaching a mission m, the mission is unable to compute its marginal utility $\frac{\partial U_m}{\partial x_s}$, because all it knows is the rate of the stream of type "$g \bullet f$", which contributes to its utility; it is unaware of both the source rate of s and the details of the fusion operations f and g. Here $g \bullet f$ refers to the composition function of the form $g(f(x_s, ...))$. The solution in this case is to use the "chain rule" for partial derivatives and compute $\frac{\partial U_m}{\partial x_s}$ as $\frac{\partial U_m}{\partial g(f(x_s,...))^{rec}} * \frac{\partial g(f(x_s,...))}{\partial f(x_s,...)} * \frac{\partial f(x_s,...)}{\partial x_s}$, where the fusion point for g and f provide the second and third terms, respectively.

Accordingly, in our NUM-INP protocol, the forward path carries only the data, but no meta-data. Nodes propagate the marginal utility, congestion cost and energy cost as metadata in signaling messages carried on the reverse forwarding path; nodes use these feedback messages to compute the compression levels and the source rates for the next iteration, in addition to updating and propagating them upstream.

For each stream r that a mission m receives, it sends a feedback (periodically), to the node that forwarded this stream. The feedback message consists of: i) A *marginal utility MU* field, where the mission enters its marginal utility with respect to the received flow rate ($\frac{\partial U}{\partial x_s^{rec}}$); this is used for computing the 'willingness-to-pay' according to the chain rule. ii) A 4-tuple consisting of the fields *flow name* (the ID of the 'flow'), *rate information (RI)* (the rate at which the mission receives the flow, *power information (PI)* (the energy cost attributed to this flow) and *congestion information (CI)* (the normalized congestion cost at all the cliques that this node belongs to).

If an intermediate node was a branching point on the multicast forwarding tree, it collects the feedback from all its child nodes and combines them into a single feedback message.

The cost fields are updated at each node in the reverse path, to compute the cumulative cost along the path, and the fusion points make additional modifications to capture the effect of fusion operation (according to the chain rule). For example, when a forwarding node A receives a feedback message for flow f from a downstream node, it adds its own energy cost for f to the PI field (i.e., $PI = PI + (\eta_A + \delta)P^A_{tot}(f)$) and its own congestion cost for f to the CI field (i.e., $CI = CI + \sum_{\forall q:(A,f)\in q} \mu_q \frac{x_{out}(f,A)}{C_{A,f}}$) before passing the feedback message to its upstream neighbor. If A is also the fusion point where the fused flow f originates, then all the fields in the table are further multiplied by the term $\frac{l_{A,f}}{x_{out}(f,A)} * x_{in}(s, A) * \frac{\partial f}{\partial x_{in}(s,A)}$, before propagating the feedback upstream. Using the meta-data in the feedback message, the forwarding nodes and source nodes compute the compression levels and source rates for the next iteration, according to Eq. (5) and (6). Fig. (1)-(3) illustrate the propagation of feedback and computation of compression level for a simple example. In Fig. (2), $v = r$ in the feedback to r and $v = s$ in the one to s; $p_1 = (\eta_A + \delta)P^A_{tot}(f)$, $c_1 = \sum_{\forall q:(A,f)\in q} \mu_q \frac{x_{out}(f,A)}{C_{A,f}}$.

5 Adaptive Operator Placement

In the previous section, we assumed that the locations of the fusion operators are fixed and given a priori. In this section, we describe how the NUM-INP framework can be enhanced to additionally determine the optimal placement of the fusion operators. Ideally, the communication cost is lowest if a fusion operation takes place as close to the sources as possible. However, due to energy constraints, nodes closer to the source may not be able to perform the fusion operation; in such situations, higher utility may be obtained by pushing the operator to a node downstream. Our approach is to integrate operator placement into the NUM framework (in parallel to source rate adaptation and adaptive compression quality), albeit as an "outer" optimization loop that occurs at a slower time-scale.

With the help of an operator graph, the forwarding trees and the mission subscription information, the nodes in a network can determine if they are candidate-locations for a fusion operator. For example, for the simplistic network shown in Fig. (4), where mission M requires the fused flow, $f(x_{s_1}, x_{s_2})$, the fusion can take place at node A or B or C. We assume that each node runs a preliminary protocol (details of which are not relevant to this work) to determine which fusion operations can be performed at that node. We also assume that the fusion operations can be expressed as functions of the rates of their input flows.

