When In-Network Processing Meets Time: Complexity and Effects of Joint Optimization in Wireless Sensor Networks

(**Technical Report: WSU-CS-DNC-TR-09-01**)

Qiao Xiang, *Student Member, IEEE,* Hongwei Zhang, *Member, IEEE*
Jinhong Xu, Xiaohui Liu, Loren J. Rittle

***Abstract*—As sensornets are increasingly being deployed in mission-critical applications, it becomes imperative that we consider application QoS requirements in in-network processing (INP). Towards understanding the complexity of joint QoS and INP optimization, we study the problem of jointly optimizing packet packing (i.e., aggregating shorter packets into longer ones) and the timeliness of data delivery. We identify the conditions under which the problem is strong NP-hard, and we find that the problem complexity heavily depends on aggregation constraints (in particular, maximum packet size and re-aggregation tolerance) instead of network and traffic properties. For cases when the problem is NP-hard, we show that there is no polynomial-time approximation scheme (PTAS); for cases when the problem can be solved in polynomial time, we design polynomial time, offline algorithms for finding the optimal packet packing schemes. To understand the impact of joint QoS and INP optimization on sensornet performance, we design a distributed, online protocol *tPack* that schedules packet transmissions to maximize the local utility of packet packing at each node. Using a testbed of 130 TelosB motes, we experimentally evaluate the properties of tPack. We find that jointly optimizing data delivery timeliness and packet packing and considering real-world aggregation constraints significantly improve network performance. Our findings shed light on the challenges, benefits, and solutions of joint QoS and INP optimization, and they also suggest open problems for future research.**

***Index Terms*—Wireless network, sensor network, real-time, packet packing, in-network processing**

I. Introduction

After the past decade of active research and field trials, wireless sensor networks (which we call *sensornet*s hereafter) have started penetrating into many areas of science, engineering, and our daily life. They are also envisioned to be an integral part of cyber-physical systems such as those for alternative energy, transportation, and healthcare. In supporting mission-critical, real-time, closed loop sensing and control, CPS sensornets represent a significant departure from traditional sensornets which usually focus on open-loop sensing, and it is critical to ensure messaging quality (e.g., timeliness of data delivery) in CPS sensornets. The stringent application requirements in CPS make it necessary to rethink about sensornet design, and one such problem is in-network processing.

For resource constrained sensornets, in-network processing (INP) improves energy efficiency and data delivery performance by reducing network traffic load and thus channel contention. Over the past years, many INP methods have been proposed for query processing [1], [2], [3], [4] and general data collection [5], [6], [7], [8], [9], [10]. Not focusing on mission-critical sensornets, however, these works have mostly ignored the timeliness of data delivery when designing INP mechanisms. Recently, Becchetti $et\ al.$ [11] and Oswald $et\ al.$ [12] examined the issue of data delivery latency in in-network processing. Theoretical in nature, these studies assumed *total aggregation* where any arbitrary number of information elements (e.g., reports after an event detection) can be aggregated into one single packet, which may well be infeasible in many practical settings. Thus, the interaction between specific, real-world INP methods and data delivery timeliness remains a largely unexplored issue in sensornet systems. This is an important issue because 1) it affects the efficiency and quality of real-time embedded sensing and control, and 2) as we will show later in the paper, data aggregation constraints (e.g., aggregation capacity limit and re-aggregation tolerance) affect, to a greater extent than network and traffic properties, the complexity and the protocol design in jointly optimizing INP and the timeliness of data delivery.

Towards understanding the interaction between INP and data delivery latency in foreseeable real-world sensornet deployments, we focus on a widely used, application-independent INP method — *packet packing* where multiple short packets are aggregated into a single long packet [13], [14]. In sensornets (especially those for real-time sensing and control), an information element from each sensor is usually short, for instance, less than 10 bytes [15], [1]. Yet the header overhead of each packet is relatively high in most sensornet platforms, for instance, up to 31 bytes at the MAC layer alone in IEEE 802.15.4 based networks. It is also expected that more header overhead will be introduced at other layers (e.g., routing layer) as we standardize sensornet protocols such as in the effort of the IETF working groups 6LowPAN [16] and

An extended abstract containing some preliminary results of this paper has appeared in IEEE RTSS 2009.

Qiao Xiang, Hongwei Zhang, and Xiaohui Liu are with the Department of Computer Science, Wayne State University, U.S.A. E-mail: {xiangq27,hongwei,xiaohui}@wayne.edu.

Jinhong Xu is with the Department of Computer Science, Indiana University, USA. E-mail: jinhxu@indiana.edu.

Loren J. Rittle is with with the Content and Context-aware Solutions group, Applied Research, Motorola Mobility, USA. E-mail: ljrittle@motorola.com

ROLL [17]. Besides header overhead, MAC coordination also introduces non-negligible overhead in wireless networks [14]. If we only transmit one short information element in each packet transmission, the high overhead in packet transmission will significantly reduce the network throughput; this is especially the case for high speed wireless networks such as IEEE 802.15.4a ultrawideband (UWB) networks. Fortunately, the maximum size of packet payload is usually much longer than that of each information element, for instance, 128 bytes per MAC frame in 802.15.4. Therefore, we can aggregate multiple information elements into a single packet to reduce the amortized overhead of transmitting each element. Packet packing also reduces the number of packets contending for channel access, hence it reduces the probability of packet collision and improves information delivery reliability, as we will show in Section VI. The benefits of packet packing have also been recognized by the IETF working groups 6LowPAN and ROLL.

Unlike total aggregation assumed in [11] and [12], the number of information elements that can be aggregated into a single packet is constrained by the maximum packet size, thus we have to carefully schedule information element transmissions so that the degree of packet packing (i.e., the amount of sensing data contained in packets) can be maximized without violating application requirement on the timeliness of data delivery. As a first step toward understanding the complexity of jointly optimizing INP and QoS with aggregation constraints, we analyze the impact that aggregation constraints have on the computational complexity of the problem, and we prove the following:

- When a packet can aggregate three or more information elements, the problem is strong NP-hard, and there is no polynomial-time approximation scheme (PTAS).
- When a packet can only aggregate two information elements, the complexity depends on whether two elements in a packet can be separated and re-packed with other elements on their way to the sink: if the elements in a packet can be separated, the problem is strong NP-hard and there is no PTAS for the problem; otherwise it can be solved in polynomial time by modeling the problem as a maximum weighted matching problem in an interval graph.
- The above conclusions hold whether or not the routing structure is a tree or a linear chain, and whether or not the information elements are of equal length.

Besides shedding light on the complexity and protocol design of jointly optimizing data delivery timeliness and packet packing (as well as other INP methods), these findings incidentally answer several open questions on the complexity of batch-process scheduling in interval graphs [18].

To understand the impact of jointly optimizing packet packing and data delivery timeliness, we design a distributed, online protocol *tPack* that schedules packet transmissions to maximize the local utility of packet packing at each node while taking into account the aggregation constraint imposed by the maximum packet size. Using a testbed of 130 TelosB motes, we experimentally evaluate the properties of tPack. We find that jointly optimizing data delivery timeliness and packet packing and considering real-world aggregation constraints significantly improve network performance (e.g. in terms of high reliability, high energy efficiency, and low delay jitter).

TABLE I
NOTATIONS USED IN SECTIONS II & III

Common notations	
K	maximum number of information elements allowed in a packet
$ETX_{v_i v_j}(l)$	expected number of transmissions taken to successfully deliver a packet of length l along link (v_i, v_j)
$t_{v_i v_k}(l)$	maximum time taken to deliver a packet of length l from v_i to v_k in the absence of packet packing and packing-oriented scheduling
Notations used in Section II only	
R	root of a directed collection tree
x	an information element
v_x	the node where x is generated
l_x	length of x
r_x	time when x is generated
d_x	deadline of delivering x to R
s_x	spare time in delivering x
$[r_x, d_x]$	lifetime of x
Notations used in Section III only	
n	number of variables in a SAT instance
m	number of clauses in a SAT instance
X_j	jth variable of a SAT instance
C_i	ith clause of a SAT instance
x_i^j	information element corresponding to the variable X_j in a clause C_i
$[r_i^j, d_i^j]$	lifetime of x_i^j
ax_k^j	kth auxiliary information element for variable X_j
$[r_{ax_k}^j, d_{ax_k}^j]$	lifetime of ax_k^j
z_i	information element generated by node v_i^c
$[r_i, d_i]$	lifetime of z_i
t_1	transmission time from any leaf node to its parent
t_2	transmission time from any node v_j to node v
t_3	transmission time from node v to node s
t_4	transmission time from any node v_i^c to node v

The rest of the paper is organized as follows. We analyze the benefits of packet packing in lossy wireless networks in Section II. We discuss the system model and precisely define the joint optimization problem in Section III. Then we analyze the complexity of the problem in Section IV, and present the tPack protocol in Section V. We experimentally evaluate the performance of tPack and study the impact of packet packing as well as joint optimization in Section VI. We discuss related work in Section VII, and conclude the paper in Section VIII. For convenience, we summarize in Table I the notations used in Sections III and IV.

II. WHY PACKET PACKING?

While aggregating short information elements reduces the overhead of transmitting each information element, it increases the length of packets being transmitted. Given that packet delivery rate of a wireless link decreases as packet length increases, a longer packet with aggregated information elements may be retransmitted more often, for reliable data delivery, than the shorter packets without aggregation. To understand whether packet packing is still beneficial in the presence of lossy wireless links, therefore, we need to understand whether the increased packet loss rate overshadows the benefits of packet packing. To this end, we mathematically analyze the issue as follows.

For simplicity, we assume in this section that the status (i.e., success or failure) of different packet transmissions are independent, and we corroborate the analytical results through testbed based measurement in Section VI where temporal link correlation exists. For convenience, we define the following notations:

l_1	:	payload length of an unpacked packet, i.e., the length of a single information element;
p_1	:	delivery rate of an unpacked packet;
k	:	packing ratio, i.e., the ratio of the payload length of a packed packet to that of an unpacked packet;
h	:	the ratio of header length to payload length in an unpacked packet.

Then, for a packed packet with packing ratio k, the ratio of the overall length of the packed packet to that of an unpacked packet is $\frac{kl_1+hl_1}{l_1+hl_1}$. Thus, the delivery rate p_k of the packed packet can be calculated as follows:

$$p_k = p_1^{\frac{kl_1+hl_1}{l_1+hl_1}} = p_1^{\frac{k+h}{1+h}}$$

To reflect the overhead of transmitting a packet pkt over a wireless link, we define the *amortized cost* (AC) of transmitting pkt as follows:

$$AC_{pkt} = \frac{ETX_{pkt}}{len_{pkt}} \tag{1}$$

where len_{pkt} is the payload length of pkt, and ETX_{pkt} is the expected number of transmissions taken to successfully deliver pkt over the wireless link. Given that the expected number of transmissions to successfully deliver a packet with packing ratio k is $\frac{1}{p_k}$, the amortized cost of transmitting a packet with packing ratio k, denoted by AC_k, can be calculated as follows:

$$AC_k = \frac{1/p_k}{kl_1} = \frac{1}{kl_1p_k} \tag{2}$$

Since an unpacked packet has a packing ratio of 1, the amortized cost of transmitting an unpacked packet is AC_1, that is, $\frac{1}{l_1p_1}$.

For a given packing ratio k, the ratio R_k of AC_1 to AC_k reflects whether packet packing is beneficial, that is, packet packing is beneficial if $R_k > 1$. Precisely, R_k is calculated as follows:

$$R_k = \frac{AC_1}{AC_k} = kp_1^{\frac{k-1}{1+h}}$$

In a typical sensornet system [15], [19], the ratio h of header length to that of a single information element is around 3, and the packing ratio can be up to 12. For $h = 3$, Figure 1 shows R_k as a function of p_1 and k, when $h = 3$. From the figure, we can see that packet packing reduces the amortized cost of packet transmission as long as the link reliability is no less than 40%, which is usually the case in practice (e.g., link reliability was ∼75% even in heavily loaded sensornet systems [15], [19]). We also see that, if link reliability is greater than 67%, the amortized cost of packet transmission

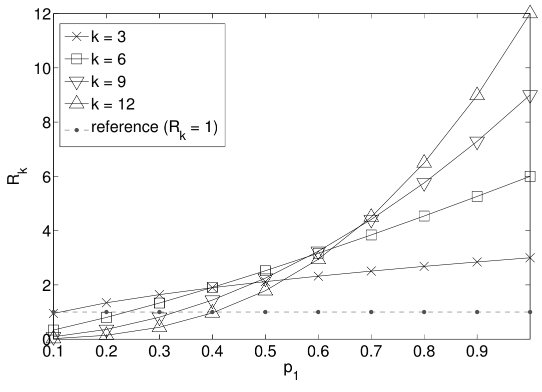


Fig. 1. $R_k = \frac{AC_1}{AC_k}$

always decreases as the packing ratio increases. Since link reliability is usually greater than 67% in practice, we can always try to maximize the packing ratio so that the amortized cost of packet transmission is reduced.

Denoting k^* as the optimal packing ratio that minimize the amortized cost for transmitting a packet, we then study the relationship between k^* and p_1. From (2), we have:

$$AC_k = \frac{1}{kl_1p_k} = \frac{1}{kl_1p_1^{\frac{k+h}{1+h}}} \tag{3}$$

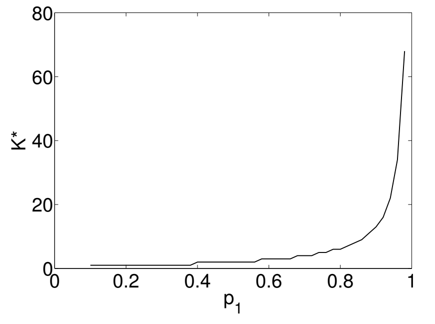
To minimize AC_k, we need to maximize $f(k) = kl_1p_1^{\frac{k+h}{1+h}}$. When $k \in R^+$, $f(k)$ is a convex function. Let $f'(k) = 0$. we have $k_R^* = \frac{1+h}{\ln p_1^{-1}}$. Therefore, when $k \in N^+$, k^* is calculated as follows:

$$k^* = \arg min_{k\in\{1,\lceil k_R^*\rceil,\lfloor k_R^*\rfloor\}}\{AC_k\} \tag{4}$$

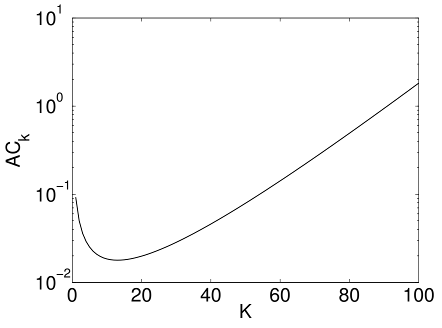
In Figure 2(b), k^* increases as the link reliability increases. When p_1 is greater than 75%, k^* increase faster, which implies that packet packing can bring more benefit on amortized cost when link reliability is high. Figure 2(b) shows the relationship between AC_k and k when $p_1 = 0.9$. From the figure we can find that it is not always beneficial to pack as many small packets as possible. There exists a threshold on the packing ratio. When k exceeds this threshold, the amortized cost will increase. This motivates us to explore how to perform packing at each node in the network.

Remarks. The above analysis focuses on a single link, but the observations easily carry over to multi-hop networks since link reliability p_1 reflects the impact of channel fading and collision even in the case of multi-hop networks.[1] The analysis has not considered the benefits (e.g., fewer number of packet collisions) of reduced channel contention as a result of packet packing (which reduces the number of packets contending for channel access). We will study the impact of these factors through testbed based measurement in Section VI.

[1]Note that the increased per-packet transmission time as a result of increased packet length will not cause more collision, since the time taken to transmit a packet (e.g., ∼ 4 milliseconds) is usually much less than the inter-packet interval (e.g., usually at least a few seconds).



(a) K^*



(b) AC_K

Fig. 2. K^* and AC_K when $l_1 = 12$, $h = 0.375$, and $K_{max} = 100$.

III. System model and problem definition

Having verified the benefits of packet packing in lossy wireless networks in Section II, we now discuss the system model and define the joint optimization problems we will focus on in this paper.

A. System model

We consider a directed collection tree $T = (V, E)$, where V and E are the set of nodes and edges in the tree. $V = \{v_i : i = 1 \dots N\} \cup \{R\}$ where R is the root of the tree and represents the data sink of a sensornet, and $\{v_i : i = 1 \dots N\}$ are the set of N sensor nodes in the network. An edge $\langle v_i, v_j \rangle \in E$ if v_j is the parent of v_i in the collection tree. The parent of a node v_i in T is denoted as p_i. We use $ETX_{v_i v_j}(l)$ to denote the expected number of transmissions required for delivering a packet of length l from a node v_i to its ancestor v_j, and we use $t_{v_i v_k}(l)$ to denote the maximum time taken to deliver a packet of length l from v_i to v_k in the absence of packet packing and packing-oriented scheduling.

Each information element x generated in the tree is identified by a 4-tuple (v_x, l_x, r_x, d_x) where v_x is the node that generates x, l_x is the length of x, r_x is the time when x is generated, and d_x is the deadline by which x needs to be delivered to the sink node R. We use $s_x = d_x - (r_x + t_{v_x R}(l_x))$ to denote the *spare time* for x, and we define the *lifetime of* x as $[r_x, d_x]$.

B. Problem definition

Given a collection tree T and a set of information elements $X = \{x\}$ generated in the tree, we define the problem of jointly optimizing packet packing and the timeliness of data delivery as follows:

Problem $\mathbb{P}$: given T and X, schedule the transmission of each element in X to minimize the total number of packet transmissions required for delivering X to the sink R while ensuring that each element be delivered to R before its deadline.

In an application-specific sensornet, the information elements generated by different nodes depend on the application but may well be of equal length [15]. Depending on whether the sensornet is designed for event detection or data collection, moreover, the information elements X may follow certain arrival processes. Based on the specific arrival process of X, the following special cases of problem $\mathbb{P}$ tend to be of practical relevance in particular:

Problem $\mathbb{P}_0$: same as $\mathbb{P}$ except that 1) the elements of X are of equal length, and 2) X includes at most one element from each node; this problem can represent sensornets that detect rare events.

Problem $\mathbb{P}_1$: same as $\mathbb{P}$ except that 1) the elements of X are of equal length, and 2) every two consecutive elements generated by the same node v_i are separated by a time interval whose length is randomly distributed in $[a, b]$; this problem can represent periodic data collection sensornets (with possible random perturbation to the period).

Problem $\mathbb{P}_2$: same as $\mathbb{P}$ except that the elements of X are of equal length; this problem represents general application-specific sensornets.

IV. Complexity of joint optimization

The complexity of problem $\mathbb{P}$ depends on aggregation constraints such as maximum packet size and whether information elements in a packet can be separated and repacked with other elements. For convenience, we use K to denote the maximum number of information elements that can be packed into a single packet. (Note that K depends on the maximum packet size and the lengths of information elements in problem $\mathbb{P}$.) In what follows, we first analyze the case when $K \geq 3$ and then the case when $K = 2$, and we discuss how aggregation constraints affect the problem complexity.

A. Complexity when $K \geq 3$

We first analyze the complexity and the hardness of approximation for problem $\mathbb{P}_0$, then we derive the complexity of $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}$ accordingly. The analysis is based on reducing the Boolean-satisfiability-problem (SAT) [20] to $\mathbb{P}_0$ as we show below.

Theorem 1: When $K \geq 3$, problem $\mathbb{P}_0$ is strong NP-hard whether or not the routing structure is a tree or a linear chain.

Proof: To prove that $\mathbb{P}_0$ is strong NP-hard, we first present a polynomial transformation f from the SAT problem to $\mathbb{P}_0$, then we prove that an instance Π of SAT is satisfiable if and only if the optimal solution of $\Pi' = f(\Pi)$ has certain minimum number of transmissions.

