DEAR: Delay-bounded Energy-constrained Adaptive Routing in Wireless Sensor Networks

Shi Bai, Weiyi Zhang*, Guoliang Xue†, Jian Tang‡, and Chonggang Wang‡

*Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455
†AT&T Labs Research, Middletown, NJ 07748
‡School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287

Abstract—Reliability and energy efficiency are critical issues in wireless sensor networks. In this work, we study Delay-bounded Energy-constrained Adaptive Routing (DEAR) problem with reliability, differential delay, and transmission energy consumption constraints in wireless sensor networks. We aim to route the connections in a manner such that link failure does not shut down the entire stream but allows a continuing flow for a significant portion of the traffic along multiple paths. This flexibility enabled by a multi-path routing scheme has the tradeoff of differential delay among the different paths. This requires increased memory in the base station to buffer the traffic until the data arrives on all the paths. Therefore, differential delay between the multiple paths should be bounded in a range to reduce additional hardware cost in the base station. Moreover, the energy consumption constraint should also be satisfied when transmitting packets among multiple paths. We present a pseudo-polynomial time solution to solve a special case of DEAR, representing edge delays as integers. Next, an \((1+\alpha)\)-approximation algorithm is proposed to solve the optimization version of the DEAR problem. An efficient heuristic is provided for the DEAR problem. We present numerical results confirming the advantage of our schemes as the first solution for the DEAR problem.

Keywords: wireless sensor network; multi-path routing; differential delay; polynomial time approximation; restricted maximum flow.

I. INTRODUCTION

Many wireless sensor network (WSN) applications have been emerging in recent years due to the commercially available low-cost and diverse sensors. Rapid improvements in low-cost hardware have prompted the development of Wireless Multimedia Sensor Networks (WMSNs), in which sensors can be equipped with audio and visual information collection modules, and collect multimedia data such as video and audio streams, images, and scalar data from environments [4]. WSNs have also been deployed for data collection in habitat monitoring where most traffic is multi-hop multi-point-to-point converge cast in the upstream from sensor nodes to the base station. Thus, WSN applications could be delay-sensitive such as in mission-critical sensor networks and smart grids, and may demand high reliability in packet-level and event-level [32]. Routing and transmitting data from sensors to the base station is probably the most important function in WSNs. In communication networks, including WSNs, routing is commonly along the shortest path [5] [6]. However, these protocols are not able to support applications which require a certain minimum end-to-end bandwidth together with a bounded start-up delay to meet high user satisfaction [7], especially in resource-limited WSNs. As a result, researchers have proposed to use multi-path routing to support high bandwidth multimedia applications in bandwidth limited networks [8], [30]. In many cases, splitting the traffic over multiple paths may provide a lower cost solution than finding a single path meeting requested capacity. Meanwhile, simultaneous routing along multiple disjoint paths can result in increased resiliency against network failures. Extensive work has been done on multipath routing in both wired and wireless networks [18], [23], [26], [27], [31].

However, most previous protection/restoration schemes were designed for the all-or-nothing protection which causes an overkill for data traffic. Though the provisioning of two disjoint paths provides better survivability, it imposes at least a 100% protection bandwidth overhead. In [1], the authors argued that while achieving the reliability goals, multiple disjoint path scheme makes very inefficient use of network resource. While voice generates constant bit rate traffic, data traffic is bursty giving the advantage that data applications can continue operation, possibly at a lowered performance, even if the capacity along the path is reduced. In other words, unlike voice which has a binary service up or down condition, data services can survive gradual degradation as the available bandwidth reduces. Consequently, Acharya et al. [1] proposed a scheme named PESO which aims to route the connections such that link (or node failure) does not shut down the entire stream but allows important amount of traffic to still continue to flow. PESO allows the traffic to be split and routed along multiple paths such that a single link failure does not affect more than \(x\%\) of the total bandwidth. Thus, PESO creates a novel way to look at the bandwidth overbuild-reliability tradeoff.