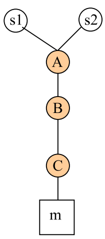

Fig. 4. Example network

Our approach is to allow all candidate locations to perform fusion on an *arbitrary fraction* of the input streams, and transmit the rest as raw streams. This fraction is variable and is adjusted iteratively in a NUM-based control loop, and it converges at the optimal value. Let k be a representative candidate node for the fusion operation $f(x_{s_1}, x_{s_2}, x_{s_3}, ...x_{s_n})$ that fuses flows $F = \{s_1, s_2, s_3, ..., s_n\}$. Let θ^k_{f,s_i} (where $s_i \in F$) be the fraction (lying between 0 and 1) of the input flow s_i that is fused at node k. The rest of the input flow is passed on downstream, where the next candidate node fuses all or a fraction of it, and so on. The mission sink is always a candidate for all fusion operators, and can absorb any residual "unfused" stream data.

For the example shown in Fig. (4), node A fuses according to $f(\theta^A_{f,s1}x_{s_1}, \theta^A_{f,s2}x_{s_2})$ and forwards input flows s_1, s_2 and the fused flow, f^A at rates $l_{A,s_1}(1-\theta^A_{f,s_1})x_{s_1}$, $l_{A,s_2}(1-\theta^A_{f,s_2})x_{s_2}$ and $l_{A,f}f(\theta^A_{f,s_1}x_{s_1}, \theta^A_{f,s_2}x_{s_2})$, respectively, where $l_{k,s}$ refers to the compression factor for flow s at node k. Subsequently, node B forwards the input flows at rate $l_{B,s}l_{A,s}(1-\theta^A_{f,s})(1-\theta^B_{f,s})x_s$, where $s \in \{s_1, s_2\}$, along with flow f^A (i.e., flow fused at A) compressed at $l_{B,f}$. It also forwards the new 'sub-flow' f^B fused at B at rate $l_{B,f}f(l_{A,s_1}(1-\theta^A_{f,s_1})\theta^B_{f,s_1}x_{s_1}, l_{A,s_2}(1-\theta^A_{f,s_2})\theta^B_{f,s_2}x_{s_2})$. If the optimal value of θ after convergence is 1 at a node, then that node is the unique optimal location for fusion. It is also possible that the optimal configuration is for multiple nodes to share the responsibility of fusion (i.e., two or more of the candidate nodes will have $0 < \theta < 1$). Such 'fractional fusion' can be interpreted as a process of "time-sharing" the responsibility of fusion across the candidate nodes.

The generic model in Eq. (2)-(4) holds for this problem too; the source rates and compression factors continue to be adjusted according to Eq. (5) and Eq. (6), respectively. By taking the Lagrangian of the "θ-enhanced" NUM objective, we derive the θ-adjustment algorithm for a fusion operation op to be: $\frac{d}{dt}\theta^k_{op,s} =$

$$\kappa\theta^k_{op,s}\Big(\sum_{m\in Miss(s)} \frac{\partial U_m}{\partial \theta^k_{op,s}} - \sum_{\forall q\in Path(s)} \mu_q \sum_{\forall (v,s)\in q} \frac{\partial x_{out}(s,v)}{\partial \theta^k_{op,s} C_{vs}} - \sum_{\forall v\in Path(s)} (\eta_v + \delta) \frac{\partial P^v_{tot}}{\partial \theta^k_{op,s}}\Big) \quad (7)$$

We observe from Eq. (7) that the θs at candidate fusion points depend on the forwarding rates and power consumption at all downstream nodes that receive the flows, either directly or after fusion, from this node. It must be noted that in this problem, the values of x_{in}, x_{out}, as well as the nodes in the sets $path(i)$ must now be computed depending on the values of θ's. We prove in [19] that this algorithm converges at the optimal solution.

5.1 Protocol-Level Modifications for Operator Placement

The introduction of adaptive operator placement requires modifications to the signaling mechanism along the reverse forwarding path. This is because, a mission subscribing to a fused flow now receives multiple 'sub-flows', each fused at a different candidate location, along with the original flows (to be fused directly at the mission). Hence, the feedback message now consists of a table of 4-tuples, called the Feedback Information Table (FIT), instead of a single entry. The fields in the 4-tuple remain the same as described in Section 4.4 and there is an entry (row) in FIT corresponding to each sub-flow received at the mission. The nodes along the reverse-forwarding path update the cost information for each of the sub-flows, and the fusion-point for each sub-flow is responsible for augmenting the meta-data with the chain-rule information. In order to reduce the signaling overhead, we maintain a special row in FIT, called the *cumulative* entry for each original flow (i.e., each input to the fusion operation); at each candidate fusion point, the meta-data in the row corresponding to its sub-flow is added to the *cumulative* entries and the row is removed. Thus as the feedback message propagates upwards, the FIT reduces in size, with all its entries eventually collapsing to the *cumulative* rows.