Given an instance Π of the SAT problem which has n Boolean variables $X_1, \dots, X_n$ and m clauses $C_1, \dots, C_m$, we derive a polynomial time transformation from Π to an instance Π' of $\mathbb{P}_0$ with $K \geq 3$ as follows. Firstly, we construct a tree with n+2 nodes shown in Figure 3. In this tree, each node v_j,

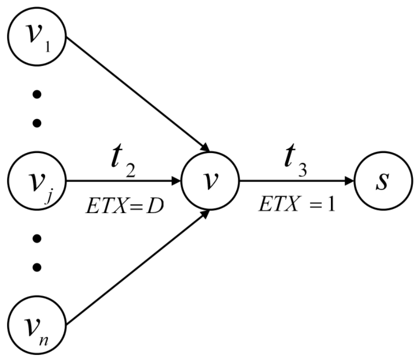


Fig. 3. A tree with $n+2$ nodes

where $j = 1, \ldots, n$ corresponds to the variable X_j. Node v is an intermediate node and node S is the base station. $ETX_{v_j v}$ is D, where $D \gg 1$, and ETX_{vs} is 1. (For now, we do not consider the impact of packet length on link reliability and thus ETX.) The transmission time $t_{v_j v} = t_2$ and $t_{vs} = t_3$. This operation takes $O(n)$ time.

Secondly, assume that variable X_j appears k_j times in total in the m clauses. Then we add $2k_j + 3$ children to node v_j, labeled as $v_0^j, \ldots, v_{2k_j+2}^j$, and m children to node v, labeled as $v_1^c, \ldots, v_m^c$. Each new edge has a ETX of 1. The transmission time from each child of v_j to v_j is t_1, and the transmission time from v_i^c to v is t_4. This operation takes $O(nm)$ time and the whole tree is shown in Figure 4.

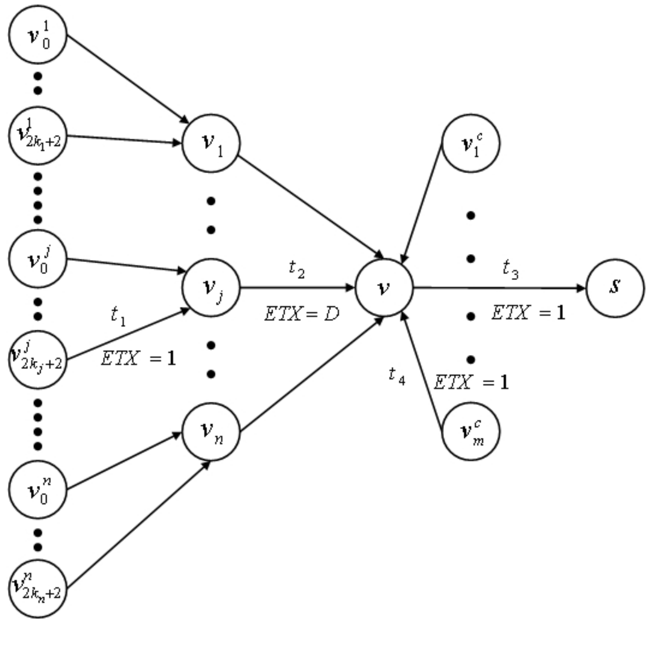


Fig. 4. Reduction from SAT to $\mathbb{P}_0$ when $K \geq 3$

After constructing the tree, we define the information elements and their lifetimes as follows. For each subtree rooted at node v_j, we first define $2k_j + 1$ information elements and then assign them one by one to the leaf nodes $v_1^j, \ldots, v_{2k_j+1}^j$ of this subtree. If variable X_j occurs unnegated in clause C_i, we create an information element x_i^j with lifetime $[r_i^j, d_i^j] = [(3i+1)(n+1)+j, (3i+2)(n+1)+j+t_1+t_2+t_3]$. If X_j occurs negated in clause C_i, we create an information element $x_i^j : [r_i^j, d_i^j] = [3i(n+1)+j, (3i+1)(n+1)+j+t_1+t_2+t_3]$. Let $i_1^j < \ldots < i_{k_j}^j$ denote the indices of the clauses in which variable X_j occurs. For every two messages $x_{i_t^j}^j$ and $x_{i_{t+1}^j}^j$, $t = 1, \ldots, k_j - 1$, define an information element $ax_{i_t^j}^j : [r_{a_t}^j, d_{a_t}^j] = [d_{i_t^j}^j - t_1 - t_2 - t_3, r_{i_{t+1}^j}^j + t_1 + t_2 + t_3]$. We also define $ax_0^j : [r_{a_0}^j, d_{a_0}^j] = [j, r_{i_1^j}^j + t_1 + t_2 + t_3]$, and $ax_{k_j}^j : [r_{a_{k_j}}^j, d_{a_{k_j}}^j] = [d_{i_{k_j}^j}^j - t_1 - t_2 - t_3, 3(m+1)(n+1)+j+t_1+t_2+t_3]$. In this way, every two consecutive information elements in this sequence overlap in their lifetimes, and the size of the overlap is $t_1+t_2+t_3$. After defining these $2k_j+1$ information elements, we set the source of each element one by one from node v_1^j to node $v_{2k_j+1}^j$. For each node v_0^j, we define an element $z_0^j : [j, j + t_1 + t_2 + t_3]$. For each node $v_{2k_j+2}^j$, we define an element $z_{2k_j+2}^j : [3(m+1)(n+1) + j, 3(m+1)(n+1) + j + t_1 + t_2 + t_3]$. Figure 5 demonstrates

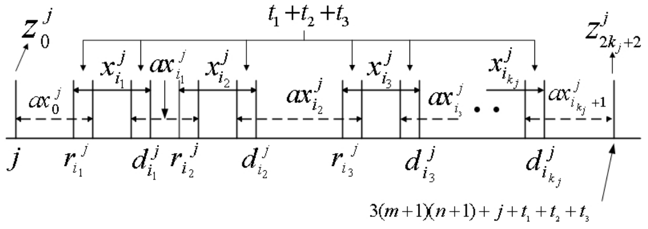


Fig. 5. Lifetimes of information elements

how the lifetimes of these $2k_j + 3$ information elements are defined.

Similarly, we define m information elements generated by nodes $v_1^c, \ldots, v_m^c$, with element $z_i : [r_i, d_i] = [(3i+1)(n+1)+t_1+t_2-t_4, (3i+2)(n+1)+t_1+t_2+t_3]$, $i = 1, \ldots, m$, being generated by node v_i^c. Then, for nodes v_1 to v_n, we define an information element for each of them with lifetime $[4(m+1)(n+1)+i, 4(m+1)(n+1)+i+t_2+t_3]$, $i = 1, \ldots, n$. For node v, define an information element with lifetime $[4(m+1)^2(n+1)+i, 4(m+1)^2(n+1)+i+t_3]$.

The whole process to assign an information element for each sensor will take $O(nm)$ time. Therefore, the time complexity of the whole transformation is $O(n) + O(nm) + O(nm) = O(nm)$, which is polynomial in n and m.

Given the instance Π' of $\mathbb{P}_0$ formulated as above, the following claims hold for the optimal packing scheme:

Claim 1: If nodes $v_1^c, \ldots, v_m^c$ are ignored, the minimum total number of transmission in Π' is $C_{t0} = \sum_{j=1}^{n}(2k_j + 1) + \sum_{j=1}^{n}[(k_j+1)(D+1)] + 2n(D+1) + 2n + 1$.

Claim 2: If nodes $v_1^c, \ldots, v_m^c$ are ignored, in the optimal packing scheme in Π', every information element q generated by a leaf node of node $v_j, j = 1, \ldots, n$, is forwarded to the source's parent at time r_q, and then leaves the parent to next hop either at time $r_q + t_1$ or at time $d_q - (t_2 + t_3)$.

Claim 3: If nodes $v_1^c, \ldots, v_m^c$ are ignored, in the optimal packing scheme in Π', for each $j = 1, \ldots, n$, all the information elements x_i^j leaves node v_j for v either at time $r_i^j + t_1$, or at the time $d_i^j - (t_2 + t_3)$.
Then, we have

Claim 4: The minimum number of transmissions required in Π', denoted by C_{t1}, is $C_{t0} + m$ if and only if the SAT problem Π is satisfiable.
(We relegate the proofs of these claims to the appendix.)

Then, Claim 4 and the fact that the reduction shown in Figure 4 is a polynomial reduction from SAT to $\mathbb{P}_0$ imply that $\mathbb{P}_0$ is strong NP-hard when $K \geq 3$.

Note that the above proof did not consider the impact of packet length on link reliability and thus ETX. As long as we construct the reduction so that the ETX along links $\langle v_j, v\rangle, j = 1, \ldots, n$ is significantly greater than that along link $\langle v, s\rangle$, however, the above analysis can be easily extended to and still hold for cases where ETX is a function of packet length.

We have also proved that $\mathbb{P}_0$ is NP-hard when the routing structure is a linear chain. For conciseness, we relegate the detailed discussion to the appendix.

■

Having proved the strong NP-hardness of $\mathbb{P}_0$ when $K \geq 3$, we analyze the hardness of approximation for $\mathbb{P}_0$ using a gap-preserving reduction from MAX-3SAT to $\mathbb{P}_0$ [21], and we have

Theorem 2: When $K \geq 3$, there exists $\epsilon \geq 1$ such that it is NP-hard to achieve an approximation ratio of $1 + \frac{1}{200N}(1 - \frac{1}{\epsilon})$ for problem $\mathbb{P}_0$, where N is the number of information elements in $\mathbb{P}_0$.

Proof: We first show that the reduction presented in Figure 4 is a gap-preserving reduction [21] from MAX-3SAT to problem $\mathbb{P}_0$. It is easy to verify that the proof of Theorem 1 holds if the discussion of the proof is based on 3SAT instead of the general SAT problem, in which case $\sum_{j=1}^{n} k_j = 3m$ and we denote the reduction as f. Therefore, if a 3SAT problem Π is satisfiable, the minimum cost of the $\mathbb{P}_0$ problem $\Pi' = f(\Pi)$ is

$$\begin{aligned} C_{t1} &= C_{t0} + m \\ &= (\textstyle\sum_{j=1}^{n}(2k_j + 1) + \sum_{j=1}^{n}(k_j + 1)(D + 1) + \\ &\quad 2n(D + 1) + 2n + 1) + m \\ &= m(3D + 10) + n(3D + 6) + 1 \end{aligned} \tag{5}$$

Since $n < 4m$, (5) implies that

$$\begin{aligned} C_{t1} &< m(3D + 10) + n(3D + 10) \\ &< 5m(3D + 10) \end{aligned} \tag{6}$$

Note that the proof of Theorem 1 holds if $D = n + \sum_{j=1}^{n}(2k_j + 3) = 6m + n$, which is the number of information elements generated by the descendants of node v. Thus, (6) implies that

$$\begin{aligned} C_{t1} &< 5m(3(6m + n) + 10) \\ &= 5m(18m + 3n + 10) \\ &< 5m(18m + 3 \times 4m + 10) \\ &= 5m(30m + 10) \\ &< 5m(30m + 10m) \\ &= 200m^2 \end{aligned} \tag{7}$$

If only m_0 of the m clauses in Π are satisfiable, then the minimum cost in $\Pi' = f(\Pi)$ (with $K \geq 3$ is $C_{t1} + m - m_0$. This is because $(m - m_0)$ number of z_i's cannot be packed with any other packet and have to be sent from node v to s alone, which incurs an extra cost of 1 each. Accordingly, if less than m_0 of the m clauses in Π are satisfiable, then the minimum cost C' in $\Pi' = f(\Pi)$ is greater than $C_{t1} + m - m_0$. Letting $\epsilon = \frac{m}{m_0}$, (7) implies that

$$\begin{aligned} \frac{C'}{C_{t1}} &> \frac{C_{t1} + m - m_0}{C_{t1}} \\ &= \frac{C_{t1} + \epsilon m_0 - m_0}{C_{t1}} \\ &= 1 + \frac{(\epsilon - 1)m_0}{C_{t1}} \\ &> 1 + \frac{(\epsilon - 1)m_0}{200m^2} \\ &= 1 + \frac{\epsilon - 1}{200m}\frac{1}{\epsilon} \\ &= 1 + \frac{1}{200m}(1 - \frac{1}{\epsilon}) \\ &\geq 1 + \frac{1}{200N}(1 - \frac{1}{\epsilon}) \end{aligned} \tag{8}$$

where N is the number of non-sink nodes in the network and $N \geq m$.

Let $OPT(\Pi)$ and $OPT(\Pi')$ be the optima of a MAX-3SAT problem Π and the corresponding $\mathbb{P}_0$ problem $\Pi' = f(\Pi)$. Then the polynomial-time reduction f from MAX-3SAT to $\mathbb{P}_0$ satisfy the following properties:

$$\begin{aligned} OPT(\Pi) = 1 &\implies OPT(\Pi') = C_{t1} \\ OPT(\Pi) < \tfrac{1}{\epsilon} &\implies OPT(\Pi') > C_{t1}(1 + \tfrac{1}{200N}(1 - \tfrac{1}{\epsilon})) \end{aligned} \tag{9}$$

From [21], we know that there exists a polynomial-time reduction f_1 from SAT to MAX-3SAT such that, for some fixed $\epsilon > 1$, reduction f_1 satisfies

$$\begin{aligned} I \in SAT &\implies \text{MAX-3SAT}(f_1(I)) = 1 \\ I \notin SAT &\implies \text{MAX-3SAT}(f_1(I)) < \tfrac{1}{\epsilon} \end{aligned} \tag{10}$$

Then, (9) and (10) imply the following:

$$\begin{aligned} I \in SAT &\implies OPT(f(f_1(I))) = C_{t1} \\ I \notin SAT &\implies OPT(f(f_1(I))) > C_{t1}(1 + \tfrac{1}{200N}(1 - \tfrac{1}{\epsilon})) \end{aligned} \tag{11}$$

Therefore, it is NP-hard to achieve an approximation ratio of $1 + \frac{1}{200N}(1 - \frac{1}{\epsilon})$ for problem $\mathbb{P}_0$.

■

Based on the definition of polynomial time approximation scheme (PTAS) and Theorem 2, we then have

Corollary 1: There is no polynomial time approximation scheme (PTAS) for problem $\mathbb{P}_0$ when $K \geq 3$.

Based on the findings for $\mathbb{P}_0$, we have

Theorem 3: When $K \geq 3$, problems $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}$ are strong NP-hard whether or not the routing structure is a tree or a linear chain, and there is no polynomial-time approximation scheme (PTAS) for solving them.

Proof: To prove the hardness results for $\mathbb{P}_1$, let's consider a special case Π_1 of $\mathbb{P}_1$ where 1) every node is generating information elements using the same period p_0 and the same spare time s_0 for information elements, 2) p_0 is significantly larger than s_0, and 3) p_0 is significantly larger than the latest time r_0 when a node generates its first information element such that the following holds: in the optimal packing scheme for Π_1, no two elements from the same node can be aggregated into the same packet, and the i-th information element from one node cannot be packed with the j-th element from another node unless $i = j$. It is easy to see that the special case Π_1 does exist by properly choosing the parameters p_0, s_0, and r_0. Therefore, solving Π_1 becomes the same as solving an instance Π_0 of $\mathbb{P}_0$ where the information elements consist of the first element from every node of Π_1. Therefore, $\mathbb{P}_1$ is at least as hard as $\mathbb{P}_0$. Since $\mathbb{P}_0$ is strong NP-hard, $\mathbb{P}_1$ is strong NP-hard, and the there is no PTAS for the problem.

Since $\mathbb{P}_1$ is a special case of $\mathbb{P}_2$, and $\mathbb{P}_2$ is a special case of $\mathbb{P}$, both $\mathbb{P}_2$ and $\mathbb{P}$ are strong NP-hard too, and there is no PTAS for them.

■

Theorems 1 and 3 show that the joint optimization problems are strong NP-hard and there is no PTAS, whether or not the routing structure is a tree or a linear chain and whether or not the information elements are of equal length. In contrast, Becchetti $et\ al.$ [11] showed that, for total aggregation, the joint optimization problems are solvable in polynomial time via dynamic programming on chain networks. Therefore, we see that aggregation constraints make the difference on whether a problem is tractable for certain networks, and thus it is important to consider them in the joint optimization. Incidentally, we note that Theorem 3 also answers the open question on the complexity of Problem (P4) of batch-process scheduling in interval graphs [18].

B. Complexity when $K = 2$

We showed in Section IV-A that the problems $\mathbb{P}_i, i = 0, 1, 2,$ and $\mathbb{P}$ are all strong NP-hard and there is no PTAS for these problems when $K \geq 3$. We prove in this section that, when $K = 2$, the complexity of these problems depends on whether information elements in a packet can be separated and re-packed with other elements (which we call *re-aggregation* hereafter) on their way to the sink. When re-aggregation is disallowed, these problems are solvable in polynomial time; otherwise they are strong NP-hard. Note that, when $K \geq 3$, these problems are all strong NP-hard even if re-aggregation is disallowed, which can be seen from the proof of Theorem 1. Note also that, even though re-aggregation may well be allowed in most sensornet systems when the in-network processing (INP) method is packet packing, re-aggregation may not be possible or allowed when INP is data fusion such as lossy data compression [22]. Via the study on the impact of re-aggregation, therefore, we hope to shed light on the structure of the joint optimization problems when general INP methods are considered.

In what follows, we first analyze the case when re-aggregation is allowed, then we analyze the case when re-aggregation is disallowed.

1) When re-aggregation is allowed: Use a method similar to that of Theorem1, we prove

Theorem 4: When $K = 2$ and re-aggregation is allowed, problem $\mathbb{P}_0$ is strong NP-hard, and this result holds whether or not the routing structure is a tree or a linear chain.

Proof: Given an instance Π of SAT problem with n Boolean variables $X_1, \ldots, X_n$ and m clauses $C_1, \ldots, C_m$, we derive a polynomial time transformation from Π to an instance Π'' of problem $\mathbb{P}_0$ with $K = 2$ as follows. The transformation is the same as what we present through Figure 4 except for the following changes:

- Define a node p between node v and node s, and m children $p_1, \ldots, p_m$ of node p. Additionally, define $ETX_{vp} = ETX_{ps} = ETX_{p_ip} = 1$, and $t_{vp} = t_3, t_{ps} = t_5$, and $t_{p_ip} = t_6$.
- Define m information elements g_i's generated by nodes $p_1, , p_m$: $g_i : [r_i^p, d_i^p] = [(3i+1)(n+1)+n+0.1+t_1+t_2+t_3-t_6, (3i+1)(n+1)+n+0.1+t_1+t_2+t_3+t_5]$, and for node p, define an information element g with lifetime $[5(m+1)^2(n+1)+i, 5(m+1)^2(n+1)+i+t_5]$.
- For all parameters defined during the transformation in Figure 4, replace t_3 by $t_3 + t_5$.

Therefore, the time complexity of the new transformation is still $O(nm)$, and the new reduction is shown in Figure 6.