Meanwhile, there are extensive works which focused on the method that how to split, reassemble and re-construct the traffic packets [1], [10], [21]. The source node partitions the data into many parts using coding theory and transmits each part along a different path. By using this method, all packets received from these paths are needed to be intercepted and store in the buffer memory in the destination node in order to find out what is transmitted in the process. But the ability to split
and route traffic separately introduces a unique problem. When traffic is routed over different physical paths, they may incur different amounts of delay and thus, reach at different times. This difference of delay of paths is popularly called differential delay. Presence of differential delay requires increased memory in the destination node to buffer the traffic until the data arrives on all the paths [10], [21]. In turn, this potentially raises the network element cost, making deployments more expensive. More seriously, buffer overflows can cause data corruption and bring down the service. Obviously, it is necessary to handle differential delay in order to correctly re-construct the data at the destination. Differential delay problem, introduced by Srivastava et al. in [28], motivated the need to study differential delay in routing algorithm for various classes of graph. The authors proposed the initial heuristic based solutions to solve this problem. [2], [3] introduced the concept of differential delay minimization when adding a new circuit to an existing group of circuits. It showed the hardness of the problem and proposed heuristics. In [29], the authors studied the cumulative differential delay problem, which seeks to minimize the sum of the difference of delays of all the paths of a solution compared to the highest delay path. The authors of [16] applied the differential delay constraint to mesh networks.

Another important problem for wireless sensor networks is the network lifetime. While the functionalities of sensor nodes are improving as a result of technical innovation, sensor nodes are still constrained by the limited power conservation [19]. Hence, it is significant to use of energy of sensors efficiently to increase the lifetime of WSNs. [13] proposed a braided multi-path scheme which is viable for energy-efficient recovery from isolated and pattern failures. In [25], Schurgers et al. proposed a practical guideline that advocates a uniform resource utilization to achieve energy efficient routing in wireless sensor networks.

In this work, we will study an Delay-bounded Energy-constrained Adaptive Routing (DEAR) problem that considers adaptive reliability, differential delay, and deliverable energy constraints. To the best of our knowledge, this is the first work that jointly studies the adaptive multipath routing, differential delay, and energy consumption problems in WSNs.

The rest of this paper is organized as follows. We define the DEAR problem and show our contribution in Section II. In Section III, a pseudo-polynomial time solution is presented for a special case of DEAR, representing edge delays and differential delay bounds as integers. Then, an (1 + α)-approximation algorithm is proposed to solve the optimization version of the DEAR problem in Section IV, which is followed by an efficient heuristic algorithm for DEAR in Section V. We present numerical results in Section VI to support our theoretical analysis, and conclude our work in Section VII.

II. PROBLEM STATEMENT

In this section, we first demonstrate some observation which was not fully considered by previous work. Then we present a new reliable routing problem DEAR based on our observation.

A. Background and definitions

A wireless sensor network usually consists of at least one base station (BS) and a set of resource-constraint sensor nodes physically scattered around the base station. In many sensor systems, the base station is a machine which has high computational capability, sufficient memory, and unlimited power supply [33]. Therefore, we can take advantages of high computational capabilities of BS to perform our multi-path routing scheme. During the system construction process, each node will report the identity of its neighbors to BS. So the base station will obtain the network topology of this particular sensor network.

We use a graph to model a wireless sensor network (WSN). Given a WSN in Fig. 1(a), we model the network as a directed graph $G = (V, E, b, d, w, \beta)$ in Fig. 1(b), where $V$ represents the set of sensor nodes and BS, and $E$ represents the set of links. Each link $e = (u, v)$ is associated with a bandwidth $b(e) > 0$, a propagation delay $d(e) \geq 0$, and a transmission energy consumption $w(e) > 0$ [24]. For each node $v \in V$, we use $\tau(v)$ and $\beta(v)$ to denote the transmission delay and the residual energy of sensor $v$, respectively.