In the example network of Fig. (4), mission m receives flows fused at A, B, C and also the raw streams s_1 and s_2 (if the fusion points do not fuse all the data). Hence, m sends feedback to C with marginal utility as $\frac{\partial U_m}{\partial(x_{fA}+x_{fB}+x_{fC}+f(x_{s1},x_{s2}))}$ (where x_{f^k} refers to the rate of flow of type f that is fused at node k), and FIT with five rows, corresponding to s_1, s_2, f^A, f^B and f^C. When C receives this message, it does the following: (i) updates congestion and energy cost for all the sub-flows, (ii) adds the rate and cost information for f^C to the corresponding fields in the *cumulative* entry and (iii) removes row f^C. Subsequently nodes B and A update the message in a similar fashion, such that the feedback that arrives at source s_1 consists of only two rows in FIT: s_1 and $cumulative^{s_1}$ (and similarly for s_2). Please see [19] for a more detailed example.

The forwarding nodes use the feedback message to compute the θ and compression values for the next iteration, and the source nodes compute the new flow rates. The pseudo-code for this adaptation process is given in [19]. We note that only minimal amount of information is signaled and the algorithms have been devised such that Eq. (5, 6, 7) can be computed precisely from just this minimal meta-data and locally available information.

6 NUM Modifications to Address Practical Constraints

For mathematical tractability, the NUM-based technique for "optimal" variable in-network compression and operator placement requires both these processes to be represented as continuous variables. These assumptions are likely to be violated in practice. We now describe how the NUM algorithm can be modified to address both these practical limitations.

Discrete Compression Levels: Most of the commonly used compression techniques provide for multiple, but *discrete*, compression levels. For instance, *gzip* provides 9 levels of compression, JPEG allows a range of 0 to 100 levels, and MP3 allows compression ratios ranging from 12:1 to 10:1. The discontinuity arising from such integral choices prevents the direct application of NUM's gradient search techniques and in fact, makes the problem *NP-hard* [19]. Our NUM- based heuristic is to run the protocols using a continuous compression model, but simply map the computed $l_{k,s}$ value to the nearest valid discrete compression level at each iteration.

Solitary Operator Location: Our theoretical model assumes that a particular fusion operator may be "split" (in different fractions) across multiple nodes. In practice, many operators may not be conducive to such fractional splitting over infinitesimal time-scales. In such cases, our heuristic solution is to assign the responsibility for fusion to the node with the "largest θ". A heuristic based approach is required because the problem of determining the best single location for a fusion operator is an *NP-hard* combinatorial problem as well [19]. The selection of this single fusion point may be performed at each iteration of the NUM θ-loop (Eq. (7)). To achieve this, the highest cumulative θ value of downstream nodes is also propagated up the reverse forwarding path; the most upstream node among the fusion candidates can then designate the node with the most fusion responsibility as the sole fusion point. However, to ensure rapid convergence, the other terms (in the Feedback Information Table) carried in the signaling messages are based on the use of the 'virtual' continuous-θ values.

7 Evaluation

In this section we evaluate the performance of the NUM-INP protocol based on a packet-level simulation on an 802.11-based multi-hop wireless network, using the discrete-event simulator Qualnet [20]. The values of α^k_{recv}, α^k_{trans} and

α^k_{comp} are taken as $0.75\mu J/bit$, $0.6\mu J/bit$ and $0.54\mu J/bit$, based on the data from [14].

Utility Gain Due to In-network Processing: Fig. (5) illustrates the rates obtained from adaptive in-network compression on a sample simulated topology, where the flows from sources 1 and 2 are fused at node 3 and the fused flow is forwarded to missions $A-H$. The compression factor and transmission rate at each node, and the rate at which each mission receives data (x^{rec}) are shown in the figure. The utility of a mission is of the form $\gamma ln(1+x^{rec})$. For missions A and B, $\gamma = 100$; for missions C and D $\gamma = 20$; for missions E and F $\gamma = 1$; for missions G and H $\gamma = 0.25$. As illustrated, in our model, missions that have higher utility receive the fused flow at higher data rate. On the contrary, if there is no in-network compression, then all the missions receive at a uniform rate of 11.57 kbps. The values shown within parentheses are the compression factors and rates when only four discrete compression levels (0.25, 0.5, 0.75 and 1.0) are allowed. We observe that the rates with discrete compression are fairly close to the optimal values that can be achieved when the compression is a continuous-valued variable.