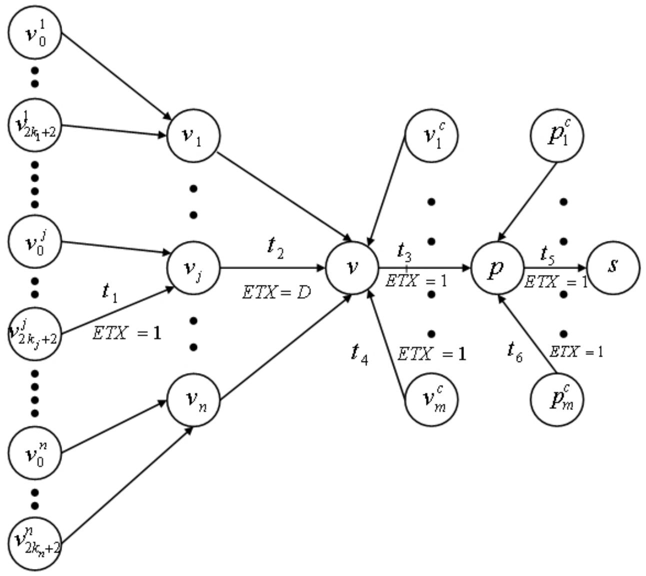


Fig. 6. Reduction from SAT to $\mathbb{P}_0$ when $K = 2$

Then, the following claims hold for Π'':

Claim 5: If nodes $v_1^c, \ldots, v_m^c$, and nodes $p_1, \ldots, p_m$ are ignored, the minimum number of transmissions in Π'' is $C'_{t0} = \sum_{j=1}^{n}(2k_j+1)+\sum_{j=1}^{n}[(k_j+1)(D+2)]+2n(D+2)+2n+3$.

Claim 6: If nodes $v_1^c, \ldots, v_m^c$, and nodes $p_1, \ldots, p_m$ are ignored, in the optimal packing scheme of Π'', every information element q generated by a leaf node of node v_j,$j = 1, \ldots, n$, is forwarded to the source's parent at time r_q, and then leaves the parent to next hop either at time $r_q + t_1$, or at time $d_q - (t_2 + t_3 + t_5)$.

Claim 7: If nodes $v_1^c, \ldots, v_m^c$, and nodes $p_1, \ldots, p_m$ are ignored, in the optimal packing scheme of Π'', for each $j = 1, \ldots, n$, all the information elements x_i^j leave node v_j for v either at time $r_i^j + t_1$, or at time $d_i^j - (t_2 + t_3 + t_5)$.

These claims can be proved in the same way as how Claims 1, 2, and 3 are proved respectively, and we skip the details here. Then, we have

Claim 8: The minimal number of transmissions required in Π'', denoted by C'_{t1}, is $C'_{t0} + 4m$ if and only if the SAT problem Π is satisfiable.

(We relegate the proof to the appendix.)

Then, Claim 8 and the fact that the reduction shown in Figure 6 is polynomial imply that $\mathbb{P}_0$ is strong NP-hard when $K = 2$.

Note that the above proof did not consider the impact of packet length on link reliability and thus ETX. As long as we construct the reduction so that the ETX along links $\langle v_j, v\rangle, j = 1, \ldots, n$ is significantly greater than that along links $\langle v, p\rangle$ and $\langle p, s\rangle$, however, the above analysis can be easily extended to and still hold for cases where ETX is a function of packet length.

Note also that the above proof can be extended to the case when all the information elements are generated at the same time, as well as the case when the routing structure is a linear chain (with information elements having different generation time).

■

Then, we prove the hardness of approximation using a gap-preserving reduction from MAX-3SAT, and we have

Theorem 5: When $K = 2$ and re-aggregation is allowed, there exists $\epsilon \geq 1$ such that it is NP-hard to achieve an approximation ratio of $1 + \frac{1}{120N}(1 - \frac{1}{\epsilon})$ for problem $\mathbb{P}_0$, where N is the number of information elements in $\mathbb{P}_0$.

Proof: The proof is similar to that of Theorem 2. We relegate the details to the appendix. ■

Based on the definition of polynomial time approximation scheme (PTAS) and Theorem 5, we then have

Corollary 2: There is no polynomial time approximation scheme (PTAS) for problem $\mathbb{P}_0$ when $K = 2$ and re-aggregation is allowed.

Based on the relations among $\mathbb{P}_0$, $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}$, we have

Theorem 6: When $K = 2$ and re-aggregation is allowed, problems $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}$ are strong NP-hard whether or not the routing structure is a tree or a linear chain, and there is no polynomial-time approximation scheme (PTAS) for solving them.

Proof: The proof is similar to that of Theorem 3. ■

Theorems 4 and 6 show that, when $K = 2$ and re-aggregation is allowed, the joint optimization problems are strong NP-hard whether or not the routing structure is a tree or a linear chain, and whether or not the information elements are of the same length. That is, the complexity of these problems when $K = 2$ and re-aggregation is allowed is very much similar to the case when $K \geq 3$.

2) When re-aggregation is prohibited: When $K = 2$ and re-aggregation is prohibited, we can solve problem $\mathbb{P}$ (and thus its special versions $\mathbb{P}_0$, $\mathbb{P}_1$, and $\mathbb{P}_2$) in polynomial time by transforming it into a maximum weighted matching problem in an interval graph. An interval graph G_I is a graph defined on a set I of intervals on the real line such that 1) G_I has one and only one vertex for each interval in the set, and 2) there is an edge between two vertices if the corresponding intervals intersect with each other. Given an instance of problem $\mathbb{P}$, we solve it using Algorithm 1 as follows:

Algorithm 1 Algorithm for solving $\mathbb{P}$ when $K = 2$ and re-aggregation is prohibited

```
1: Generate an interval graph $G_I(V_I, E_I)$ for problem $\mathbb{P}$ as follows:
   • Select an arbitrary information element $q$ generated by node $v_q$ at time $r_q$ and with spare time $s_q$, define an interval $[r_q, r_q + s_q]$ for $q$ on the real line.
   • For each remaining information element $p$ generated by node $v_p$ at time $r_p$ and with spare time $s_p$, let node $v_{pq}$ be the common ancestor of $v_p$ and $v_q$ that is the farthest away from $R$ among all common ancestors of $v_p$ and $v_q$, then define an interval $[r_q - t_{v_q v_{pq}} + t_{v_p v_{pq}}, r_q - t_{v_q v_{pq}} + t_{v_p v_{pq}} + s_q]$ for information element $p$.
   • Let $V_I = \emptyset$. Then, for each information element $s$, define a vertex $s$ and add it to $V_I$.
   • Let $E_I = \emptyset$. If the two intervals that represent any two information elements $u$ and $h$ overlap with each other, define an edge $(u, h)$ and add it to $E_I$; then assign edge $(u, h)$ with a weight $com(u, h) = ETX_{v_{uh}R}(l_u) + ETX_{v_{uh}R}(l_h) - ETX_{v_{uh}R}(l_u + l_h)$, where $l_u$ and $l_h$ are the length of $u$ and $h$ respectively.
2: Solve the maximum weighted matching problem for $G_I$ using Edmonds' Algorithm [23].
3: For each edge $(u, h)$ in the matching, information elements $u$ and $h$ are packed together at node $v_{uv}$. For all other vertices not in the matching, their corresponding information elements are sent to the sink alone without being packed with any other information element.
```

For Algorithm 1, we have

Theorem 7: When $K = 2$ and re-aggregation is prohibited, Algorithm 1 solves problem $\mathbb{P}$ in $O(n^3)$ time, where n is the number of information elements considered in the problem.This holds whether or not the routing structure is a tree or a linear chain, and whether or not the information elements are of equal length.

Proof: It is easy to see that if information elements u and h are packed together, the total number of transmissions taken to deliver u and h is $ETX_{v_u R}(l_u) + ETX_{v_h R}(l_h) - ETX_{v_{uh} R}(l_u) - ETX_{v_{uh} R}(l_h) + ETX_{v_{uh} R}(l_u + l_h) = ETX_{v_u R}(l_u) + ETX_{v_h R}(l_h) - com(u, h)$. Let V_I be the set of vertices in the interval graph G_I, M be a matching in G_I, V_1 be the set of nodes in M, and $V_2 = V_I / V_1$. Then the weight of M, denoted by W_M, is as follows:

$$\begin{array}{rcl} W_M & = & \sum_{(u,h)\in M} com(u,h) \\ & = & \sum_{(u,h)\in M}[ETX_{v_u R}(l_u) + ETX_{v_h R}(l_h) \\ & & -(ETX_{v_u R}(l_u) + ETX_{v_h R}(l_h) \\ & & -com(u,h))] \\ & = & \sum_{(u,h)\in M}(ETX_{v_u R}(l_u) + ETX_{v_h R}(l_h)) \\ & & -\sum_{(u,h)\in M}[ETX_{v_u R}(l_u) + ETX_{v_h R}(l_h) \\ & & -com(u,h)] \\ & = & \sum_{s\in V_1} ETX_{sR}(l_s) + \sum_{v\in V_2} ETX_{vR}(l_v) \\ & & -\{\sum_{(u,h)\in M}[ETX_{v_u R}(l_u) + ETX_{v_h R}(l_h) \\ & & -com(u,h)] \\ & & +\sum_{v\in V_2} ETX_{vR}(l_v)\} \\ & = & \sum_{v\in V_I} ETX_{vR}(l_v) \\ & & -\{\sum_{(u,h)\in M}[ETX_{v_u R}(l_u) + ETX_{v_h R}(l_h) \\ & & -com(u,h)] + \sum_{v\in V_2} ETX_{vR}(l_v)\} \end{array} \tag{12}$$

Note that $\sum_{v \in V_I} ETX_{vR}(l_v)$ is a fixed value, and $\sum_{(u,h)\in M}[ETX_{v_uR}(l_u) + ETX_{v_hR}(l_h) - com(u,h)] + \sum_{v \in V_2} ETX_{vR}(l_v)$ is the total number of transmissions, denoted by ETX_{total}, incurred in the packing scheme generated by Algorithm 1. Therefore, ETX_{total} is minimized if and only if W_M is maximized, which means that solving the maximum weighted matching problem can give us an optimal solution to the original packet packing problem.

Let n denote the total number of information elements in this problem. The whole algorithm consists of three parts. The first one is to define an interval graph and assign weights to each node and edge in the graph, whose time complexity is $O(n^2)$. The second part is to solve the maximum weighted matching problem, whose time complexity is $O(n^3)$ by Edmonds' Algorithm [23]. And the third part is to convert the optimal matching problem to the optimal packing scheme, whose time complexity is $O(n)$. Therefore, the time complexity of the whole algorithm is $O(n^2) + O(n^3) + O(n) = O(n^3)$. ■

By the definition of the weight $com(u, h)$ for elements u and h in Algorithm 1, the solution generated by the maximum weighted matching tends to greedily pack elements as soon as possible after they are generated. This observation motivates us to design a local, greedy online algorithm *tPack* in Section V for the general joint optimization problems, and the effectiveness of this approach will be demonstrated through competitive analysis and testbed-based measurement study in Sections V and VI. Note that, incidentally, Theorem 7 also answers the open question on the complexity of scheduling batch-processes with release times in interval graphs [18].

V. A UTILITY-BASED ONLINE ALGORITHM

We see from Section IV that problem $\mathbb{P}$ and its special cases in sensornets are strong NP-hard in most system settings, and there is no polynomial-time approximation scheme (PTAS) for these problems. Instead of trying to find global optimal solution, therefore, we focus on designing a distributed, approximation algorithm *tPack* that optimizes the local utility of packet packing at each node. Given that packet arrival processes are usually unknown a priori, we consider the online version of the optimization problem.

Based on the definition of $\mathbb{P}$, its optimization objective is to minimize

$$AC = \frac{TX_{net}}{\sum_{x \in X} l_x} \tag{13}$$

where TX_{net} is the total number of transmissions taken to deliver each information element $x \in X$ to the sink before its deadline. For convenience, we call AC the *amortized cost* of delivering $\sum_{x \in X} l_x$ amount of data. In what follows, we design an online algorithm tPack based on this concept of amortized cost of data transmission.

When node j has a packet pkt in its data buffer, j can decide to transmit pkt immediately or to hold it. If j transmits pkt immediately, information elements carried in pkt may be packed with packets at j's ancestors to reduce the amortized cost of data transmissions from those nodes; if j holds pkt, more information elements may be packed with pkt so that the amortized cost of transmission from j can be reduced. Therefore, we can define the *utility* of transmitting or holding pkt as the expected reduction in amortized data transmission cost as a result of the corresponding action, and then the decision on whether to transmit or to hold pkt depends on the utilities of the two actions. For simplicity and for low control overhead, we only consider the immediate parent of node j when computing the utility of transmitting pkt. We will show the goodness of this local approach through competitive analysis later in this section and through testbed-based measurement in Section VI.

In what follows, we first derive the utilities of holding and transmitting a packet, then we present a scheduling rule that improves the overall utility.

A. Utility calculation

For convenience, we define the following notations:

L	:	maximum payload length per packet;
$ETX_{jp}(l)$	:	expected number of transmissions taken to transport a packet of length l from node j to its ancestor p;
p_j	:	the parent of node v_j in the routing tree.

The utilities of holding and transmitting a packet pkt at a node v_j depend on the following parameters related to traffic pattern:

- With respect to v_j itself and its children:

r_l	:	expected rate in receiving another packet pkt' from a child or locally from an upper layer;
s_l	:	expected payload size of pkt'.

- With respect to the parent of v_j:

r_p	:	expected rate for the parent to transmit another packet pkt'' that does not contain information elements generated or forwarded by v_j itself;
s_p	:	expected payload size of pkt''.

The utilities of holding and transmitting a packet pkt also depend on the following constraints posed by timeliness requirement for data delivery as well as limited packet size:

- Grace period t'_f for delivering pkt: the maximum allowable latency in delivering pkt minus the maximum time taken to transport pkt from v_j to the sink without being held at any intermediate node along the route.
  If $t'_f \le 0$, pkt should be transmitted immediately to minimize the extra delivery latency.
- Spare packet space s'_f of pkt: the maximum allowable payload length per packet minus the current payload length of pkt.
  Parameter s'_f and the size of the packets coming next from an upper layer at v_j or from v_j's children determine how much pkt will be packed and thus the potential utility of locally holding pkt.

In the design and analysis of this section, we assume that packet arrival process (i.e.,, r_l, r_p), packet payload size and spare space (i.e., s_l, s_p, s'_f), and grace period (i.e., t'_f) are independent of one another. Then, the utilities of holding and transmitting a packet are calculated as follows.

Utility of holding a packet. When a node v_j holds a packet pkt, pkt can be packed with incoming packets from v_j's children or from an upper layer at v_j. Therefore, the utility of holding pkt at v_j is the expected reduction in the amortized cost of transmitting pkt after packing pkt. The utility depends on (a) the expected number of packets that v_j will receive within t'_f time (either from a child or locally from an upper layer), and (b) the expected payload size s_l of these packets. Given that the expected packet arrival rate is r_l, the expected number of packets to be received at v_j within t'_f time is $t'_f r_l$. Thus, the expected overall size $\mathcal{S}'_l$ of the payload to be received within t'_f time is

$$\mathcal{S}'_l = \frac{t'_f}{t_l} s_l$$

Given the spare space s'_f in the packet pkt, the expected size $\mathcal{S}_l$ of the payload that can be packed into pkt can be approximated[2] as

$$\mathcal{S}_l = \min\{\mathcal{S}'_l, s'_f\} = \min\{\frac{t'_f}{t_l} s_l, s'_f\}$$

Therefore, the expected amortized cost AC_l of transporting the packet to the sink R after the anticipated packing can be approximated as[2]

$$AC_l = \frac{1}{L - s'_f + \mathcal{S}_l} ETX_{jR}(L - s'_f + \mathcal{S}_l)$$

where $(L - s'_f)$ is the payload length of pkt before packing.

Since the amortized cost AC'_l of transporting pkt without the anticipated packing is

$$AC'_l = \frac{1}{L - s'_f} ETX_{jR}(L - s'_f)$$

the utility U_l of holding pkt is

$$U_l \quad = \quad AC'_l - AC_l \tag{14}$$

Utility of immediately transmitting a packet. If node v_j transmits the packet pkt immediately to its parent p_j, the utility comes from the expected reduction in the amortized cost of packet transmissions at p_j as a result of receiving the payload carried by pkt. When v_j transmits pkt to p_j, the grace period of pkt at p_j is still t'_f, thus the expected number of packets that do not contain information elements from v_j and can be packed with pkt at p_j is $t'_f r_p$, and we use P_{pkt} to denote this set of packets. Given the limited payload that pkt carries, it may happen that not every packet in P_{pkt} gets packed (to full) via the payload from pkt. Accordingly, the utility U_p of immediately transmitting pkt is calculated as follows:

- If *every packet in P_{pkt} gets packed to full* with payload from pkt, i.e., $t'_f r_p(L - s_p) \leq L - s'_f$:

  Then, the overall utility U'_p can be approximated as[2]

$$\begin{aligned} U'_p &= \frac{\frac{t'_f}{t_p} ETX_{p_jR}(s_p)}{\frac{t'_f}{t_p} s_p} - \frac{\frac{t'_f}{t_p} ETX_{p_jR}(L)}{\frac{t'_f}{t_p} L} \\ &= \frac{ETX_{p_jR}(s_p)}{s_p} - \frac{ETX_{p_jR}(L)}{L} \end{aligned} \tag{15}$$

- If *not every packet in P_{pkt} gets packed to full* with payload from pkt, i.e., $t'_f r_p(L - s_p) > L - s'_f$:

  In this case, $\lfloor \frac{L-s'_f}{L-s_p} \rfloor$ number of packets are packed to full; if mod$(L - s'_f, L - s_p) > 0$, there is also a packet that gets partially packed with mod$(L - s'_f, L - s_p)$ length of payload from pkt. Thus the total number of packets that benefit from the packet transmission is $\lceil \frac{L-s'_f}{L-s_p} \rceil$. Denoting mod$(L - s'_f, L - s_p)$ by l_{mod} and letting I_{mod} be 1 if $l_{mod} > 0$ and 0 otherwise, then the overall utility U''_p can be approximated as[2]

$$\begin{aligned} U''_p \quad = \quad & \frac{\lceil \frac{L-s'_f}{L-s_p} \rceil ETX_{p_jR}(s_p)}{\lceil \frac{L-s'_f}{L-s_p} \rceil s_p} - \\ & \frac{\lfloor \frac{L-s'_f}{L-s_p} \rfloor ETX_{p_jR}(L) + I_{mod} ETX_{p_jR}(s_p + l_{mod})}{\lceil \frac{L-s'_f}{L-s_p} \rceil s_p + L - s'_f} \end{aligned} \tag{16}$$

Therefore, the utility U_p of immediately transmitting pkt to p_j can be computed as

$$U_p = \left\{ \begin{array}{ll} U'_p & \text{if } t'_f r_p(L - s_p) \leq L - s'_f \\ U''_p & \text{otherwise} \end{array} \right. \tag{17}$$

where U'_p and U''_p are defined in Equations (15) and (16) respectively.

B. Scheduling rule

Given a packet to be scheduled for transmission, if the probability that the packet is immediately transmitted is P_t $(0 \leq P_t \leq 1)$, then the expected utility $U_t(P_t)$ is

$$\begin{aligned} U_t(P_t) &= P_t \times U_p + (1 - P_t) U_l \\ &= U_l + P_t(U_p - U_l) \end{aligned} \tag{18}$$

where U_p and U_l are the utilities of immediately transmitting and locally holding the packet respectively. To maximize U_t, P_t should be set according to the following rule:

$$P_t = \left\{ \begin{array}{ll} 1 & \text{if } U_p > U_l \\ 0 & \text{otherwise} \end{array} \right.$$

That is, the packet should be immediately transmitted if the utility of immediate transmission is greater than that of locally holding the packet. For convenience, we call this local, distributed decision rule *tPack* (for *time-sensitive packing*). Interested readers can find the discussion on how to implement tPack in TinyOS in [24].