**Definition 1 (Packet Allocation).** Let $P$ be a set of $s$–BS paths, where each path $p \in P$ is associated with a packet allocation $L(p)$ which is the allocated packets through the path $p$. The aggregated packet of link $e$, denoted by $q(e)$, is the sum of the packet allocations on link $e$ of the paths in $P$:

$$q(e) = \sum_{e \in P, p \in P} L(p).$$

**Definition 2 (Differential delay).** Let $P = \{p_1, p_2, \ldots, p_m\}$ be the set of paths from the source node $s$ to the base station BS. The delay of $P$ which is denoted by $d(P)$, is the delay of the longest path in $P$. Let $d_h$ and $d_i$ be the highest and lowest path delays in $P$, respectively, then the differential delay $D_P$ of paths in $P$ is defined as $D_P = d_h - d_i$.

**Definition 3 (Energy consumption).** For a sensor node $s_i$, the total energy consumption is defined as $E(i) = \sum_{e \in o(i)} q(e)w$, where $o(i)$ is the set of links from $s_i$ to its neighbors on multi-paths, $q(e)$ is the packet size transmitted on link $e$, and $w$ is the energy consumption of transmitting 1 bit.

We model the transmission energy consumption using the modulation scaling studied in [25] [36]. The energy consumption of transmitting 1 bit can be calculated as $w = [C \cdot (2^p - 1) + F] \cdot \frac{1}{\tau}$, where $C$ is the parameter determined by the quality of transmission and noise power, and $F$ is the parameter estimates the
power consumption of the electronic circuitry of the transmitting node. In our work, assuming that the parameters $C$, $F$, and $b$ are fixed, we can easily see that the energy consumption of transmitting 1 bit is a constant value. Then, for a packet with packet size $q$, the transmitting energy consumption can be simply computed as $E = w \cdot q$.

**Definition 4 (Latency/delay).** Let $s$ be the source node and $\text{BS}$ the base station. A $s$-BS path is a sequence of nodes $x_0, x_1, \ldots, x_l$ in $V$ such that $x_0 = s$, $x_l = \text{BS}$, and $(x_{i-1}, x_i)$ is a link in $E$ for $i = 1, 2, \ldots, l$. When packets are transmitted from one node to another, it is known that the communication latency/delay consists of mainly three factors [24]:

- Queueing delay: the time waiting at output link for transmission
- Transmission delay: the amount of time required to push all of the packet bits into the transmission media
- Propagation delay: the time takes for the head of the signal to travel from the sender to the receiver

The delay studied in this paper consists of the transmission delays on the nodes and the propagation delays on the links. Thus, the delay of the path $p$ is defined as

$$d(p) = \sum_{e \in p} d(e) + \sum_{v \in p} \tau(v)$$

(2.2)

To compute the transmission delay on each node, we adopt the model in [24]. Given $q$ bits packet to be sent from transmitting node $v$ to receiving node $t$, with the rate of transmission, $R$ bps, the transmission delay can be calculated as $\tau(v) = \frac{q}{R}$.

**B. Observation: Transmission Delay is Critical**

As we mentioned that we include both transmission delay and queueing delay for latency, please note that most previous work only considered the propagation delay on the transmission links for delay calculation, and ignored the transmission delay and queueing delay. While ignoring transmission delay might be appropriate for most wireline networks connected by fibers and cables, we argue that transmission delay should not be ignored for the communications in the resource limited WSNs. Currently, most wireless sensing devices have about 100-meter maximum transmission range [37] [38]. Given the fact that electromagnetic radio travels at the speed of light, $3 \times 10^8$ meters/second, propagation delay between two sensors is no more than $3.33 \times 10^{-7}$ second. On the other hand, IEEE 802.15.4 radios can provide standard data rate at 250kbps [38], and non-standard higher data rate up to 2000kbps [17]. Thus, the transmission delay for sending out a small 2kb packet needs at least $1 \times 10^{-3}$ second. Comparing with the propagation delay, we observe that transmission delay is a dominant factor for network transmissions latency in WSNs, and cannot be ignored. Therefore, we claim that transmission delay should be considered within the routing strategy in WSNs.