Fig. (6) compares the utilities of a network under three cases: *a)* with only source rate adaptation (according to WSN-NUM) but no in-network compression, *b)* optimal variable quality compression with pre-specified fusion locations and *c)* with joint optimization of compression and operator placement. The simulated network consists of 100 nodes of random topology in a 1500m x 1500m field. There are 25 missions and 25 sources and 15 fusion operations, whose initial locations are picked randomly from the sets of candidate locations (given by operator task graph). We can see that with NUM-INP, the global utility of the network is higher (by about 30%); the joint optimization of the operator locations results in a further 18% gain in system utility.

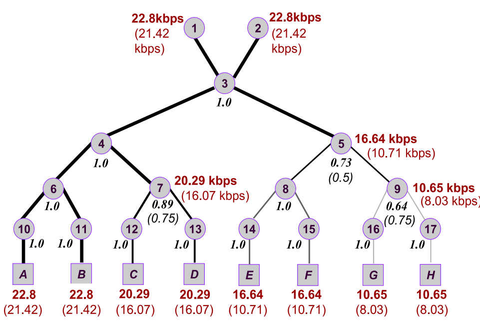

Fig. 5. Illustration of adaptive in-network compression with continuous and discrete levels

Performance Scalability: Fig. (7) shows the percentage gain in utility achieved by NUM-INP protocol, compared to simple source rate adaptation (WSN-NUM), when the number of missions and sources in the network are varied. We see that the gain increases with an increase in the number of competing missions and sensor sources. We experimented with different topologies and observed similar results in all cases. The relative gain with in-network processing is higher when the number of missions is larger; adaptive in-network compression and fusion helps to alleviate congestion bottlenecks, while adhering to the energy consumption constraints. We also tested the signaling overhead for different numbers of candidate nodes and fusion operations and the overhead was very low, in the order of tens of bytes per second.

NUM-INP under "Realistic Constraints": We study the impact of discrete compression levels by computing the loss in overall utility as a function of the number of discrete compression levels permitted. We map a compression factor value to a particular level, depending on how many levels are available. For example, when 10 levels of compression are allowed, we let $level\ 1 = 0.1$, $level\ 2 = 0.2$, and so on. Fig. (8) plots the system utility (normalized over the optimal utility with continuous compressibility). We see that the utility is at least 95% of the optimal for 10 or more number of discrete levels, but drops rapidly if the number of distinct compression levels is very small.

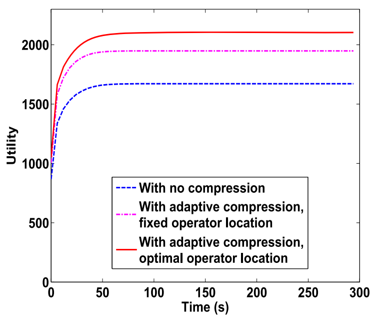

Fig. 6. Impact of in-network processing

Fig. (9) shows the normalized utility as a function of the number of fusion operators, when partial fusion is prohibited and fusion occurs at a solitary node (as described in Section 6). For each fusion operator, the number of candidate nodes was randomly chosen to be between 2 and 10. We see that the utility remains close to the optimal even as the number of in-network fusion operations is increased, with only at most 5% loss in system utility. By comparing this result to Fig. (6), where adaptive operator placement offers an additional 18% gain in utility, we see that joint optimization of compression and operator placement is beneficial, even if fractional operator placement is not permitted.