Competitive analysis. To understand the performance of tPack as compared with an optimal online algorithm, we analyze the competitive ratio of tPack. Since it is difficult to analyze the competitive ratio of non-oblivious online algorithms for arbitrary network and traffic pattern in the joint optimization and tPack is a non-oblivious algorithm, we only study the competitive ratio of tPack for complete binary

[2]We use this approximation because it is usually difficult to estimate and store the complete distributions of random variables in resource-constrained sensor nodes.

trees where all the leaf nodes generate information elements according to a common data generation process, and we do not consider the impact of packet length on link ETX. We denote these special cases of problem $\mathbb{P}$ as problem $\mathbb{P}'$. The theoretical analysis here is to get an intuitive understanding of the performance of tPack; we experimentally analyze the behaviors of tPack with different networks, traffic patterns, and application requirements through testbed-based measurement in Section VI. We relegate the study on the competitive ratio of tPack as well as the lower bound on the competitive ratio of non-oblivious online algorithms for the general problem $\mathbb{P}$ as a part of our future work. (Note that the best results so far on the lower bound of the competitive ratio of joint INP- and latency- optimization also only considered the cases where only leaf nodes generate information elements [12], and these results are for oblivious algorithms and for cases where no aggregation constraint is considered [12].)

Then, we have

Theorem 8: For problem $\mathbb{P}'$, tPack is $\min\{K, \max_{v_j \in V_{>1}} \frac{2ETX_{v_jR}}{2ETX_{v_jR}-ETX_{p_jR}}\}$-competitive, where K is the maximum number of information elements that can be packed into a single packet, $V_{>1}$ is the set of nodes that are at least two hops away from the sink R.

Proof: For convenience, we denote the optimal packing scheme as OPT. By definition, tPack is at least K-competitive since, considering the packets transmitted by a given node v_i in the routing tree, the length of the packet containing an information element x in OPT is no more than K times the length of the packet containing x in tPack.

To get a tighter performance bound for tPack, we first analyze the packet length for the packets transmitted by a leaf node v_j. Suppose that v_j transmits a packet pkt with length l_{pkt} when the latency requirement could have allowed packing another l' amount of data with the packet. In this case, the utility of holding pkt is

$$U_l = \frac{ETX_{v_jR}}{l_{pkt}} - \frac{ETX_{v_jR}}{l_{pkt}+l'} = ETX_{v_jR}\frac{l'}{l_{pkt}(l_{pkt}+l')} \quad (19)$$

By definition, the utility of immediately transmitting pkt is no more than the transmission utility that would be generated if the information elements of pkt are all packed into another packet pkt^* at p_j, the parent of v_j, that was transmitted to p_j from its the child other than v_j. Given that the routing tree is a complete binary tree and that the leaf nodes generate information elements according to a common data generation process, the lengths of packets that are transmitted along links at the same tree level are expected to be the same. Thus we can assume that the payload length of pkt^* is also l_{pkt}. Therefore, the utility of immediately forwarding pkt at v_j satisfy the following inequality

$$U_p \le \frac{ETX_{p_jR}}{l_{pkt}} - \frac{ETX_{p_jR}}{l_{pkt}+l_{pkt}} = \frac{ETX_{p_jR}}{2l_{pkt}} \quad (20)$$

By the design of tPack, we know that $U_l < U_p$. From (19) and (20), thus we have

$$ETX_{v_jR}\frac{l'}{l_{pkt}(l_{pkt}+l')} < \frac{ETX_{p_jR}}{2l_{pkt}}$$

Thus

$$l' < \frac{a}{2-a}l_{pkt} \quad (21)$$

where $a = \frac{ETX_{p_jR}}{ETX_{v_jR}}$.

Due to the constraint imposed by application's requirement on the timeliness of data delivery, we know that the length of the packet, denoted by l_{opt}, that contains the information elements of pkt in OPT is no more than $l_{pkt}+l'$. Then from (21), we know that

$$l_{opt} \le l_{pkt}+l' < \frac{2}{2-a}l_{pkt} = \frac{2ETX_{v_jR}}{2ETX_{v_jR}-ETX_{p_jR}}l_{pkt}$$

That is,

$$\frac{l_{opt}}{l_{pkt}} < \frac{2ETX_{v_jR}}{2ETX_{v_jR}-ETX_{p_jR}} \quad (22)$$

For a node v_i that is not a leaf node, the same analysis applies. Given a packet pkt' of length $l_{pkt'}$ that is transmitted by v_i when the latency requirement could have allowed packing another l'' amount of data with pkt', we have

$$l'' < \frac{a'}{2-a'}l_{pkt'} \quad (23)$$

where $a' = \frac{ETX_{p_iR}}{ETX_{v_iR}}$. Moreover, the length of the packet, denoted by $l_{opt'}$, that contains the information elements of pkt' in OPT is no more than $l_{pkt'}+l''$; this is due to the following reasons:

- If a packet pkt_{max} contains $l_{pkt'}+l''$ amount of data payload without constrained by packet size limit, then the spare time of pkt_{max} is 0.
- Consider a packet pkt'' transmitted by v_i in OPT whose length is $l_{opt'}$. If v_i holds pkt'' until its spare time is 0 (instead of transmitting pkt'') in OPT, the resulting length of the new packet pkt''_0 is no more than $l_{pkt'}+l''$. This is because data flows faster toward the sink in tPack as compared with OPT, and pkt' reaches v_i earlier than pkt'' does.
- Therefore, $l_{opt'}$ is no more than the length of pkt''_0, which is no more than $l_{pkt'}+l''$. Thus, $l_{opt'} \le l_{pkt'}+l''$

Therefore, we have

$$\frac{l_{opt'}}{l_{pkt'}} < \frac{2ETX_{v_iR}}{2ETX_{v_iR}-ETX_{p_iR}} \quad (24)$$

From (22) and (24), we know that tPack is at least $O(\max_{v_j \in V_{>1}} \frac{2ETX_{v_jR}}{2ETX_{v_jR}-ETX_{p_jR}})$-competitive. Therefore, tPack is $\min\{K, \max_{v_j \in V_{>1}} \frac{2ETX_{v_jR}}{2ETX_{v_jR}-ETX_{p_jR}}\}$-competitive for problem $\mathbb{P}'$.

■

From Theorem 8, we see that tPack is 2-competitive if every link in the network is of equal ETX value.

C. Implementation

From the discussion in Section V-A, a node v_j needs to obtain the following parameters when calculating the utilities of holding and transmitting a packet:

- On routing tree: $ETX_{jR}(l)$, p_j, and $ETX_{p_jR}(l)$;
- On traffic pattern: r_l, s_l, r_p, s_p, and K.

Parameters related to routing tree can be provided by the routing component in a given system platform. Given a link $\langle j,p\rangle$, $ETX_{jp}(l)$ as a function of packet length l can be estimated using $ETX_{jp}(1)$, the ETX value of transmitting a packet of one unit length, as follows:

$$ETX_{jp}(l) = 1/(\frac{1}{ETX_{jp}(1)})^l = ETX_{jp}(1)^l$$

Accordingly, the routing component only needs to estimate $ETX_{jp}(1)$ instead of the ETX values for packets of arbitrary length.

For parameters related to traffic pattern, v_j can estimate by itself the parameters r_l and s_l, and K is readily available and fixed for each specific platform. To enable each node v_j to obtain parameters r_p and s_p, every node i in the network estimates the expected rate r_i to transmit two consecutive packets at i itself and the expected size s_i of these packets. Then, every node i shares with its neighbors the parameters r_i and s_i by piggybacking these information onto data packets or other control packets in the network. When a node v_j overhears parameter r_{p_j} and s_{p_j} from its parent p_j, v_j can approximate r_p and s_p with $r_{p_j} - r_j \frac{s_j}{s_{p_j}}$ and s_{p_j} respectively. The derivation is as follows.

Approximation of r_p and s_p: Since information elements generated or forwarded by the children of node p_j are treated in the same manner (without considering where they are from), the expected size of the packet being transmitted by p_j does not depend on whether the packet contains information elements generated or forwarded by v_j. Thus, v_j can simply regard s_{p_j} as s_p, the expected size of the packet transmitted by p_j that does not contain information elements coming from v_j.

Now we derive r_p as follows. Since the amount of payload transmitted by p_j per unit time is $r_{p_j}s_{p_j}$ and the amount of payload transmitted by v_j is r_js_j per unit time, the amount of payload l_p that are transmitted by p_j but are not from v_j per unit time is calculated as: $l_p = r_{p_j}s_{p_j} - r_js_j$. Thus, the expected rate r_p that p_j transmits packets that do not contain information elements from v_j is calculated as: $r_p = l_p/s_{p_j} = r_{p_j} - r_j\frac{s_j}{s_{p_j}}$. $\square$

VI. Performance evaluation

To characterize the impact of packet packing and its joint optimization with data delivery timeliness, we experimentally evaluate the performance of tPack in this section. We first present the experimentation methodology and then the measurement results.

A. Methodology

Testbed. We use the *NetEye* wireless sensor network testbed at Wayne State University [25]. NetEye is deployed in an indoor office as shown in Figure 7. We use a 10×13 grid of TelosB motes in NetEye, where every two closest neighboring motes are separated by 2 feet. Out of the 130 motes in NetEye, we randomly select 120 motes (with each mote being selected with equal probability) to form a random network for our experimentation. Each of these TelosB motes is equipped with a 3dB signal attenuator and a 2.45GHz monopole antenna.

Fig. 7. *NetEye* wireless sensor network testbed

In our measurement study, we set the radio transmission power to be -25dBm (i.e., power level 3 in TinyOS) such that multihop networks can be created. We also use channel 26 of the CC2420 radio to avoid external interference from sources such as the campus WLANs. We use the TinyOS collection-tree-protocol (CTP) [26] as the routing protocol to form the routing structure, and we use the Iowa's Timesync protocol [27] for network wide time synchronization.

Protocols studied. To understand the impact of packet packing and its joint optimization with data delivery timeliness, we comparatively study the following protocols:[3]

- *noPack*: information elements are delivered without being packed in the network.
- *simplePack*: information elements are packed if they happen to be buffered in the same queue, but there is not packing-oriented scheduling.
- *SL*: the *spread latency* algorithm proposed in [11], where the spare time of an information element is evenly spent at each hop from its source to the sink without considering specific network conditions (e.g., network-wide traffic pattern). SL was proposed with total aggregation in mind without considering aggregation constraints such as maximum packet size.
- *CC*: the *common clock* algorithm proposed in [11], where the spare time of an information element is only partly spent at the node where it is generated. Same as SL, CC was proposed with total aggregation in mind.
- *tPack*: the packing- and timeliness-oriented scheduling algorithm that maximizes the local utility at each node, as we discussed in Section V. (We have also evaluated another version of tPack, denoted by *tPack-2hop*, where the forwarding utility U_p considers both the parent node and the parent's parent; we find that tPack-2hop does not bring significant improvement over tPack while introducing higher overhead and complexity, thus our discussion here only focuses on tPack.)

We have implemented, in TinyOS [28], a system library which includes all the above protocols. The implementation takes 40 bytes of RAM (plus the memory required for regular packet buffers) and 4,814 bytes of ROM.

Performance metrics. For each protocol we study, we

[3]We use the terms protocols, algorithms, and decision rules interchangeably in this paper.

evaluate their behavior based on the following metrics:

- *Packing ratio*: number of information elements carried in a packet;
- *Delivery reliability*: percentage of information elements correctly received by the sink;
- *Delivery cost*: number of transmissions required for delivering an information element from its source to the sink;
- *Deadline catching ratio*: out of all the information elements received by the sink, the percentage of them that are received before their deadlines;
- *Latency jitter*: variability of the time taken to deliver information elements from the same source node, measured by the coefficient-of-variation (COV) [29] of information delivery latency.

Traffic pattern. To experiment with different sensornet scenarios, we use both periodic data collection traffic and event detection traffic trace as follows:

- $D3$: each source node periodically generates 50 information elements with an inter-element interval, denoted by Δ_r, uniformly distributed between 500ms and 3s; this is to represent high traffic load scenarios.
- $D6$: same as $D3$ except that Δ_r is uniformly distributed between 500ms and 6s; this is to represent relatively low traffic load scenarios.
- $D9$: same as $D3$ except that Δ_r is uniformly distributed between 500ms and 9s.
- E_{lites}: an event traffic where a source node generates one packet based on the Lites [30] sensornet event traffic trace.

To understand the impact of the timeliness requirement of data delivery, we experiment with different latency requirements. For periodic traffic, we consider maximum allowable latency in delivering information elements that is 1, 3, and 5 times the average element generation period, and we denote them by $L1$, $L3$, and $L5$ respectively; for event traffic, we consider maximum allowable latency that is 2s, 4s, or 6s, and we denote them by $L2'$, $L4'$, and $L6'$ respectively. Out of the 120 motes selected for experimentation, we let the mote closest to a corner of NetEye be the sink node, and the other mote serves as a traffic source if its node ID is even. For convenience, we regard a specific combination of source traffic model and latency requirement a *traffic pattern*. Thus we have 8 traffic patterns in total. To gain statistical insight, we repeat each traffic pattern 20 times. Note that, in each traffic pattern, all the information elements have the same maximum allowable latency. In our implementation, each information element is 16-byte long, and the TelosB motes allow for aggregating up to 7 information elements into a single packet (i.e., $K = 7$).

B. Measurement results

In what follows, we first present the measurement results for periodic traffic patterns $D3$, $D6$, and $D9$, then we discuss the case of event traffic pattern E_{lites}. In most figures of this section, we present the means/medians and their 95% confidence intervals for the corresponding metrics such as

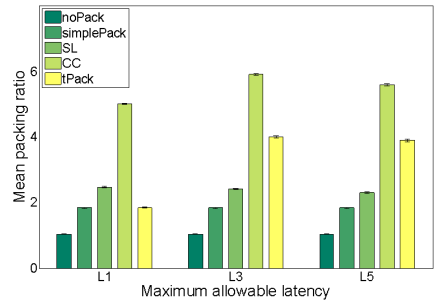


Fig. 8. Packing ratio: $D3$

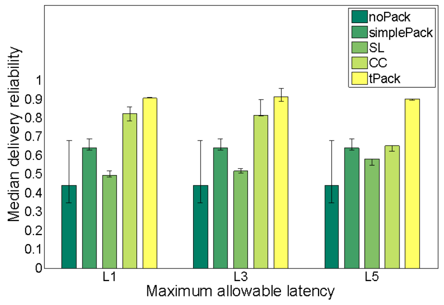


Fig. 9. Delivery reliability: $D3$

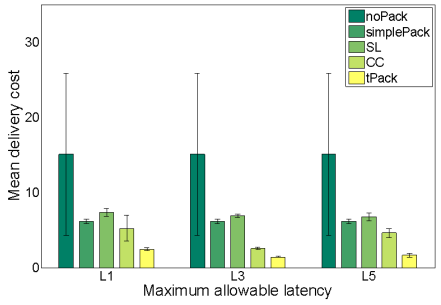


Fig. 10. Delivery cost: $D3$

the packing ratio, delivery reliability, delivery cost, deadline catching ratio, and the latency jitter.[4]

1) Periodic data traffic: For the periodic traffic pattern $D3$, Figures 8-12 show the packing ratio, delivery reliability, delivery cost, deadline catching ratio, and latency jitter in different protocols. tPack tends to enable higher degree of packet packing (i.e., larger packing ratio) than other protocols except the CC protocol. The increased packing in $tPack$ reduces channel contention and thus reduces the probability of packet transmission collision, which improves data delivery reliability. The reduced probability of transmission collision and the increased number of information elements carried per packet in $tPack$ in turn reduces delivery cost, since there are fewer number of packet retransmissions as well as fewer number of packets generated. Note that the low delivery reliability in simplePack is due to intense channel contention.

[4]The distributions for delivery reliability and latency jitter are not symmetric, thus we use medians instead of means to summarize their properties [29].

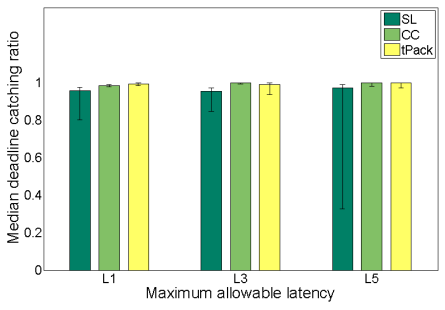


Fig. 11. Deadline catching ratio: $D3$

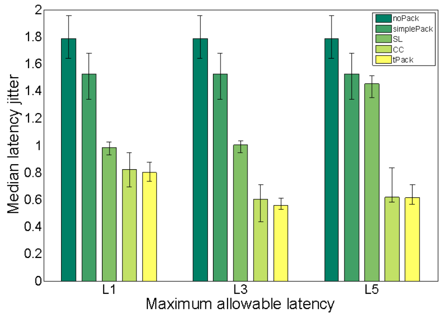


Fig. 12. Latency jitter: $D3$

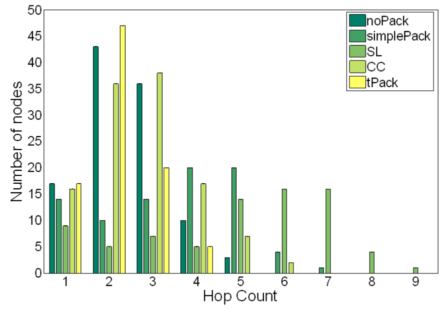


Fig. 13. Histogram of routing hop count: $D3$ with maximum allowable latency $L1$

Exceptions to the above general observation happen in the case of maximum allowable latency $L1$ or when comparing tPack with CC. In the first case, the packing ratio in tPack is lower than that in SL, but tPack still achieves much higher delivery reliability (i.e., by more than 40%) and much lower delivery cost (i.e., by a factor of more than 3). This is because the packing ratio in SL is too high such that, in the presence of high wireless channel contention due to the high traffic load of $D3$ and the stringent real-time requirement of $L1$, the resulting long packet length leads to higher packet error rate and lower packet delivery reliability (as shown in Figure 9). The routing protocol CTP adapts to the higher packet error rate in SL, and this leads to longer routes and larger routing hops in SL. This can be seen from Figure 13 which shows the histogram of routing hop counts in different protocols. The maximum hop count in tPack is 4, whereas the hop count can be up to 9 in SL. Together, the higher packet error rate

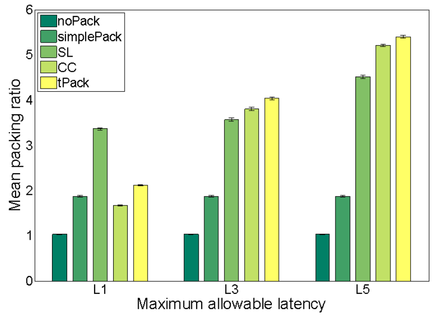


Fig. 14. Packing ratio: $D6$

and the longer routes in SL lead to larger delivery cost in SL as compared with tPack. Similar arguments apply to the case when comparing tPack with CC. From these data on the benefits of tPack in comparison with SL and CC, we can see the importance of adapting to network conditions and data aggregation constraints in in-network processing. Note that similar arguments also explain the phenomenon where SL has higher packing ratio than simplePack but lower delivery reliability and higher delivery cost under all latency settings of $D3$ traffic.

Figure 9 also shows that tPack improves data delivery reliability even when the allowable latency in data delivery is small (e..g, in the case of $L1$) where the inherent probability for packets to be packed tends to be small. Therefore, tPack can be used for real-time applications where high data delivery reliability is desirable. Figure 8 shows that the packing ratio in tPack is close to 4 except for the case of $L1$ where 1) too much packing is undesirable as discussed earlier and 2) the packing probability is significantly reduced by the limited probability for a node to wait due to stringent timeliness requirement. Our offline analysis shows that the optimal packing ratio is $\sim$5 for the traffic patterns $D3$-$L3$ and $D3$-$L5$; thus tPack achieves a packing ratio very close to the optimal, which corroborates our analytical result in Theorem 8.

Figure 11 shows the deadline catching ratio in deadline-aware data aggregation schemes tPack, SL, and CC. Though the deadline catching ratio of all the three protocols are close to 1, the catching ratio of tPack is the highest and is greater than 0.99 in all cases. The slightly higher deadline catching ratio in tPack is a result of its online adaptation of packet holding time at each hop according to in-situ channel and traffic conditions along the path. As a result of the properly controlled packet packing, the reduced channel contention and improved packet delivery reliability in tPack also help enable lower performance variability. For instance, Figure 12 shows the latency jitter in different protocols, and we see that the jitter tends to be the lowest in tPack, especially when the real-time requirement is stringent (e.g., in $L1$ and $L3$). These properties are desirable in cyber-physical-system (CPS) sensornets where real-time sensing and control require predictable data delivery performance (e.g., in terms of low latency jitter), especially in the presence of potentially unpredictable, transient perturbations.