More importantly, we observe that the transmission delay has significant impact on the measurement of differential delay in WSNs. To demonstrate our point, let us use the sample WSN in Fig. 1 for explanation. Various routing selections on the communication graph are shown in Fig. 2. Three metrics are marked on each link: bandwidth, propagation delay, and transmission energy consumption of the link. In our example, we need a routing strategy from $A$ to BS for total $Q = 12$ packets. Also we assume the transmission rate unit is packet per second, pk/s, and propagation delay is measured by seconds. Three candidate paths are listed for connection $A$–BS: $p_1 = (A, B, \text{BS})$, $p_2 = (A, C, \text{BS})$, and $p_3 = (A, \text{BS})$. Let us consider two cases:

**A). Without considering the Transmission delay**

If only the propagation delay is included, the delays of the three paths are $d(p_1) = 2$, $d(p_2) = 3$, $d(p_3) = 2$, respectively (shown in Fig. 1(b)). In this case, allocations of packets have no impact on the delivery of packets. For example, in Fig. 2(a), packets are split as 10, 2 on $p_1$ and $p_3$, while in Fig. 2(b), packets are split as 6, 6 on the same paths. In both allocations, the path delays are the same. Consequently, the differential delay for this routing selection is $d(p_1) - d(p_2) = 2 - 2 = 0$, no matter how the packets are allocated among paths.

**B). Considering the Transmission delay**

If we consider a more practical scenario with the transmission delay on each node, then allocations of packets on multiple paths will have impact on path delays, as well as the differential delays. Similarly, we assume that 12 packets are to be transmitted to BS. Assume we use the same routing and resource allocations. Then, with the transmission delay, the path delay of $p_1$, $d(p_1)$, can be calculated as:

$$d(p_1) = \sum_{e \in p_1} d(e) + \sum_{v \in p_1} \tau(v) = 2 + (10 \text{ pk/s} + 10 \text{ pk/s}) = 12$$

Similarly, $d(p_3) = 2 + \frac{2 \text{ pk}}{3 \text{ pk/s}} = 2.5$ for transmitting 2 packets. The differential delay is 9.5 between these two paths.

Now in Fig. 2(b), use the same routes, but different packet allocation. Please note that without the consideration of the transmission delay, there would be no difference between this allocation and the previous one. Now considering transmission delay, $d(p_1) = 2 + \frac{6}{5} + \frac{6}{5} = 8$, and $d(p_3) = 2 + \frac{6}{5} = 3.5$, each with 6 packets. The differential delay has changed because of a different packet allocation, even using the same routes.

**C. Problem Statement**

It is worth noting that previous work on differential delay [2] [28] [29], or delay-bounded routing [22], did not consider the impact of transmission delay, and thus cannot be directly applied to solve the problem we are about to present.

**Definition 5 (DEAR Problem).** Given a graph $G = (V, E, b, d, w, \beta)$ with node set $V$ and link set $E$, where each communication link $e$ is associated with a bandwidth $b(e) > 0$ and a propagation delay $d(e) \geq 0$. Let $R$ be a new connection request with source node $s$, base station $\text{BS}$, packet request $Q$, adaptive reliability requirement $\alpha\%$, and differential delay requirements $d_{min}$ and $d_{max}$. The Delay-bounded Energy-constrained Adaptive Routing (DEAR) problem seeks a set of paths $P$ that can provide:

**Delay Bounded:**

- Any path $p$ in $P$ must satisfy the differential delay constraint: $d_{min} \leq d(p) \leq d_{max}$.
Energy Constrained:
- The energy consumption of transmitting packets for each sensor \( i \) cannot exceed its residual energy level \( \beta(i) \)
  \[
  E(i) = \sum_{j \in \text{adj}(i)} (q(i, j) \cdot w(i, j)) \leq \beta(i)
  \]
  where \( q(i, j) \) is the number of packets sent from node \( i \) and received by node \( j \), and \( \text{adj}(i) \) is the set of neighbors which can receive packets from node \( i \).