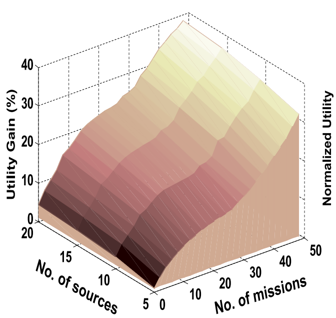

Fig. 7. Impact of number of missions and sources

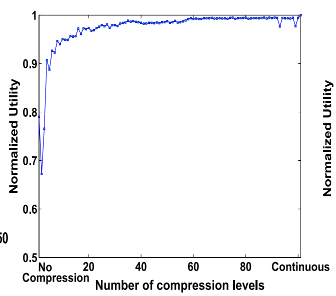

Fig. 8. Impact of discrete compression levels

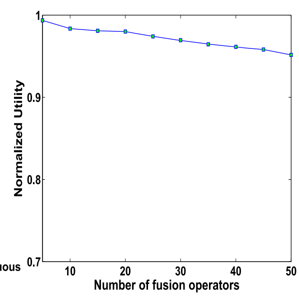

Fig. 9. Impact of single node fusion

8 Conclusion

In this work, we have developed a utility-based protocol for adaptive in-network processing, for wireless networks with streaming sensor sources, which maximizes the sum of mission utilities by jointly optimizing the source data rate, the degree of stream compression and the location of fusion operators. Our protocol can achieve up to 39% higher utility than pure source-rate adaptation, with only modest signaling overhead. In ongoing work, we are extending this framework to dynamically modify the level of in-network processing, taking network lifetime objectives into account.

References

1. Kelly, F.P., Maulloo, A.K., Tan, D.K.H.: Rate control for communication networks: shadow prices, proportional fairness and stability. JORS 49, 237–252 (1998)
2. Low, S.H., Lapsley, D.E.: Optimization flow control,I: Basic algorithm and convergence. IEEE/ACM ToN 7, 861–874
3. Freeney, L.M., Nilsson, M.: Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In: Proc. of IEEE INFOCOM (April 2001)
4. Hou, Y.T., Shi, Y., Sherali, H.D.: Rate allocation in wireless sensor networks with network lifetime requirement. In: Proc. of ACM MobiHoc (May 2004)
5. Zhang, C., Kurose, J., Liu, Y., Towsley, D., Zink, M.: A distributed algorithm for joint sensing and routing in wireless networks with non-steerable directional antennas. In: Proc. of ICNP 2006 (2006)
6. Madden, S., Franklin, M., Hellerstein, J., Hong, W.: Tag: A tiny aggregation service for ad hoc sensor networks. In: ACM SIGOPS Operating Systems Rev., December 2002, pp. 131–146 (2002)
7. Bonfils, B., Bonnet, P.: Adaptive and decentralized operator placement for in-network query processing. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 47–62. Springer, Heidelberg (2003)
8. Ahmad, Y., Cetintemel, U.: Network-aware query processing for stream-based applications. In: Proc. of VLDB 2004 (2004)
9. Srivastava, U., Munagala, K., Widom, J.: Operator Placement for in-network stream query processing. In: Proc. PODS 2005 (2005)
10. Pietzuch, P., Ledlie, J., Shneidman, J., Roussopoulos, M., Welsh, M., Seltzer, M.: Network-aware operator placement for stream-processing systems. In: Proc. of ICDE (2006)
11. Abrams, Z., Liu, J.: Greedy is good: On service tree placement for in-network stream processing. In: Proc. of ICDCS 2006 (2006)
12. Ying, L., Liu, Z., Towsley, D., Xia, C.: Distributed Operator Placement and Data Caching in Large-Scale Sensor Networks. In: Proc. INFOCOM 2008, Phoenix, AZ (2008)
13. Sadler, C.M., Martonosi, M.: Data compression algorithms for energy-constrained devices in delay tolerant networks. In: Proc. of ACM SenSys, pp. 265–278 (2006)
14. Barr, K.C., Asanović, K.: Energy-aware lossless data compression. ACM TOCS 24(3), 250–291 (2006)
15. Xia, C., Towsley, D., Zhang, C.: Distributed Resource Management and Admission Control of Stream Processing Systems with Max Utility. In: Proc. of the ICDCS, June 2007, pp. 68–75 (2007)

16. Bui, L., Srikant, R., Stolyar, A.L.: Optimal Resource Allocation for Multicast Flows in Multihop Wireless Networks. In: Proc. of IEEE CDC (December 2007)
17. Eswaran, S., Misra, A., Porta, T.L.: Utility-Based Adaptation in Mission-oriented Wireless Sensor Networks. In: Proc. of IEEE SECON (June 2008)
18. Yu, Y., Krishnamachari, B., Prasanna, V.K.: Data Gathering with Tunable Compression in Sensor Networks. IEEE TPDS 19(2), 276–287 (2008)
19. Eswaran, S., Misra, A., La Porta, T.F.: Adaptive In-network Processing for Bandwidth and Energy Constrained Mission-oriented Wireless Sensor Networks. Technical Report, Dept. of CSE, Pennsylvania State University (October 2008)
20. `http://www.qualnet.com`