Figures 14-18 and Figures 19-23 show the measurement results for periodic traffic patterns $D6$ and $D9$ respectively. We

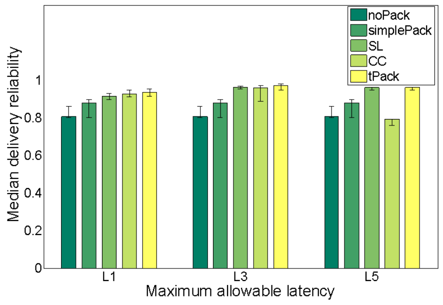


Fig. 15. Delivery reliability: $D6$

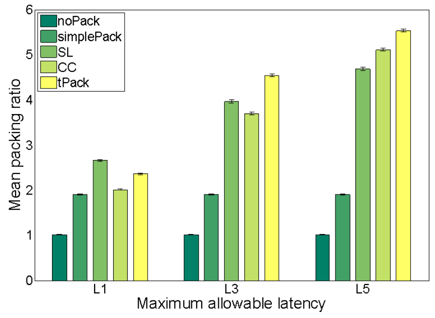


Fig. 19. Packing ratio: $D9$

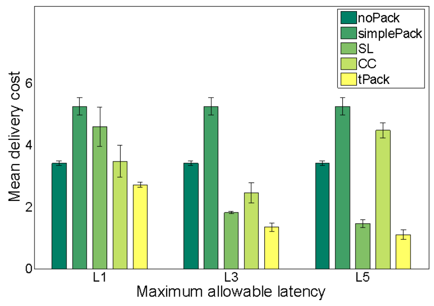


Fig. 16. Delivery cost: $D6$

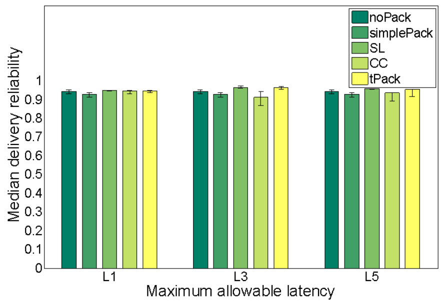


Fig. 20. Delivery reliability: $D9$

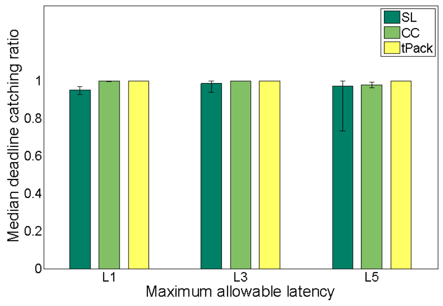


Fig. 17. Deadline catching ratio: $D6$

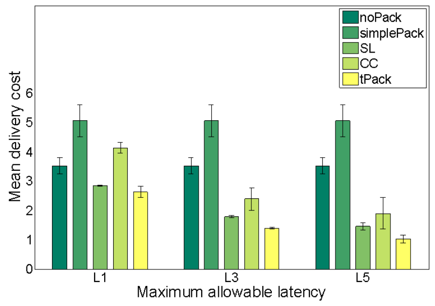


Fig. 21. Delivery cost: $D9$

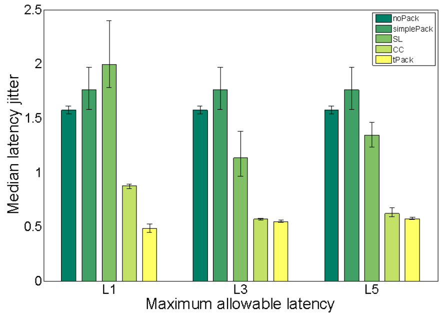


Fig. 18. Latency jitter: $D6$

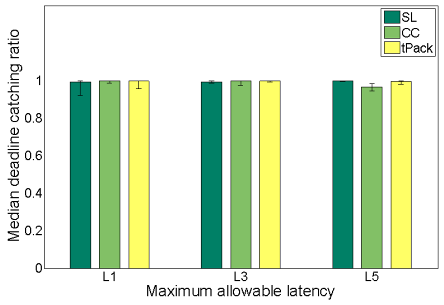


Fig. 22. Deadline catching ratio: $D9$

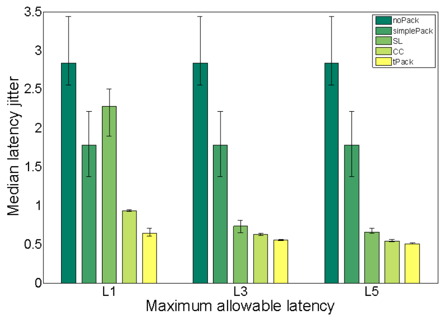


Fig. 23. Latency jitter: $D9$

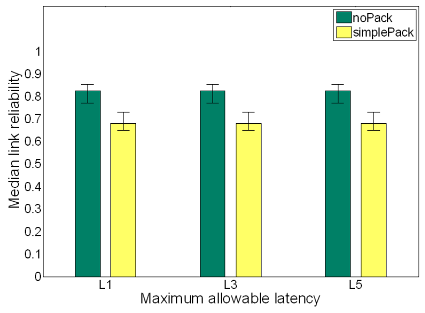


Fig. 24. Link reliability: $D6$

see that, in terms of relative protocol performance, the overall trends in $D6$ and $D9$ are similar to those in $D3$. For instance, with stringent real-time requirement in $L1$, SL achieves a lower delivery reliability and a higher delivery cost than tPack even though the packing ratio tends to be higher in SL. Due to the reduced traffic load and thus the reduced wireless channel contention and collision, however, the delivery reliability of noPack, simplePack, and SL is also relatively high compared with their delivery reliability in $D3$.

Note that, in [11], CC is shown to have a much higher competitive ratio than SL through theoretical analysis. From our measurement study, however, we see that the performance of CC is not always better than SL. For instance, CC has a lower delivery reliability and a higher delivery cost than SL in $D6 - L5$. This seemingly discrepancy is due to the fact that the theoretical analysis of [11] does not consider the limit of data aggregation capacity, nor does it consider wireless link unreliability and interference in scheduling.

Surprisingly, Figures 14-16 show that, for the traffic pattern $D6$, simplePack introduces higher delivery cost than noPack does even though the packing ratio and the end-to-end delivery reliability are higher in simplePack. One reason for this is that, partially due to the increased packet length in simplePack, the link reliability in simplePack is lower than that in noPack as shown in Figure 24.[5] The routing protocol CTP adapts to the lower link reliability in simplePack and introduces longer routing hop length, which can be seen from Figure 25

[5]The reason why simplePack still has higher end-to-end information element delivery reliability despite its lower link reliability is because each packet delivered in simplePack carries more information elements due to the higher packing ratio.

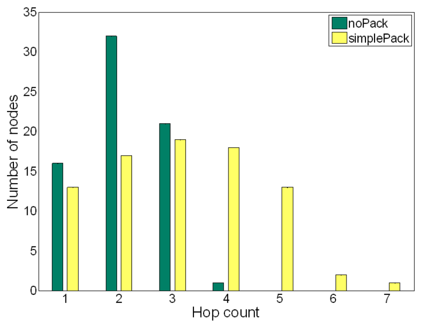


Fig. 25. Histogram of routing hop count: $D6$ with maximum allowable latency $L1$

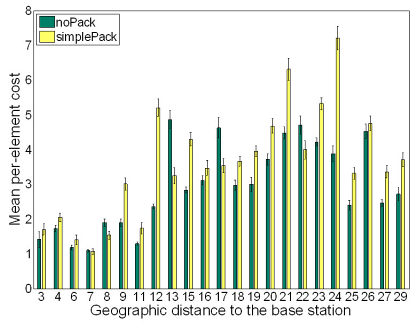


Fig. 26. Per-element delivery cost vs. geographic distance: $D6$ with maximum allowable latency $L1$

which shows the histogram of routing hop counts for noPack and simplePack in traffic pattern $D6$-$L1$. Together, the lower link reliability and the longer routes in simplePack introduce larger information delivery cost when compared with noPack in $D6$. This observation is also corroborated by the detailed analysis of the cost (e.g., mean number of transmissions) taken to deliver an information element. For instance, Figure 26 shows the mean cost of delivering an information element from a node at different geographic distances (in terms of the number of grid hops) from the base station for the traffic pattern $D6$-$L1$. (Similar phenomena are observed for other traffic patterns.) We see that, for most of the cases, the per-element delivery cost is higher in simplePack. Note that similar arguments explain why simplePack has higher delivery cost than noPack in traffic pattern $D9$ and why SL also has higher delivery cost than noPack in several cases (e.g., for traffic pattern $D6$-$L1$). In view with the consistently better performance in tPack, these observations demonstrate again the importance of considering network conditions and data aggregation constraints in in-network processing.

2) Event traffic: Figures 27-31 show the measurement results for event traffic pattern E_{lites}. The overall trend on the relative protocol performance is similar to that in the periodic traffic patterns $D3$, $D6$, and $D9$. Even though the delivery reliability tends to be high for all protocols, tPack still achieves lower delivery cost and latency jitter, ???as well as 100% deadline catching ratio.

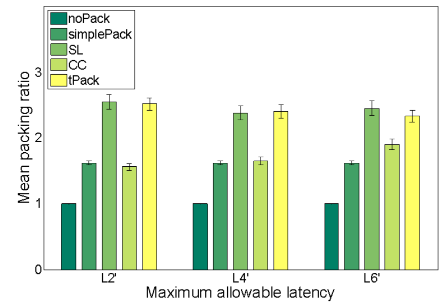


Fig. 27. Packing ratio: E_{lites}

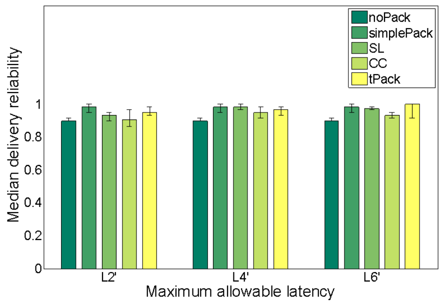


Fig. 28. Delivery reliability: E_{lites}

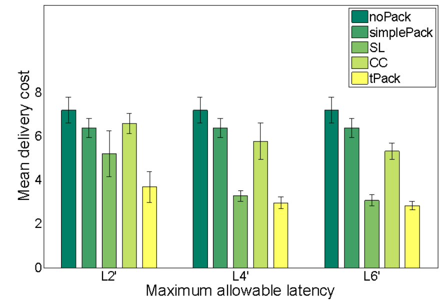


Fig. 29. Delivery cost: E_{lites}

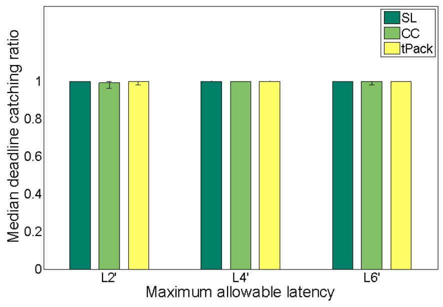


Fig. 30. Deadline catching ratio: E_{lites}

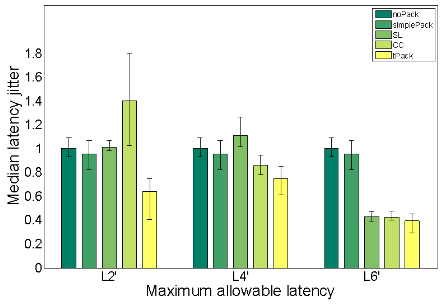


Fig. 31. Latency jitter: E_{lites}

VII. Related work

In-network processing (INP) has been well studied in sensornets, and many INP methods have been proposed for query processing [1], [2], [31], [3], [4], [32], [33], [34] and general data collection [5], [6], [7], [8], [9], [10]. When controlling spatial and temporal data flow to enhance INP, however, these methods did not consider application requirements on the timeliness of data delivery. As a first step toward understanding the interaction between INP and application QoS requirements, our study has shown the benefits as well as the challenges of jointly optimizing INP and QoS from the perspective of packet packing. As sensornets are increasingly being deployed for mission-critical tasks, it becomes important to address the impact of QoS requirements on general INP methods other than packet packing, which opens interesting avenues for further research.

As a special INP method, packet packing has also been studied for sensornets as well as general wireless and wired networks, where mechanisms have been proposed to adjust the degree of packet packing according to network congestion level [13], [35], to address MAC/link issues related to packet packing [36], [14], [37], to enable IP level packet packing [38], and to pack periodic data frames in automotive applications [39]. These works have focused on issues in local, one-hop networks without considering requirements on maximum end-to-end packet delivery latency in multi-hop networks. With the exception of [39], these works did not focus on scheduling packet transmissions to improve the degree of packet packing, and they have not studied the impact of finite packet size either. Saket et al. [39] studied packet packing in single-hop controller-area-networks (CAN) with finite packet size. Our work addresses the open questions on the complexity and protocol design issues for jointly optimizing packet packing and data delivery timeliness in multi-hop wireless sensornets.

Most closely related to our work are [11], [40], [41] where the authors studied the issue of optimizing INP under the constraint of end-to-end data delivery latency. But these studies did not consider aggregation constraints and instead assumed *total aggregation* where any arbitrary number of information elements can be aggregated into one single packet. These studies did not evaluate the impact of joint optimization on data delivery performance either. Our work focuses on settings where packet size is finite, and we show that aggregation constraints (in particular, maximum packet size and re-aggregation tolerance) significantly affect the problem complexity and protocol design. Using a high-fidelity sensornet testbed, we also systematically examine the impact of joint optimization on packet delivery performance in multi-hop wireless networks. By showing that tPack performs better than the algorithm SL and CC [11], [41], our testbed based measurement results also demonstrate the benefits of considering realistic aggregation constraints in the joint optimization.

Solis et al. [42] also considered the impact that the timing of packet transmission has on data aggregation, and the problem of minimizing the sum of data transmission cost and delay cost has been considered in [12] and [43]. These studies also assumed total aggregation, and they did not consider hard real-time requirements on maximum end-to-end data delivery latency. Ye et al. [44] considered the local optimal stopping rule for data sampling and transmission in distributed data aggregation. It did not consider hard real-time requirement either, and it did not study network-wide coordination and the limit of data aggregation. Yu et al. [45] studied the latency-energy tradeoff in sensornet data gathering by adapting radio transmission rate; it did not study the issue of scheduling data transmission to improve the degree of data aggregation.

VIII. Concluding remarks

Through both theoretical and experimental analysis, we examine the complexity and impact of jointly optimizing packet packing and the timeliness of data delivery. We find that aggregation constraints (in particular, maximum packet size and re-aggregation tolerance) affect the problem complexity more than network and traffic properties do, which suggest the importance of considering aggregation constraints in the joint optimization. We identify conditions for the joint optimization to be strong NP-hard and conditions for it to be solvable in polynomial time. For cases when it is polynomial-time solvable, we solve the problem by transforming it to the maximum weighted matching problem in interval graphs; for cases when it is strong NP-hard, we prove that there is no polynomial-time approximation scheme (PTAS) for the problem. We also develop a local, distributed online protocol tPack for maximizing the local utility of each node, and we prove the competitiveness of the protocol with respect to optimal solutions. Our testbed-based measurement study also corroborates the importance of QoS- and aggregation-constraint aware optimization of packet packing.

While this paper has extensively studied the complexity, algorithm design, and impact of jointly optimizing packet packing and data delivery timeliness, there are still a rich set of open problems. Even though we have analyzed the competitiveness of tPack for non-trivial scenarios and this has given us insight into the behavior of tPack, it remains an open question on how to characterize in a closed form the competitiveness of tPack and non-oblivious online algorithms in broader contexts. The analytical and algorithmic design mechanisms developed for packet packing may well be extensible to address other in-network processing methods such as data fusion, and a detailed study of this will help us better understand the structure of the joint optimization problem and will be interesting future work to pursue. We have focused on the scheduling aspect of the joint optimization, and we are able to use mathematical tools such as interval graphs to model the problem; on the other hand, how to mathematically model and analyze the impact of the joint optimization on spatial data flow is still an open question and is beyond the scope of most existing network flow theory, thus it will be interesting to explore new approaches to modeling and solving the joint optimization problem.

Acknowledgment

We thank Alper Murat, Nathan Fisher, Yi Feng, Dan Zhang, Gerd Finke for interesting discussions, and Xi Ju for helping with experimentation. This work is supported in part by the NSF GENI project #1633.

References

[1] S. Madden, M. Franklin, and J. Hellerstein, “TinyDB: An acquisitional query processing system for sensor systems,” in *ACM Transactions on Database Systems*, 2004.

[2] Y. Yao and J. Gehrke, “The cougar approach to in-network query processing in sensor networks,” in *ACM SIGMOD*, 2002.

[3] S. Nath, P. Gibbons, S. Seshan, and Z. Anderson, “Synopsis diffusion for robust aggregation in sensor networks,” in *ACM SenSys*, 2004.

[4] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong, “Model-driven data acquisition in sensor networks,” in *VLDB*, 2004.

[5] K.-W. Fan, S. Liu, and P. Sinha, “Scalable data aggregation for dynamic events in sensor networks,” in *ACM SenSys*, 2006.

[6] Q. Fang, F. Zhao, and L. Guibas, “Lightweight sensing and communication protocols for target enumeration and aggregation,” in *ACM MobiHoc*, 2003.

[7] R. Kumar, M. Wolenetz, B. Agarwalla, J. Shin, P. Hutto, A. Paul, and U. Ramachandran, “DFuse: A framework for distributed data fusion,” in *ACM SenSys*, 2003.

[8] J. Liu, M. Adler, D. Towsley, and C. Zhang, “On optimal communication cost for gathering correlated data through wireless sensor networks,” in *ACM MobiCom*, 2006.

[9] S. Pattem, B. Krishnamachari, and R. Govindan, “The impact of spatial correlation on routing with compression in wireless sensor networks,” in *ACM/IEEE IPSN*, 2004.

[10] S. Yoon and C. Shahabi, “The clustered aggregation (CAG) technique leveraging spatial and temporal correlation in wireless sensor networks,” *ACM Transactions on Sensor Networks*, vol. 3, no. 1, 2007.

[11] L. Becchetti, P. Korteweg, A. Marchetti-Spaccamela, M. Skuttella, L. Stougie, and A. Vitaletti, “Latency constrained aggregation in sensor networks,” in *European Symposium on Algorithms (ESA)*, 2006.

[12] Y. A. Oswald, S. Schmid, and R. Wattenhofer, “Tight bounds for delay-sensitive aggregation,” in *ACM PODC*, 2008.

[13] T. He, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “AIDA: Adaptive application independent data aggregation in wireless sensor networks,” *ACM Transaction on Embedded Computing System*, vol. May, 2004.

[14] K. Lu, D. Wu, Y. Qian, Y. Fang, and R. C. Qiu, “Performance of an aggregation-based MAC protocol for high-data-rate ultrawideband ad hoc networks,” *IEEE Transactions on Vehicular Technology*, vol. 56, no. 1, pp. 312–321, 2007.

[15] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, and H. Zhang et al. (17 authors), “A line in the sand: A wireless sensor network for target detection, classification, and tracking,” *Computer Networks (Elsevier)*, vol. 46, no. 5, 2004.

[16] “IETF 6LowPAN working group,” http://www.ietf.org/html.charters/6lowpan-charter.html.

[17] IETF, “Routing over low power and lossy networks (ROLL) working group,” http://www.ietf.org/html.charters/roll-charter.html.