Adaptive reliability:
- The size of aggregated packet of all paths in \( P \) is no less than \( Q : q(P) \geq Q \).
- Route the data such that any single link failure does not affect more than \( x\% \) of the total packets.

Let us use the previous example in Fig. 2 to illustrate the DEAR problem. Assume that the traffic requirement is that any single link failure cannot affect more than 70% of the total traffic and delay should be between \( d_{\text{min}} = 2 \) and \( d_{\text{max}} = 5 \). Three solutions are listed in Figs. 2(c) - 2(d). A feasible solution is given in Fig. 2(c), where 12 packets are divided as 2, 2, 8 on \( p_1, p_2, p_3 \), respectively. It is easy to see that any single link failure can affect as much as 8 packets, which is 67% of the total traffic. Meanwhile, the delays of these paths are between 4 and 5, which satisfy the differential delay constraint. On the other hand, the solution in Fig. 2(d) uses the same paths. But due to different packet allocation, \( p_1 \) has delay 8, instead of 4. \( d(p_3) \) reduced from 4 to 3. The differential delay requirement (between 2 and 5) is violated, even with the same routes. Thus, this is an infeasible solution. Meanwhile, in the solution in Fig. 2(e), we assign 2 and 10 packets on two paths, \( p_1 \) and \( p_3 \). Their delays are 4 and 4.5, respectively. This solution satisfies the differential delay requirement. However, if \( p_1 \) fails, more than 70% will be dropped, and the reliability requirement is violated. Thus, this is also an infeasible solution.

III. PSEUDO-POLYNOMIAL SOLUTION FOR DEAR

It is proved that the closely-related problem PESO is NP-hard [1]. Moreover, the differential delay problem has also been proved to be NP-hard [2] [28] [29]. Therefore, we believe that our proposed problem DEAR is also NP-hard. To find a provably good solution for DEAR, first we look into a special case of the DEAR problem, where link delays and differential delay bounds are integer values.

Definition 6 (IDEAR Problem). Given graph \( G = (V, E, b, d, w, \beta) \) with node set \( V \) and link set \( E \), where each link \( e \) is associated with a bandwidth \( b(e) > 0 \) and a propagation delay \( d(e) \geq 0 \). Let \( R \) be a new connection request with source node \( s \), base station \( BS \), packet request \( Q \), reliability requirement \( x\% \), and differential delay requirements \( d_{\text{min}} \) and \( d_{\text{max}} \). The differential delay requirements \( (d_{\text{min}}, d_{\text{max}}) \) and the propagation delay \( d(e) \) of each communication link \( e \) are assumed to be integers. The Integer Delay-bounded Energy-constrained Adaptive Routing (IDEAR) problem seeks a set of paths \( P \) such that:

Delay Bounded:
- Any path \( p \) in \( P \) must satisfy the differential delay constraint: \( d_{\text{min}} \leq d(p) \leq d_{\text{max}} \).

Energy Constrained:
- The energy consumption of transmitting packets for each sensor \( i \) cannot exceed its residual battery level \( \beta(i) \)
  \[
  E(i) = \sum_{j \in \text{adj}(i)} (q(i, j) \cdot w(i, j)) \leq \beta(i)
  \]

Adaptive reliability:
- The size of aggregated packet of all paths in \( P \) is no less than \( Q : q(P) \geq Q \).
- Route the data such that any single link failure does not affect more than \( x\% \) of the total packets.

To solve IDEAR, we present a pseudo-polynomial time solution based on a novel graph transformation.

A. Graph Transformation

Let an instance of IDEAR given by graph \( G = (V, E, b, d, w, \beta) \), with reliability requirement \( x\% \), packet delivery request \( Q \), differential delay bounds \( d_{\text{min}} \) and \( d_{\text{max}} \), source-base station pair \((s, BS)\). We construct a layered directed graph \( G^R = (V^R, E^R) \) as follows.