[18] G. Finke, V. Jost, M. Queyranne, and A. Sebo, “Batch processing with interval graph compatibilities between tasks,” *Discrete Applied Mathematics (Elsevier)*, vol. 156, 2008.

[19] A. Arora, R. Ramnath, E. Ertin, P. Sinha, S. Bapat, V. Naik, V. Kulathumani, H. Zhang, H. Cao, M. Sridhara, S. Kumar, N. Seddon, C. Anderson, T. Herman, N. Trivedi, C. Zhang, M. Gouda, Y. R. Choi, M. Nesterenko, R. Shah, S. Kulkarni, M. Aramugam, L. Wang, D. Culler, P. Dutta, C. Sharp, G. Tolle, M. Grimmer, B. Ferriera, and K. Parker, "Exscal: Elements of an extrem scale wireless sensor network," in *IEEE RTCSA*, 2005.

[20] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of* NP*-Completeness*. Freeman and Company, 1979.

[21] D. S. Hochbaum, *Approximation Algorithms for* NP*-hard Problems*. PWS Publishing Company, 1997.

[22] P. Wang, J. Zheng, and C. Li, "Data aggregation using distributed lossy source coding in wireless sensor networks," in *IEEE GLOBECOM*, 2007.

[23] H. Gabow, "An efficient implementation of edmonds' algorithm for maximum matchings on graphs," *Journal of ACM*, vol. 23, pp. 221–234, 1975.

[24] Q. Xiang, J. Xu, X. Liu, H. Zhang, and L. J. Rittle, "When in-network processing meets time: Complexity and effects of joint optimization in wireless sensor networks," Wayne State University (http://www.cs.wayne.edu/∼hzhang/group/TR/DNC-TR-09-01.pdf), Tech. Rep., 2009.

[25] "NetEye testbed," http://neteye.cs.wayne.edu/neteye/home.php.

[26] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *ACM SenSys*, 2009.

[27] "Iowa's timesync component," http://tinyos.cvs.sourceforge.net/viewvc/tinyos/tinyos-2.x-contrib/iowa/T2.tsync/.

[28] "TinyOS," http://www.tinyos.net/.

[29] R. Jain, *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, Inc., 1991.

[30] "An event traffic trace for sensor networks," http://www.cs.wayne.edu/∼hzhang/group/publications/Lites-trace.txt.

[31] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: a tiny aggregation service for ad-hoc sensor networks," in *OSDI*, 2002.

[32] A. Deshpande, C. Guestrin, W. Hong, and S. Madden, "Exploiting correlated attributes in acquisitional query processing," Intel Research - Berkeley, Tech. Rep., 2004.

[33] X. Li, Y. J. Kim, R. Govindan, and W. Hong, "Multi-dimensional range queries in sensor networks," in *ACM SenSys*, 2003.

[34] J. Hellerstein, W. Hong, S. Madden, and K. Stanek, "Beyond average: Toward sophisticated sensing with queries," in *IPSN*, 2003.

[35] A. Jain, M. Gruteser, M. Neufeld, and D. Grunwald, "Benefits of packet aggregation in ad-hoc wireless network," University of Colorado at Boulder, Tech. Rep. CU-CS-960-03, 2003.

[36] T. Li, Q. Ni, D. Malone, D. Leith, Y. Xiao, and T. Turletti, "Aggregation with fragment retransmission for very high-speed WLANs," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 591–604, 2009.

[37] M. Li, H. Zhu, Y. Xiao, I. Chlamtac, and B. Prabhakaran, "Adaptive frame concatenation mechanisms for qos in multi-rate wireless ad hoc networks," in *IEEE INFOCOM*, 2008.

[38] D. Kliazovich and F. Granelli, "Packet concatenation at the ip level for performance enhancement in wireless local area networks," *Wireless Networks*, vol. 14, pp. 519–529, 2008.

[39] R. Saket and N. Navet, "Frame packing algorithms for automotive applications," *Journal of Embedded Computing*, vol. 2, pp. 93–102, 2006.

[40] T. Nonner and A. Souza, "Latency constrained data aggregation in chain networks admits a PTAS," in *AAIM*, 2009.

[41] P. Korteweg, A. Marchetti-Spaccamela, L. Stougie, and A. Vitaletti, "Data aggregation in sensor networks: Balancing communication and delay costs," *Theoretical Computer Science*, vol. 410, no. 14, 2009.

[42] I. Solis and K. Obraczka, "The impact of timing in data aggregation for sensor networks," in *IEEE ICC*, 2004.

[43] A. R. Karlin, C. Kenyon, and D. Randall, "Dynamic TCP acknowledgement and other stories about $\frac{e}{e-1}$," in *ACM STOC*, 2001.

[44] Z. Ye, A. A. Abouzeid, and J. Ai, "Optimal policies for distributed data aggregation in wireless sensor networks," in *IEEE INFOCOM*, 2007.

[45] Y. Yu, V. Prasanna, and B. Krishnamachari, "Energy minimization for real-time data gathering in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 11, 2006.

[46] C. di Complessita and R. Rizzi, "Np-complete problem: Partition into triangles," *Technical Report*, 2004.

[47] "Total interval numbers of complete r-partite graphs," *Discrete Applied Mathematics*, vol. 122, no. 1-3, pp. 83 – 92, 2002.

Appendix

Appendix 1: proofs of Claims 1, 2, 3, and 4

Proof of Claim 1: It is easy to see that the information elements generated by $v_i, i = 1, \ldots, n$, and v, cannot be packed with any other information elements. Therefore, the total number of transmission for these elements is $C_{t0}^1 = n(D+1)+1$.

Since the ETX of each link from v_j to v, $j = 1, \ldots, n$ is D and $D \gg 1$, and each sensor only generates one piece of information element, in an optimal packing scheme, every information element generated by node $v_{t_j}^j$, $t_j = 1, , 2k_j + 1$, will leave its source immediately it is generated and then seek the opportunity to pack with other information elements before it is forwarded from v_j to v. Due to our definition on lifetimes for every $2k_j + 1$ elements generated by nodes $v_{t_j}^j, t_j = 1, \ldots, 2k_j+1$, only at most two consecutive information elements in this $2k_j+1$ sequence can be packed together at node v_j. For any two consecutive information elements that are packed together, the first element, which is generated by $v_{t_j}^j$ leaves node v_j at time $d_{t_j}^j - (t_2+t_3)$, and the second element, which is generated by $v_{t_j+1}^j$ leaves node v_j at time $r_{t_j+1}^j + t_1$. Thus in an optimal packing scheme, for all $2k_j + 1$ incoming elements, node v_j will pack them into at least $k_j + 1$ packets, k_j of which contain two element. In each $2k_j + 1$ sequence, either information element ax_o^j arrives at and leaves node v_j at time $j + t_1$ alone, or information element $ax_{k_j}^j$ arrives at and leaves node v_j at time $3(m+1)(n+1) + j + t_1$ alone. Thus, the total number of transmission for these elements is $C_{t0}^2 = \sum_{j=1}^{n}(2k_j+1) + \sum_{j=1}^{n}[(k_j+1)(D+1)]$.

Besides, we have $2n$ more information elements z_0^j and z_{m+1}^j, $j = 1, \ldots, n$, left. Due to the definition of lifetimes for these information elements, all of them need to leave their sources as soon as they are generated, and none of them can be packed with a packet containing two information elements we packed in the last paragraph. In an optimal packing scheme, for a fixed j, either z_0^j is packed with ax_0^j at node v_j, i.e., ax_0^j arrives at and leaves node v_j at time $j+t_1$, or z_{m+1}^j is packed with $ax_{k_j}^j$ at node v_j, i.e., $ax_{k_j}^j$ arrives at and leaves node v_j at time $3(m+1)(n+1)+j+t_1$, which is shown in Figure 32. Thus, the total number of transmission for these elements is

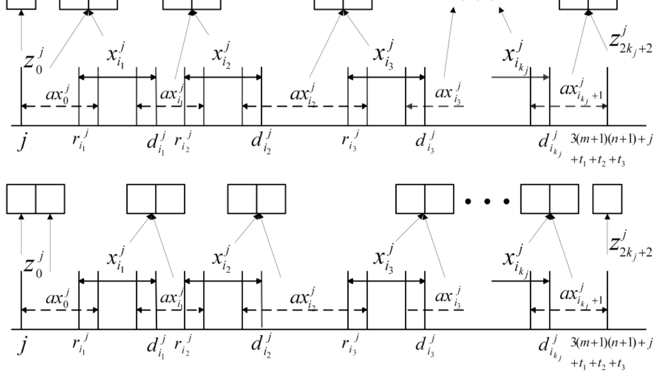


Fig. 32. Example of optimal packing when $K \geq 3$

$C^3_{t0} = 2n + n(D+1)$. Under this packing scheme, no packet will contain more than 2 elements, which also satisfies the packing size constraint. Thus, the minimal total number of transmissions in this tree is $C^1_{t0}+C^2_{t0}+C^3_{t0} = n(D+1)+1+\sum_{j=1}^{n}(2k_j+1)+\sum_{j=1}^{n}[(k_j+1)(D+1)]+2n+n(D+1) = C_{t0}$.

□

Proof of Claim 2: Correctness of this claim can be easily verified by the definition of the information elements of those leaf nodes.

□

Proof of Claim 3: Since in an optimal packing scheme, either element z^j_0 is packed with element ax^j_0, or element $z^j_{2k_j+2}$ is packed with element $ax^j_{k_j}$. If z^j_0 is packed with ax^j_0, ax^j_0 leaves v_j as soon as it arrives at v_j, when z^j_0 arrives at v_j, i.e., each element $x^j_{i^j_t}$ leaves from v_j for v at time $d^j_{i^j_t} - (t_2+t_3)$, packed with element $ax^j_{i^j_t}$, $t = 1, \dots, k_j$. If $z^j_{2k_j+2}$ is packed with $ax^j_{k_j}$, $ax^j_{k_j}$ leaves v_j at time $3(m+1)(n+1)+j+t_1$, which equals to $d^j_{a_{k_j0}} - (t_2+t_3)$, when $z^j_{2k_j+2}$ arrives at v_j, i.e., each element $x^j_{i^j_t}$ leaves from v_j for v at time $r^j_{i^j_t} + t_1$, packed with element $ax^j_{i^j_t-1}$, $t = 1, \dots, k_j$.

□

Proof of Claim 4: 1) Given a satisfying assignment for the SAT problem, an optimal packing scheme of the corresponding packet packing problem can be derived as follows: If in the assignment of SAT problem, variable X_j is set true, then all information elements x^j_i are forwarded from their sources to node v_j at time r^j_i, and are forwarded from node v_j to node v at time $r^j_i+t_1$. If X_j is set false, then all information elements x^j_i are forwarded from their sources to node v_j at time r^j_i, and are forwarded from node v_j to node v at $d^j_i - t_2 - t_3$. Similarly with the information elements generated by children nodes of node v_j, every element generated by node v^c_i, $i = 1, \dots, m$, cannot get packed at its source since v^c_i is a leaf node. As a result, each element z_i is forward by its source and arrives at node v at time $(3i+1)(n+1)+t_1+t_2-t_4+t_4 = (3i+1)(n+1)+t_1+t_2$. Then the spare period for information element z_i to wait at node v is $[(3i+1)(n+1)+t_1+t_2, (3i+2)(n+1)+t_1+t_2]$. If clause C_i is satisfied by setting X_j to be true, then information element x^j_i arrives at node v at $(3i+1)(n+1)+t_1+t_2+j \in [(3i+1)(n+1)+t_1+t_2, (3i+2)(n+1)+t_1+t_2]$, which implies z_i can be packed with any packet containing information element x^j_i. Similarly, if clause C_i is satisfied by setting X_j to be false, then information element x^j_i arrives at node v at $(3i+1)(n+1)+t_1+t_2+j \in [(3i+1)(n+1)+t_1+t_2, (3i+2)(n+1)+t_1+t_2]$, which implies z_i can be packed with any packet containing information element x^j_i. Figure 33 gives an example on how to get the optimal packing scheme from an assignment of SAT instance.

Under this scheme, no packet will contain more than 3 elements, which also satisfies the packing size constraint. Every element z_i, $i = 1, \dots, m$, can be packed at node v with a packet containing message x^j_i if clause C_i is satisfied due to variable X_j. Therefore, the additional number of transmission

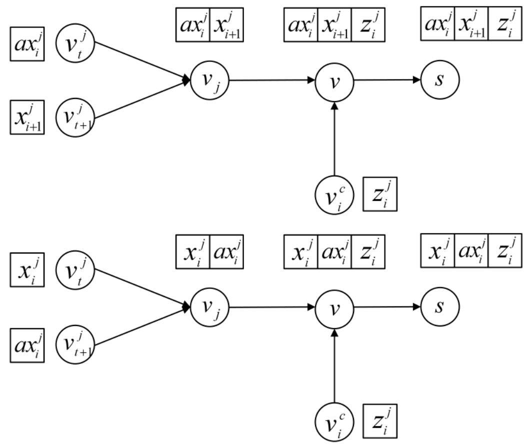


Fig. 33. Example of deriving the optimal packing scheme from the SAT assignment when $K \geq 3$

to send each element z_i to node s is m. As a result, the total number of transmission for this tree is $C_{t0} + m = C_{t1}$.

2) If we may find that the optimal packing scheme has a total number of transmission C_{t1}, which implies that every element z_i joins a packet consisting of x^j_i for some j value. If x^j_i leaves from node v_j at time $r^j_i + t_1$, and z_i joins the packet that contains x^j_i at node v, this can only happen when X_j is unnegated in clause C_i because $(3i+1)(n+1)+t_1+t_2+j \in [(3i+1)(n+1)+t_1+t_2, (3i+2)(n+1)+t_1+t_2]$ and $3i(n+1)+j \notin [(3i+1)(n+1)+t_1+t_2, (3i+2)(n+1)+t_1+t_2]$. Thus we set X_j to be true. If x^j_i leaves from node v at time $d^j_i - (t_2+t_3)$, and z_i joins the packet that contains x^j_i at node v, this can only happen when X_j is negated in clause C_i because $(3i+1)(n+1)+t_1+t_2+j \in [(3i+1)(n+1)+t_1+t_2, (3i+2)(n+1)+t_1+t_2]$ and $(3i+2)(n+1)+j+t_1+t_2 \notin [(3i+1)(n+1)+t_1+t_2, (3i+2)(n+1)+t_1+t_2]$. Thus we set X_j to be false. By this way, if we have an optimal solution to this instance of packet packing problem, we can have a satisfying assignment of the original SAT problem. Note that due to Claim 3, the following case cannot happen: element z_i gets packed with x^j_i by letting x^j_i leaves node v_j at time $r^j_i + t_1$, and in the meantime, that element z_k gets packed with x^j_k by letting x^j_k leaves node v_j at time $d^j_k - (t_2+t_3)$.

□

Appendix 2: complexity of problem $\mathbb{P}_0$ in chain networks

Claim 9: When $K \geq 3$, problem $\mathbb{P}_0$ is strong NP-hard in chain networks (when not all elements are generated at the same time).

Proof: We first define a chain composed by m nodes $v^c_1, \dots, v^c_m$, and m elements generated by nodes $v^c_1, \dots, v^c_m$: $z_i : [r_i, d_i] = [(3i+1)(n+1), (3i+2)(n+1)+T^c_i]$, $i = 1, \dots, m$, where T^c_i is the transmission time from node v^c_i to the base station.

In this section, we introduce a new concept called spare time interval, which is the interval from the generation time of an element to the latest possible time that it has to leave its source.

For each boolean variable X_j, we define a sub-chain of $2k_j + n + 1$ nodes, where k_j is the number of times that X_j occurs in all n clauses. For the first $2k_j + 1$ nodes, we first define a sequence of $2k_j + 1$ elements, and then assign them one for each node. If X_j appears unnegated in clause C_i, define an element x_i^j with a spare time interval $[r_i, d_i] = [0, n+1]$. If X_j appears negated in clause C_i, define an element x_i^j with an initial lifetime $[r_i, d_i] = [-(n+1), 0]$. Let $i_1^j < \ldots < i_{k_j}^j$ denote the indices of the clauses in which variable X_j occurs. Put element $x_{i_t^j}^j, t = 1, \ldots, k_j$ in the $(3 + 2(k_j - t))$th node of this sub-chain, and define the transmission time from the source node of element x_i^j to the $(2k_j+2)$th node of this sub-chain to be $(3i+1)(n+1)+j$. For every two messages $x_{i_t^j}^j$ and $x_{i_{t+1}^j}^j, t = 1, \ldots, k_j - 1$, define an information element $ax_{i_t^j}^j : [r_{a_t}^j, d_{a_t}^j] = [d_{i_t^j}^j - \frac{\Delta T_{t,t+1}}{2} + T_j, r_{i_{t+1}^j}^j + \frac{\Delta T_{t,t+1}}{2}]$, and define the element $ax_{i_t^j}^j$'s source to be the node between the source node of element $x_{i_t^j}^j$ and $x_{i_{t+1}^j}^j$. Define the transmission time from this node to its parent node, and from its child to itself, are both $\frac{\Delta T_{t,t+1}}{2}$. Here $\Delta T_{t,t+1}$ is the transmission time from the source node of element $x_{i_{t+1}^j}^j$ to $x_{i_t^j}^j$. And we then define element $ax_0^j : [r_{a0}^j, d_{a0}^j] = [r_{x_1}^j + \frac{(3i_1^j+1)(n+1)+j}{2} - 1, r_{x_1}^j + \frac{(3i_1^j+1)(n+1)+j}{2}]$ with source node to be the $(2k_j+2)$th node of this sub-chain with transmission time to next hop $\frac{(3i_1^j+1)(n+1)+j}{2}$ and $ax_{k_j}^j : [r_{a_{k_j}}^j, d_{a_{k_j}}^j] = [d_{i_{k_j}^j}^j - 1, d_{i_{k_j}^j}^j]$ with source node to be the second node of this sub-chain with a transmission time of 1 to next hop. Define the ETX for each link to be k, where k is a positive real number.

After defining element for the first $2k_j + 1$ elements, we define n elements for the remaining n nodes. from the $(2k_j + 2)$th node of this sub chain on, define element $z_i : [(3i + 1)(n+1), (3i+2)(n+1)]$, and the transmission time between each two consecutive nodes of these n nodes to be zero, and the ETX to be k.

After defining the sub-chain for each X_j, connect all these sub-chains one by one, and define the transmission time from the head of a chain to the end of next chain is B, which is a large positive number. At last, we add a large positive number Q to the endpoints of all spare time intervals so that the endpoints of all spare time intervals are all positive. The whole transformation process is still of a polynomial time $O(nm)$.

Then for this chain network, we can easily take the steps we used in proving Theore 1 to prove the correctness of this claim. ■

Similarly, we can define a polynomial transformation from any SAT instance to an instance of problem $\mathbb{P}_0$ when $K = 2$ and re-aggregation is allowed. Then, the following claim holds:

Claim 10: When $K = 2$ and re-aggregation is allowed, problem $\mathbb{P}_0$ is strong NP-hard in chain networks (when not all elements are generated at the same time).

Appendix 3: complexity of problem $\mathbb{P}_0$ in trees when all the information elements are generated at the same time

In this section, we will discuss the complexity of solving the following problem

Problem $\mathbb{P}_0'$: same as $\mathbb{P}_0$ except that 1) every element x in X is generated at the same time. 2) T is a tree with branches.

From the above problem definition, we can find that $\mathbb{P}_0'$ is a special case of $\mathbb{P}_0$. Similar to the approach we used in Section IV, we prove the NP-hardness of $\mathbb{P}_0'$ by reducing SAT problem to it, and we separately analyze the case when $K \geq 3$ and the case when $K = 2$.

Theorem 9: When $K \geq 3$, problem $\mathbb{P}_0'$ is strong NP-hard.

Proof: To prove this theorem, we first show that there is a polynomial transformation f' from the SAT problem to $\mathbb{P}_0'$. Then we prove that an instance Π of SAT is satisfiable if and only if the optimal solution of $\Pi' = f'(\Pi)$ has certain minimum number of transmissions.

The transformation f' is similar as the transformation f we used when proving Theorem 1. Given a instance Π of the SAT problem which has n Boolean variables $X_1, \ldots, X_n$ and m clauses $C_1, \ldots, C_m$, we first define a tree of the same structure in Figure 4, which takes $O(mn)$ time and the whole tree is shown in Figure 34.

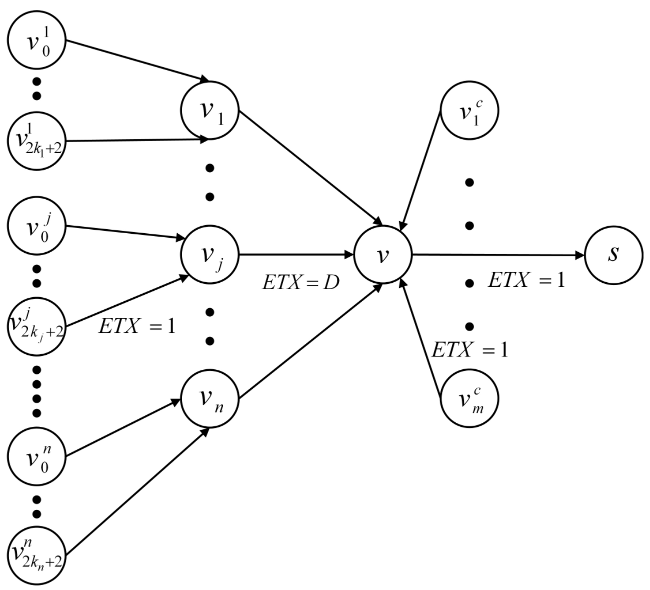


Fig. 34. Reduction from SAT to $\mathbb{P}_0'$ when $K \geq 3$

Notice that in this tree definition, we have not define the transmission time on each edge and we will define them during the process we define the information elements and their lifetimes. To begin with, We still define the transmission time from node v_j to v to be t_2, and the transmission time from v to s to be t_3.

After that, for each subtree rooted at node v_j, we first define $2k_j + 1$ information elements and then assign them one by one to the leaf nodes $v_1^j, \ldots, v_{2k_j+1}^j$ of this subtree. If variable X_j occurs unnegated in clause C_i, we create an information element x_i^j with initial lifetime $[r_i^j, d_i^j] = [(3i+1)(n+1) + j, (3i+2)(n+1) + j + t_2 + t_3]$. If X_j occurs negated in clause

C_i, we create an information element with initial lifetime x_i^j : $[r_i^j, d_i^j] = [3i(n+1)+j, (3i+1)(n+1)+j+t_2+t_3]$. Let $i_1^j < \ldots < i_{k_j}^j$ denote the indices of the clauses in which variable X_j occurs. For every two messages $x_{i_t^j}^j$ and $x_{i_{t+1}^j}^j$, $t = 1, \ldots, k_j - 1$, define an information element $ax_{i_t^j}^j$ with initial lifetime: $[r_{a_t}^j, d_{a_t}^j] = [d_{i_t^j}^j - t_2 - t_3, r_{i_{t+1}^j}^j + t_2 + t_3]$. We also define the initial lifetime of $ax_0^j : [r_{a_0}^j, d_{a_0}^j] = [j, r_{i_1^j}^j + t_2 + t_3]$, and the initial lifetime of $ax_{k_j}^j : [r_{a_{k_j}}^j, d_{a_{k_j}}^j] = [d_{i_{k_j}^j}^j - t_2 - t_3, 3(m+1)(n+1)+j+t_2+t_3]$. After defining these $2k_j+1$ information elements' initial lifetime, we set the source of each element one by one from node v_1^j to node $v_{2k_j+1}^j$. For each element e with an initial lifetime $[r_e, d_e]$, define the real lifetime of e to be $[0, d_e]$ and the transmission time from the source of e to node v as r_e. For each node v_0^j, we define an element z_0^j: $[0, j+t_2+t_3]$ and the transmission time from v_0^j to v to be j. For each node $v_{2k_j+2}^j$, we define an element $z_{2k_j+2}^j : [0, 3(m+1)(n+1)+j+t_2+t_3]$ and the transmission time from $v_{2k_j+2}^j$ to v to be $3(m+1)(n+1)+j$.

Similarly, we define m information elements generated by nodes $v_1^c, \ldots, v_m^c$, with element $z_i : [r_i, d_i] = [0, (3i+2)(n+1)+t_2+t_3]$, $i = 1, \ldots, m$, being generated by node v_i^c, and the transmission time from v_i^c to v to be $(3i+1)(n+1)+t_2$. Then, for nodes v_1 to v_n, we define an information element for each of them with lifetime $[0, t_2+t_3]$, $i = 1, \ldots, n$. For node v, define an information element with lifetime $[0, t_3]$.

The whole process to assign an information element for each sensor will take $O(nm)$ time. Therefore, the time complexity of the whole transformation is $O(n) + O(nm) + O(nm) = O(nm)$, which is polynomial in n and m.

After we construct an instance Π' of $\mathbb{P}_0'$ from an instance Π of the SAT problem, we can easily follow the steps in proving Theorem 1 to prove this theorem. ■

Theorem 10: When $K = 2$ and re-aggregation is allowed, problem $\mathbb{P}_0'$ is strong NP-hard.

Proof: Given an instance Π of SAT problem with n Boolean variables $X_1, \ldots, X_n$ and m clauses $C_1, \ldots, C_m$, we derive a polynomial time transformation from Π to an instance Π' of problem $\mathbb{P}_0'$ with $K = 2$ as follows. The transformation is the same as what we present through Figure 34 except for the following changes:

- Define a node p between node v and node s, and m children $p_1, \ldots, p_m$ of node p. Additionally, define $ETX_{vp} = ETX_{ps} = ETX_{p_ip} = 1$, and $t_{vp} = t_3$, and $t_{ps} = t_5$.
- Define m information elements g_i's generated by nodes $p_1, \ldots, p_m$: $g_i : [r_i^p, d_i^p] = [0, (3i+1)(n+1)+n+0.1+t_2+t_3+t_5]$ and the transmission time from p_i to p is $(3i+1)(n+1)+n+0.1+t_2+t_3$. For node p, define an information element g with lifetime $[0, t_5]$.
- For all parameters defined during the transformation in Figure 4, replace t_3 by $t_3 + t_5$.

Therefore, the time complexity of the new transformation is still $O(nm)$.

After constructing this tree, we can follow the same steps in proving Theorem 4 to prove the correctness of this theorem. ■

Appendix 4: proof of Claim 8

Proof of Claim 8: 1) Given a satisfying assignment for the SAT problem, an optimal packing scheme of the corresponding packet packing problem can be derived as follows: If in the assignment of SAT problem, variable X_j is set true, then all information elements x_i^j are forwarded from their sources to node v_j at time r_i^j, and are forwarded from node v_j to node v at time $r_i^j + t_1$. If X_j is set false, then all information elements x_i^j are forwarded from their sources to node v_j at time r_i^j, and are forwarded from node v_j to node v at $d_i^j - (t_2 + t_3 + t_5)$. Similarly with the information elements generated by children nodes of node v_j, every information element generated by node $v_i^c, i = 1, \ldots, m$, cannot get packed at its source since v_i^c is a leaf node. As a result, each information element z_i is forward by its source and arrives at node v at time $(3i+1)(n+1)+t_1+t_2-t_4+t_4 = (3i+1)(n+1)+t_1+t_2$. Then the spare period for information element z_i to wait at node v is $[(3i+1)(n+1)+t_1+t_2, (3i+2)(n+1)+t_1+t_2]$. If clause C_i is satisfied by setting X_j to be true, then information element x_i^j arrives at node v at $(3i+1)(n+1)+t_1+t_2+j \in [(3i+1)(n+1)+t_1+t_2, (3i+2)(n+1)+t_1+t_2]$, which implies that z_i can be packed with the packet containing information element x_i^j. Similarly, if clause C_i is satisfied by setting X_j to be false, then information element x_i^j arrives at node v at $(3i+1)(n+1)+t_1+t_2+j \in [(3i+1)(n+1)+t_1+t_2, (3i+2)(n+1)+t_1+t_2]$, which implies z_i can be packed with the packet containing information element x_i^j. However, due to the packet size constraint, one packet cannot contain more than 2 information elements. In the meantime, every information element generated by node p_i cannot get packed at its source since node p_i is a leaf node. Thus each information element g_i is forwarded by its source and arrives at node p at time $(3i+1)(n+1)+n+0.1+t_1+t_2+t_3$. Then the spare period for element g_i to wait at node p is 0. In this case, to minimize the total number of transmission, if clause C_i is satisfied by setting X_j to be true, information element x_i^j arrives at node v with information element ax_{i-1}^j at time $(3i+1)(n+1)+t_1+t_2+j$ in one packet. When this packet arrives at v, information element ax_{i-1}^j and information element z_i form a new packet while information element x_i^j waits at v until $(3i+1)(n+1)+t_1+t_2+n+0.1$. x_i^j arrives at node g at time $(3i+1)(n+1)+n+0.1+t_1+t_2+t_3$ and forms a new packet with information element g_i. In this scheme, ax_{i-1}^j first packed x_i^j at node v_j, then leaves x_i^j at node v so that x_i^j can pack another information element g_i some time later at node p, which implies that a carry-over operation is used to achieve the optimal packing scheme. Similarly, if clause C_i is satisfied by setting X_j to be false, element x_i^j is arrives at node v with element ax_i^j at time $(3i+1)(n+1)+t_1+t_2+j$ in one packet. When this packet arrives at v, information element x_i^j and information element z_i form a new packet while information element ax_i^j waits at v until $(3i+1)(n+1)+t_1+t_2+n+0.1$. ax_i^j arrives at node p at time $(3i+1)(n+1)+n+0.1+t_1+t_2+t_3$ and forms a new packet with information element g_i. In this scheme, x_i^j first packed ax_i^j at node v_j, then leaves ax_i^j

at node v so that ax_i^j can pack another information element g_i some time later at node p, which implies that a carry-over operation is used to achieve the optimal packing scheme. An demonstration on how the optimal packing scheme is derived is given in Figure 35.

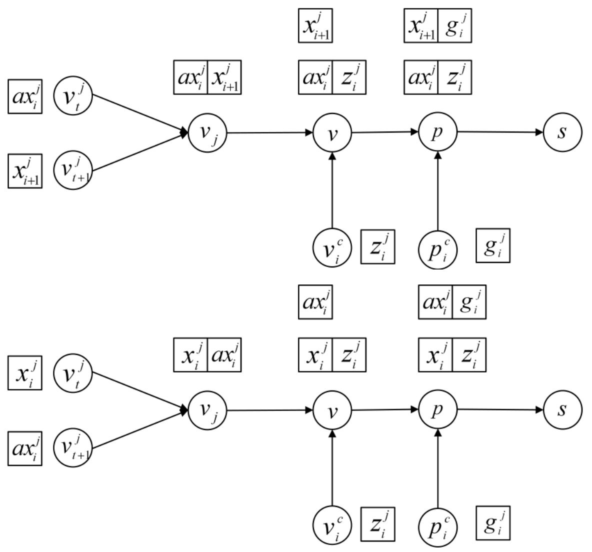


Fig. 35. Example of deriving optimal packing scheme from SAT assignment when $K = 2$

In the optimal packing scheme, every information element z_i can be packed at node v with an information element x_i^j or ax_{i-1}^j if clause C_i is satisfied due to variable X_j. Therefore, the additional number of transmission to send each information element z_i to node s is m, and the additional number of transmission to send each information element g_i to node s is m, and the additional number of transmission to break up m packet at node v and send them to node s is $2m$. As a result, the total number of transmission for this tree is $C'_{t0} + 4m = C'_{t1}$.

2) If we may find that the optimal packing scheme has a total number of transmission C'_{t1}, which implies that every information element z_i pack with one information element in a packet consisting of x_i^j for some j value, and the other information element in the old packet packs with information element g_i. If x_i^j leaves from node v_j at time $r_i^j + t_1$, and z_i packs with one information element in the packet that contains x_i^j at node v, this can only happen when X_j is unnegated in clause C_i because $(3i+1)(n+1)+t_1+t_2+j \in [(3i+1)(n+1)+t_1+t_2, (3i+2)(n+1)+t_1+t_2]$ and $3i(n+1)+j \notin [(3i+1)(n+1)+t_1+t_2, (3i+2)(n+1)+t_1+t_2]$. Thus we set X_j to be true. If x_i^j leaves from node v at time $d_i^j - (t_2+t_3+t_5)$, and z_i packs with one information element in the packet that contains x_i^j at node v, this can only happen when X_j is negated in clause C_i because $(3i+1)(n+1)+t_1+t_2+j \in [(3i+1)(n+1)+t_1+t_2, (3i+2)(n+1)+t_1+t_2]$ and $(3i+2)(n+1)+j+t_1+t_2 \notin [(3i+1)(n+1)+t_1+t_2, (3i+2)(n+1)+t_1+t_2]$. Thus we set X_j to be false. By this way, if we have an optimal solution to this instance of packet packing problem, we can have a satisfying assignment of the original SAT problem. Note that due to Claim 7, the following case cannot happen: element z_i gets packed with x_i^j by letting x_i^j leaves node v_j at time $r_i^j + t_1$, and in the meantime, that element z_k gets packed with x_k^j by letting x_k^j leaves node v_j at time $d_k^j - (t_2+t_3+t_5)$.

$\square$

Appendix 5: proofs of Theorem 5

We first show that the reduction presented in Figure 6 is a gap-preserving reduction [21] from MAX-3SAT to problem $\mathbb{P}'_0$. It is easy to verify that the proof of Theorem 4 holds if the discussion of the proof is based 3SAT instead of the general SAT problem, in which case $\sum_{j=1}^n k_j = 3m$ and we denote the reduction as f. Therefore, if a 3SAT problem Π is satisfiable, the minimum cost of the $\mathbb{P}'_0$ problem $\Pi' = f(\Pi)$ is

$$\begin{aligned} C'_{t1} &= C'_{t0} + 4m \\ &= (\textstyle\sum_{j=1}^n (2k_j+1) + \sum_{j=1}^n (k_j+1)(D+2) + \\ &\quad 2n(D+2) + 2n + 3) + 4m \\ &= m(3D+16) + n(3D+9) + 3 \end{aligned} \tag{25}$$

Since $n < 4m$, (25) implies that

$$\begin{aligned} C'_{t1} &< m(3D+16) + n(3D+16) \\ &< 5m(3D+16) \end{aligned} \tag{26}$$

Note that the proof of Theorem 4 holds if $D = n + \sum_{j=1}^n (2k_j+3) = 6m+n$, which is the number of information elements generated by the descendants of node v. Thus, (26) implies that

$$\begin{aligned} C'_{t1} &< 5m(3(6m+n)+16) \\ &= 5m(18m+3n+16) \\ &< 5m(18m+3\times 4m+16) \\ &= 5m(30m+16) \\ &< 5m(30m+16m) \\ &= 240m^2 \end{aligned} \tag{27}$$

If only m_0 of the m clauses in Π are satisfiable, then the minimum cost in $\Pi' = f(\Pi)$ (with $K = 3$ is $C'_{t1} + (m - m_0)$. This is because $(m - m_0)$ number of z_i's cannot be packed with any other packet and have to be sent from node v to s alone, which incurs an extra cost of 2 each. Accordingly, if less than m_0 of the m clauses in Π are satisfiable, then the minimum cost C' in $\Pi' = f(\Pi)$ is greater than $C'_{t1} + 2(m - m_0)$. Letting $\epsilon = \frac{m}{m_0}$, (27) implies that

$$\begin{aligned} \frac{C'}{C'_{t1}} &> \frac{C'_{t1}+2(m-m_0)}{C'_{t1}} \\ &= \frac{C'_{t1}+2(\epsilon m_0 - m_0)}{C'_{t1}} \\ &= 1 + 2\frac{(\epsilon-1)m_0}{C'_{t1}} \\ &> 1 + 2\frac{(\epsilon-1)m_0}{240m^2} \\ &= 1 + \frac{\epsilon-1}{120m}\frac{1}{\epsilon} \\ &= 1 + \frac{1}{120m}(1-\frac{1}{\epsilon}) \\ &\geq 1 + \frac{1}{120N}(1-\frac{1}{\epsilon}) \end{aligned} \tag{28}$$

where N is the number of non-sink nodes in the network and $N \geq m$.

Let $OPT(\Pi)$ and $OPT(\Pi')$ be the optima of a MAX-3SAT problem Π and the corresponding $\mathbb{P}'_0$ problem $\Pi' = f(\Pi)$.

Then the polynomial-time reduction f from MAX-3SAT to $\mathbb{P}'_0$ satisfy the following properties:

$$\begin{array}{lll} OPT(\Pi) = 1 & \Longrightarrow & OPT(\Pi') = C'_{t1} \\ OPT(\Pi) < \frac{1}{\epsilon} & \Longrightarrow & OPT(\Pi') > C'_{t1}(1 + \frac{1}{120N}(1 - \frac{1}{\epsilon})) \end{array} \tag{29}$$

From [21], we know that there exists a polynomial-time reduction f_1 from SAT to MAX-3SAT such that, for some fixed $\epsilon > 1$, reduction f_1 satisfies

$$\begin{array}{lll} I \in SAT & \Longrightarrow & \text{MAX-3SAT}(f_1(I)) = 1 \\ I \notin SAT & \Longrightarrow & \text{MAX-3SAT}(f_1(I)) < \frac{1}{\epsilon} \end{array} \tag{30}$$

Then, (29) and (30) imply the following:

$$\begin{array}{lll} I \in SAT & \Longrightarrow & OPT(f(f_1(I))) = C'_{t1} \\ I \notin SAT & \Longrightarrow & OPT(f(f_1(I))) > C'_{t1}(1 + \frac{1}{120N}(1 - \frac{1}{\epsilon})) \end{array} \tag{31}$$

Therefore, it is NP-hard to achieve an approximation ratio of $1 + \frac{1}{120N}(1 - \frac{1}{\epsilon})$ for problem $\mathbb{P}_0$.

Appendix 6: Utility calculation in tPack-2hop

In Section V we proposed a utility based online algorithm called *tPack* for packet packing problem. While *tPack* calculates the utility of immediately transmitting a packet based on only next one hop, it is trying to use the greedy approach to help node make a local decision, which may affect the performance of the whole network. To explore the effects brought by this 1-hop greedy decision making policy, we will derive the utility of transmitting a packet by looking at next 2 hops, which is called *tPack-2hop*.

Except the notations used in Section V, we define some additional notations in the following:

With respect to the grandparent of v_j:

t_g	:	expected time till the parent transmits another packet pkt''' that does not contain information elements generated or forwarded by v_j itself or its parent;
s_g	:	expected payload size of pkt'''.