1) For each node \( u \in V \), there exist \((d_{\text{max}} + 1)\) nodes \( u_0[u], u_1[u], \ldots, u_{d_{\text{max}}} \) in \( V^R \). Each \( u[t] \) means that node \( u \) can transmit packets at time \( t \).

2) For each link \( e = (u, v) \) in \( G \), \( E^R \) contains \( d_{\text{max}} - d(e) \) links in the form \((u[i], v[i+d(e)+1])\), where \( i = 0, 1, \ldots, (d_{\text{max}} - d(e) - 1) \). This means that if node \( u \) sends packets at time \( i \) using communication link \( (u, v) \), sensor node \( v \) will receive the packets at time \( i+d(e)+1 \), counting transmission delays. The bandwidth of each link \((u[i], v[i+d(e)+1])\) is set to be \( b(e) \). This bandwidth ensures that the packets sent by \( u \) at time \( i \) cannot exceed \( b(e) \).

3) For each node \( u \neq BS \), \( E^R \) contains \( d_{\text{max}} \) links in the form of \((u[i], u[i+1])\), \( i = 0, 1, \ldots, d_{\text{max}} - 1 \). The bandwidth of each link is set to be \( \infty \). Such link \((u[i], u[i+1])\) takes into account the transmission delay. For instance, if sensor node \( u \) received packets at time \( i \), and cannot send out all the packets at time \( i \) due to the constraint of
bandwidth \( b(u, v) \), then a flow will go through the link \((u[i], u[i+1])\) which means that \(u\) could send the residual packets at time \(i+1\).

4) For BS, \( E^R \) contains \( d_{max} - d_{min} \) links in the form of \((BS[i],BS[i+1])\), \(i = d_{min}, \ldots, d_{max} - 1\). Each link bandwidth is set to \(\infty\). This ensures that only the packets, which arrive at BS no earlier than \(d_{min}\) and no later than \(d_{max}\), will be calculated for the packet allocation.

Let us use an example in Fig. 1(b) for illustration. Given source \(A\) and BS, packet demand 12, reliability requirement \(x\% = 70\%\), differential delay \(d_{min} = 2\) and \(d_{max} = 5\), the corresponding layered graph \(G^R\) is represented in Fig. 3. For simplicity, not all the links are shown in Fig. 3. For each node in the original graph, we construct \(d_{max} + 1 = 5 + 1 = 6\) corresponding nodes in \(G^R\) shown in Fig. 3.

![Graph transformation technique](image)

Fig. 3. Graph transformation technique

For each node \(u\) which is not BS, we construct \(d_{max}\) links in the form of \((u[i], u[i+1]), i = 0, 1, \ldots, d_{max} - 1\). For node \(A\) in the original graph, the links \((A[0], A[1]), \ldots, (A[4], A[5])\) are constructed in the layered graph shown in Fig. 3. The bandwidth of each of these links are set to be \(\infty\). These links are used to handle the transmission delay of each sensor node. For BS, \(d_{max} - d_{min} = 3\) links: \((BS[0], BS[1]), (BS[1], BS[2]), (BS[2], BS[3]), (BS[3], BS[4])\), are constructed. All the bandwidths of these links are assigned as \(+\infty\). These are used to satisfy the differential delay requirements.

To find a solution for an instance of IDEAR from \(s\) to \(BS\) in \(G\), we search paths in \(G^R\) from \(s[0]\) to \(BS[d_{max}]\). Note that any path \(p\) from \(s[0]\) to \(BS[d_{max}]\), as well as its corresponding path \(p\) in \(G\), guarantees that its delay is between \(d_{min}\) and \(d_{max}\). Since the aggregated flow on links \((u[i], v)\), \(v = 0, \ldots, d_{max} - d(u, v) - 1\) is no more than \(x\% \cdot Q\), link \((u, v)\) has no more than \(x\% \cdot Q\) flow value on it in the corresponding \(s\)–BS flow in \(G\). In our example, given total 12 packets, and adaptive reliability at \(70\%\), the number of packets that can be allocated on each link in \(G\) is at most 8.4. Correspondingly, in Fig. 3, all links in the form \((u[i], v[i+d(u, v)+1]), i = 0, \ldots, (d_{max} - d(u, v) - 1)\), can carry aggregated packets no more than 8.4.