Utility of holding a packet. In *tPack-2hop*, the holding utility is the same as it was in the old *tPack*, which is calculated as:

$$\begin{aligned} U_l &= AC'_l - AC_l \\ &= \tfrac{1}{L-s'_f} ETX_{jR}(L - s'_f) \\ &\quad -\tfrac{1}{L-s'_f+\mathcal{S}_l} ETX_{jR}(L - s'_f + \mathcal{S}_l) \end{aligned} \tag{32}$$

Utility of immediately transmitting a packet. Different from the old *tPack*, the utility of immediately transmitting a packet in *tPack-2hop* depends on the utility to transmitting a packet to the next hop or to the next 2 hops. Similar with the definition of P_{pkt}, the expected number of packets that do not contain information elements from v_j and can be packed with pkt at v_j's grandparent g_j is $\frac{t'_f}{t_g}$, and we define P'_{pkt} to be the set of these packets. The utility to immediately transmit a packet to p_j, v_j's parent, is computed as follows:

- If *every packet in P_{pkt} and P'_{pkt} gets packed to full* with payload from pkt and every packet, i.e., $\frac{t'_f}{t_p}(L - s_p) \leq L - s'_f$ and $\frac{t'_f}{t_g}(L - s_g) \leq L - s'_f - \frac{t'_f}{t_p}(L - s_p)$:

  Then, the overall utility U'_p is

$$\begin{aligned} U'_p &= \frac{\frac{t'_f}{t_p} ETX_{p_j R}(s_p)}{\frac{t'_f}{t_p} s_p} - \frac{\frac{t'_f}{t_p} ETX_{p_j R}(L)}{\frac{t'_f}{t_p} L} \\ &\quad + \frac{\frac{t'_f}{t_g} ETX_{g_j R}(s_g)}{\frac{t'_f}{t_g} s_g} - \frac{\frac{t'_f}{t_g} ETX_{g_j R}(L)}{\frac{t'_f}{t_g} L} \\ &= \frac{ETX_{p_j R}(s_p)}{s_p} - \frac{ETX_{p_j R}(L)}{L} \\ &\quad + \frac{ETX_{g_j R}(s_g)}{s_g} - \frac{ETX_{g_j R}(L)}{L} \end{aligned} \tag{33}$$

- If *every packet in P_{pkt} gets packed to full* with payload from pkt but *not all packets in P'_{pkt} can get fully packed*, i.e., $\frac{t'_f}{t_p}(L - s_p) \leq L - s'_f$ and $\frac{t'_f}{t_g}(L - s_g) > L - s'_f - \frac{t'_f}{t_p}(L - s_p)$:

  Denoting mod$(L - s'_f - \frac{t'_f}{t_p}(L - s_p), L - s_g)$ by g_{mod} and letting G_{mod} be 1 if $g_{mod} > 0$ and 0 otherwise. Let $G = G_{mod} ETX_{g_j R}(s_g + g_{mod})$, the overall utility U''_p is

$$\begin{aligned} U''_p &= \frac{\frac{t'_f}{t_p} ETX_{p_j R}(s_p)}{\frac{t'_f}{t_p} s_p} - \frac{\frac{t'_f}{t_p} ETX_{p_j R}(L)}{\frac{t'_f}{t_p} L} \\ &\quad + \frac{\lceil \frac{L-s'_f}{L-s_g} - \frac{t'_f}{t_p}(L-s_p) \rceil ETX_{g_j R}(s_g)}{\lceil \frac{L-s'_f}{L-s_g} - \frac{t'_f}{t_p}(L-s_p) \rceil s_g} \\ &\quad - \frac{\lfloor \frac{L-s'_f}{L-s_g} - \frac{t'_f}{t_p}(L-s_p) \rfloor ETX_{g_j R}(L) + G}{\lceil \frac{L-s'_f}{L-s_g} - \frac{t'_f}{t_p}(L-s_p) \rceil s_g + L - s'_f} \\ &= \frac{ETX_{p_j R}(s_p)}{s_p} - \frac{ETX_{p_j R}(L)}{L} \\ &\quad + \frac{\lceil \frac{L-s'_f}{L-s_g} - \frac{t'_f}{t_p}(L-s_p) \rceil ETX_{g_j R}(s_g)}{\lceil \frac{L-s'_f}{L-s_g} - \frac{t'_f}{t_p}(L-s_p) \rceil s_g} \\ &\quad - \frac{\lfloor \frac{L-s'_f}{L-s_g} - \frac{t'_f}{t_p}(L-s_p) \rfloor ETX_{g_j R}(L) + G}{\lceil \frac{L-s'_f}{L-s_g} - \frac{t'_f}{t_p}(L-s_p) \rceil s_g + L - s'_f} \end{aligned} \tag{34}$$

- If *not every packet in P_{pkt} gets packed to full* with payload from pkt, i.e., $\frac{t'_f}{t_p}(L - s_p) > L - s'_f$: The utility is computed as it was in the old *tPack*:

$$\begin{aligned} U'''_p &= \frac{\lceil \frac{L-s'_f}{L-s_p} \rceil ETX_{p_j R}(s_p)}{\lceil \frac{L-s'_f}{L-s_p} \rceil s_p} - \\ &\quad \frac{\lfloor \frac{L-s'_f}{L-s_p} \rfloor ETX_{p_j R}(L) + I_{mod} ETX_{p_j R}(s_p + l_{mod})}{\lceil \frac{L-s'_f}{L-s_p} \rceil s_p + L - s'_f} \end{aligned} \tag{35}$$

Therefore, the utility U_p of immediately transmitting pkt to p_j in *tPack-2hop* is

$$U_p = \begin{cases} U'_p & \text{if } \frac{t'_f}{t_p}(L - s_p) \leq L - s'_f \\ & \text{and } \frac{t'_f}{t_g}(L - s_g) \leq L - s'_f - \frac{t'_f}{t_p}(L - s_p) \\ U''_p & \text{if } \frac{t'_f}{t_p}(L - s_p) \leq L - s'_f \\ & \text{and } \frac{t'_f}{t_g}(L - s_g) > L - s'_f - \frac{t'_f}{t_p}(L - s_p) \\ U'''_p & \text{otherwise} \end{cases} \tag{36}$$

where U'_p, U''_p and U'''_p are defined in Equations (33) , (34) and (35) respectively.

After computing U_p, we need calculate another utility, the utility to immediate forwarding pkt to g_j, v_j's grandparent. We can compute this utility as same as we did to get U_p.

- If *every packet in P'_{pkt} and P_{pkt} gets packed to full* with payload from pkt and every packet, i.e., $\frac{t'_f}{t_g}(L - s_g) \leq L - s'_f$ and $\frac{t'_f}{t_p}(L - s_p) \leq L - s'_f - \frac{t'_f}{t_g}(L - s_g)$:
  Then, the overall utility U'_g is

$$\begin{aligned} U'_g &= \frac{\frac{t'_f}{t_g}ETX_{g_jR}(s_g)}{\frac{t'_f}{t_g}s_g} - \frac{\frac{t'_f}{t_g}ETX_{g_jR}(L)}{\frac{t'_f}{t_g}L} \\ &\quad + \frac{\frac{t'_f}{t_p}ETX_{p_jR}(s_p)}{\frac{t'_f}{t_p}s_p} - \frac{\frac{t'_f}{t_p}ETX_{p_jR}(L)}{\frac{t'_f}{t_p}L} \\ &= \frac{ETX_{g_jR}(s_g)}{s_g} - \frac{ETX_{g_jR}(L)}{L} \\ &\quad + \frac{ETX_{p_jR}(s_p)}{s_p} - \frac{ETX_{p_jR}(L)}{L} \end{aligned} \tag{37}$$

- If *every packet in P'_{pkt} gets packed to full* with payload from pkt but *not all packets in P_{pkt} can get fully packed*, i.e., $\frac{t'_f}{t_g}(L - s_g) \leq L - s'_f$ and $\frac{t'_f}{t_p}(L - s_p) > L - s'_f - \frac{t'_f}{t_g}(L - s_g)$:
  Denoting mod$(L - s'_f - \frac{t'_f}{t_g}(L - s_g), L - s_p)$ by p_{mod} and letting P_{mod} be 1 if $p_{mod} > 0$ and 0 otherwise. Let $P = P_{mod}ETX_{p_jR}(s_p + p_{mod})$ the overall utility U''_g is

$$\begin{aligned} U''_g &= \frac{\frac{t'_f}{t_g}ETX_{g_jR}(s_g)}{\frac{t'_f}{t_g}s_g} - \frac{\frac{t'_f}{t_g}ETX_{g_jR}(L)}{\frac{t'_f}{t_g}L} \\ &\quad + \frac{\lceil\frac{L-s'_f}{L-s_p} - \frac{t'_f}{t_g}(L-s_g)\rceil ETX_{p_jR}(s_p)}{\lceil\frac{L-s'_f}{L-s_p} - \frac{t'_f}{t_g}(L-s_g)\rceil s_p} \\ &\quad - \frac{\lfloor\frac{L-s'_f}{L-s_p} - \frac{t'_f}{t_g}(L-s_g)\rfloor ETX_{p_jR}(L)+P}{\lceil\frac{L-s'_f}{L-s_p} - \frac{t'_f}{t_g}(L-s_g)\rceil s_p + L - s'_f} \\ &= \frac{ETX_{g_jR}(s_g)}{s_g} - \frac{ETX_{g_jR}(L)}{L} \\ &\quad + \frac{\lceil\frac{L-s'_f}{L-s_p} - \frac{t'_f}{t_g}(L-s_g)\rceil ETX_{p_jR}(s_p)}{\lceil\frac{L-s'_f}{L-s_p} - \frac{t'_f}{t_g}(L-s_g)\rceil s_p} \\ &\quad - \frac{\lfloor\frac{L-s'_f}{L-s_p} - \frac{t'_f}{t_g}(L-s_g)\rfloor ETX_{p_jR}(L)+P}{\lceil\frac{L-s'_f}{L-s_p} - \frac{t'_f}{t_g}(L-s_g)\rceil s_p + L - s'_f} \end{aligned} \tag{38}$$

- If *not every packet in P'_{pkt} gets packed to full* with payload from pkt, i.e., $\frac{t'_f}{t_g}(L - s_g) > L - s'_f$: Denoting mod$(L - s'_f, L - s_g)$ by gr_{mod} and letting GR_{mod} be 1 if $gr_{mod} > 0$ and 0 otherwise,, the overall utility U'''_g is

$$\begin{aligned} U'''_g &= \frac{\lceil\frac{L-s'_f}{L-s_g}\rceil ETX_{g_jR}(s_g)}{\lceil\frac{L-s'_f}{L-s_g}\rceil s_g} - \\ &\quad \frac{\lfloor\frac{L-s'_f}{L-s_g}\rfloor ETX_{g_jR}(L) + GR_{mod}ETX_{g_jR}(s_g+gr_{mod})}{\lceil\frac{L-s'_f}{L-s_g}\rceil s_g + L - s'_f} \end{aligned} \tag{39}$$

Therefore, the utility U_p of immediately transmitting pkt to p_j in *tPack-2hop* is

$$U_g = \begin{cases} U'_g & \text{if } \frac{t'_f}{t_g}(L - s_g) \leq L - s'_f \\ & \text{and } \frac{t'_f}{t_p}(L - s_p) \leq L - s'_f - \frac{t'_f}{t_g}(L - s_g) \\ U''_g & \text{if } \frac{t'_f}{t_p}(L - s_p) \leq L - s'_f \\ & \text{and } \frac{t'_f}{t_p}(L - s_p) > L - s'_f - \frac{t'_f}{t_g}(L - s_g) \\ U'''_g & \text{otherwise} \end{cases} \tag{40}$$

where U'_g, U''_g and U'''_g are defined in Equations (37) , (38) and (39) respectively.

Therefore, the utility U_p of immediately transmitting pkt to p_j in *tPack-2hop*is

$$U_{im} = max(U_p, U_g) \tag{41}$$

Therefore, the scheduling rule in *tPack-2hop* is that the packet should be immediately transmitted if $U_l < U_{im}$, otherwise the packet should wait at node v_j.

Approximation of s_g and t_g: parameter s_g and t_g are approximated as same as s_p and t_p was approximated.

$$s_g = s_{g_j} \tag{42}$$

$$t_g = \frac{1}{r_g} = \frac{t_{g_j} \times t_{p_j} \times s_{g_j}}{t_{p_j} \times s_{g_j} - t_{g_j} \times s_{p_j}} \tag{43}$$

Appendix 7: Complexity of $\mathbb{P}_0$ when ETX is a convex function of packet length

In this section, we will analyze the computation complexity of $\mathbb{P}_0$ when ETX is a convex function of packet length. To start with, we first define the following two problems.

Problem $\mathbb{P}_0^{CC}$: The same as $\mathbb{P}_0$ except that 1) the network is a chain network. 2) ETX is a convex function of packet length.

Problem $\mathbb{P}_0^{EC}$: The same as $\mathbb{P}_0$ except that 1) the network is a chain network. 2) ETX is an exponential function of packet length.

We first prove $\mathbb{P}_0^{EC}$ is NP-hard. Then the NP-hardness of $\mathbb{P}_0^{CC}$ is easily proved.

Theorem 11: $\mathbb{P}_0^{EC}$ is NP-hard regardless of re-aggregation.

Proof:

We first study the case where re-aggregation is prohibited.

Lemma 1: $\mathbb{P}_0^{EC}$ is NP-hard when re-aggregation is prohibited.

Proof: We first define a new problem as follows:

Problem $\mathbb{P}_0^{subg}$: there are n elements generated at different time at node A, each of which has an individual deadline and an arbitrary integer length. A packet can hold at most K elements and ETX is an exponential function of packet length. Find a packing scheme to send all n elements from node A to its parent node B such that the sum of ETX of all packets are minimized.

Lemma 2: $\mathbb{P}_0^{subg}$ is NP-hard.

Proof: We prove this lemma by a reduction from the PARTITION problem.

PARTITION Problem: Given a finite set A and a size $s(a) \in Z^+$ for each $a \in A$, find a subset $A' \subseteq A$ such that $\sum_{a\in A'} s(a) = \sum_{a\in A-A'} s(a)$.

Given any instance X of PARTITION problem, we can reduce it to an instance Y of problem $\mathbb{P}_0^{subg}$:

Given a finite set A of elements and a size $s(a) \in Z^+$ for each $a \in A$, any two elements in A can be packed together and a packet can at most elements with length $\sum_{a\in A} s(a)$. ETX is an exponential function of packet length with base p, where $p > 1$. find a subset $A' \subseteq A$ such that $p^{\sum_{a\in A'} s(a)} + p^{\sum_{a\in A-A'} s(a)}$ is no greater than $2p^{\frac{\sum_{a\in A} s(a)}{2}}$.

We can easily find that there exists a solution to X if and only if there exists a solution to Y since $p^{\sum_{a\in A'} s(a)} + p^{\sum_{a\in A-A'} s(a)} \geq 2p^{\frac{\sum_{a\in A} s(a)}{2}}$ and the equal sign holds if and only if $\sum_{a\in A'} s(a) = \sum_{a\in A-A'} s(a)$. Therefore, problem $\mathbb{P}_0' - sub - general$ is at least as hard as the PARTITION problem, which means it is NP-hard. ■

Since problem $\mathbb{P}_0^{subg}$ is NP-hard, solving the following problem is also NP-hard.

multiple-$\mathbb{P}_0^{subg}$: Solve n instances of $\mathbb{P}_0^{subg}$ with total set size $|A(i)| = i, i = 1, ..., n$, and $A(i)$ equal to the partition of $A(i-1)$ plus an element with unit length.

The NP-hardness of this problem is out of question since we already proved Lemma 2. Given an instance X of multiple-$\mathbb{P}_0^{subg}$, we can transform it to an instance Y of $\mathbb{P}_0^{EC}$ as follows. Define a chain with $n+1$ nodes, labeled as $1, 2, ..., n, n+1$. Node $n+1$ is the sink and node 1 is the leaf node. Define the link reliability of link $(i, i+1)$, $p_{i,i+1}$ is far less than $p_{i+1,i+2}$. And put the ith $\mathbb{P}_0^{subg}$ on the node i. And define the latency requirement of each element to be very large so that each two unit element can get packed.

Now we proved that solving Y is equivalent to solve X. Since $p_{i,i+1} \ll p_{i+1,i+2}$, to get the optimal solution to Y, we need solve the first $\mathbb{P}_0^{subg}$ problem, then the second, and so on, which is exactly the same way to solve X. Therefore, $\mathbb{P}_0^{EC}$ is NP-hard when reaggregation is prohibited. ■

Next, we study the complexity of $\mathbb{P}_0^{EC}$ when re-aggregation is not prohibited.

Lemma 3: $\mathbb{P}_0^{EC}$ is NP-hard when re-aggregation is not prohibited.

Proof: We define another new problem as follows:

Problem $\mathbb{P}_0^{sub}$: there are n elements generated at different time at node A, each of which has an individual deadline and the same integer length. A packet can hold at most K elements and ETX is an exponential function of packet length. Find a packing scheme to send all n elements from node A to its parent node B such that the sum of ETX of all packets are minimized.

Lemma 4: $\mathbb{P}_0'^{sub}$ is NP-hard.

Proof: We can model $\mathbb{P}_0^{sub}$ on a multiple-interval graph model. The transformation from interval graph to multiple-interval graph is straightforward. For each node v, we define a set S_v of intervals. Any two nodes v and u are adjacent if and only if $S_v \bigcup S_u \neq \emptyset$. We prove $\mathbb{P}_0^{sub}$ by a reduction from PARTITION INTO TRIANGLES problem.

The NP-hardness of the PARTITION INTO TRIANGLES problem is proved by Schaefer in 1974 by a reduction from X3C problem. There is another proof by Mirko Morandini in [46]. In this paper, PARTITION INTO TRIANGLES problem is proved to be NP-hard via a simpler reduction from 3DM problem. The author proved that PARTITION INTO TRIANGLES problem is NP-hard even in 3-partite graph.

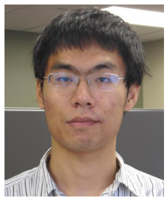
In [47], the authors showed that any r-partite graph can be modeled as a multiple-interval graph. Therefore, the PARTITION INTO TRIANGLES problem is also NP-hard in multiple-interval graph.

We can then easily make a reduction from any instance of PARTITION INTO TRIANGLES to $\mathbb{P}_0^{sub}$ since every instance of PARTITION INTO TRIANGLES is an instance of $\mathbb{P}_0^{sub}$ with packet size equal to 3 and a link reliability ensuring 3 is the optimal packet size. Therefore, the problem $\mathbb{P}_0^{sub}$ is NP-hard. ■

Combining the proof when re-aggregation is prohibited and the previous lemma, we can easily prove that $\mathbb{P}_0^{EC}$ is NP-hard when re-aggregation is not prohibited.. ■

Therefore, $\mathbb{P}_0^{EC}$ is NP-hard regardless of re-aggregation. ■

Since $\mathbb{P}_0^{EC}$ is also a special case of $\mathbb{P}_0^{CC}$, $\mathbb{P}_0^{CC}$ is also NP-hard.



Qiao Xiang Qiao Xiang is a PhD candidate in the Department of Computer Science at Wayne State University. He received his Bachelor degrees in Engineering and Economics from Nankai University, China. His research interests lie in wireless cyber-physical systems.

Hongwei Zhang Hongwei Zhang (S'01-M'07/ACM S'01-M'07) received his B.S. and M.S. degrees in Computer Engineering from Chongqing University, China and his Ph.D. degree from The Ohio State University. He is currently an assistant professor of computer science at Wayne State University. His primary research interests lie in the modeling, algorithmic, and systems issues in wireless, vehicular, embedded, and sensor networks. His research has been an integral part of several NSF, DARPA projects such as the KanseiGenie and ExScal projects. (URL: http://www.cs.wayne.edu/∼hzhang).

Jinhong Xu Jinhong Xu is Master student of Financial Risk Management at Simon Fraser University. His research interests in computer science include pervasive and mobile computing as well as wireless sensor networks.

Xiaohui Liu Xiaohui Liu received his B.S. degree in Computer Science from Wuhan University, China. He is currently a PhD candidate in the Department of Computer Science at Wayne State University. His primary research interests lie in real-time, QoS routing in wireless and sensor networks. He is a student member of ACM.

Loren J. Rittle Loren J. Rittle received the B.S. degree in computer engineering from Iowa State University in 1990. He is currently a principal staff member of the Content and Context-aware Solutions group, Applied Research, Motorola Mobility. He is named inventor on eight issued US patents. He is a member of the ACM.