With the maximum flow (whose value \(\geq Q\)) calculated by IDEAR, we can use arc-chain decomposition [11], [12] to generate paths. Each path has its delay bounded between \(d_{min}\) and \(d_{max}\), and carries no more than \(x\% \cdot Q\) traffic. For our example in Fig. 1(b), Algorithm 1 will find a flow which is illustrated in Fig. 4. It will be decomposed to four paths (Line 4): \(p_1^R = (A[0], B[2], BS[4], BS[5])\) (marked by solid blue links), \(p_2^R = (A[0], BS[3], BS[5])\) (marked by dashed black links), \(p_3^R = (A[0], BS[3], BS[4], BS[5])\) (marked by solid red links) and \(p_4^R = (A[0], A[1], BS[2], BS[5])\) (marked by dashed red links), in the layered graph \(G^R\). It is worth noting that the flow value on link \((A[0], A[1])\), 4 represents that sensor \(A\) can not transmit all the packets at time 0 due to the transmission delay. The residual 4 packets could be sent out at time 1. In other words, link \((A[0], A[1])\) represents the transmission delay of node \(A\). The blue path has corresponding path \(p_1 = (A, B, BS)\) in \(G\) with packets 2 and path delay 4. The black dashed path has corresponding path \(p_2 = (A, C, BS)\) in \(G\) with packets 2 and path delay 5. The red paths have corresponding path \(p_3 = (A, BS)\) in \(G\) with packets 8 and path delay 4. The corresponding packet allocation result is shown in Fig. 2(c).

### B. Restricted Maximum Flow Scheme

A critical part of our solution for IDEAR is in Line 2 of Algorithm 1. To find the paths, we need to find a restricted maximum flow in \(G^R\). The restrictions of our flow solution are (1) the energy constraint on each node; (2) the aggregated packets on the corresponding links (in \(G^R\)) of a link in \(G\). To solve the restricted maximum network flow problem in the layered graph \(G^R\), we propose a Linear Program (LP) solution.

**Maximize** \(F\) \hspace{1cm} (3.1)

subject to:

\[
\sum_{u \in ad(s[0])} f(s[0], u) - \sum_{u \in ad(s[0])} f(u, s[0]) = F \hspace{1cm} (3.2)
\]

\[
\sum_{(x, y) \in E^R} f(x, y) = \sum_{(y, z) \in E^R} f(y, z) \forall y \neq s[0], BS[d_{max}] \hspace{1cm} (3.3)
\]

\[
d_{max} - d(x) - 1 \leq \sum_{x=0}^{d_{max} - d(x) - 1} f(u[i], v[i+d(u, v)+1]) \leq x\% \cdot Q, \forall x = (u, v) \hspace{1cm} (3.4)
\]

\[
\sum_{i=0}^{d_{max} - d(x) - 1} \sum_{v \in ad(u[i])} f(u[i], v) \cdot w(u, v) \leq \beta(u), \forall u \in V \hspace{1cm} (3.5)
\]

\[
0 \leq f(x, y) \leq b(x, y), \forall (x, y) \in E^R \hspace{1cm} (3.6)
\]
In our restricted maximum flow formulation, the objective is to find a maximum flow (allocated packets) in $G^R$. The variables are flow values (real valued) on the links in $G^R$. Constraints (3.2) and (3.3) ensure the flow conservation in the network. Constraint (3.4) ensures that for each link $(u, v)$ in $G$, the aggregated flow on all links $[u_{ij}, v_{i+1, j+1}] \in E^R$ is no more than $x\%$ of the total traffic (the reliability requirement). Constraint (3.5) ensures that for each sensor $u$, the energy consumption for transmitting packets is no more than the residual energy of $u$. These variables are nonnegative (ensured by Constraint 3.6). These constraints make restricted maximum flow different from the conventional network flow problem [12].

Our linear programming problem has $O(m \cdot d_{max})$ variables. Meanwhile, assume that $L$ is the input size of the instance of IDEAR($G, s, BS, Q, x, d_{min}, d_{max}$). The input size of our linear programming is of $d_{max}$ times the input size of IDEAR($G, s, BS, Q, x, d_{min}, d_{max}$), say $O(d_{max}L)$. Based on the theory in [35], it takes $O(m \cdot d_{max)^3(d_{max}L)}$ time to solve the LP formulation. This shows that our solution is a polynomial time scheme.

IV. APPROXIMATION SCHEME FOR ODEAR PROBLEM

The DREAL problem studied in Section II-C and III is a decision problem [9]. In this section, we first introduce its corresponding optimization problem, denoted as ODEAR. Because of the hardness of ODEAR, the best we can do is to present an approximation scheme for the ODEAR problem unless P = NP [14]. Then we will study a special case of the ODEAR problem (denoted as SPDEAR), and present an $(1+\alpha)$-approximation scheme for SPDEAR. Finally, we propose an $(1+\alpha)$-approximation algorithm for the ODEAR problem based on the approximation scheme for SPDEAR.

Definition 7 (ODEAR Problem). Given graph $G = (V, E, b, d, w, \beta)$ with node set $V$ and link set $E$, where each link $e \in E$ is associated with a bandwidth $b(e) > 0$ and a propagation delay $d(e) \geq 0$. Let $R$ be a new connection request with source node $s$, base station $BS$, packet delivery request $Q$, reliability requirement $x\%$, and delay requirements $d_{min}$. The optimization version of the DEAR problem ODEAR seeks a set of $s$–$BS$ paths, $P$, together with a feasible packet allocation $L$ such that:

Objective: Minimize $D$

Subject to:

1. $q(p) \geq Q$ (2) $d_{min} \leq d(p) \leq D, \ \forall p \in P$

2. $q(e) \leq x\% \cdot Q, \ \forall e \in E$ (4) $E(i) \leq \beta(i), \forall i \in V$

It also implies that ODEAR is NP-hard. Thus, the best we can do is to present an approximation scheme or algorithm for the ODEAR problem. We first study a special case of ODEAR (denoted by SPDEAR), where $d_{min} = 0$. A Fully Polynomial Time Approximation Scheme (FPTAS) is presented for this special case. Then we provide an $(1+\alpha)$-approximation algorithm for the ODEAR problem based on the FPTAS for the SPDEAR problem. Before introducing an FPTAS for the SPDEAR problem, we first give our observation on the relation between ODEAR and SPDEAR.

Lemma 1. If there is no feasible solution for an instance of SPDEAR($G, s, BS, x, Q$), there is no feasible solution for the instance ODEAR($G, s, BS, x, Q, d_{min}$).

Proof. Assume that SPDEAR($G, s, BS, x, Q$) does not have a feasible solution while ODEAR($G, s, BS, x, Q, d_{min}$) has a feasible solution path set $P$. Therefore, we have $d_{min} \leq d(p) \leq d(P)$ for each path $p \in P$ and $q(P) \geq Q$ with each link having no more than $x\%$ of the total packets on it. It is easy to see that this path set $P$ is also a feasible solution for SPDEAR($G, s, BS, x, Q$) because of $0 \leq d(p) \leq d(P)$ for each path $p \in P$. This will contradict our assumption. Therefore, if an instance of SPDEAR($G, s, BS, x, Q$) has no feasible solution, an instance of ODEAR($G, s, BS, x, Q, d_{min}$) must have no feasible solution too.

A. Fully Polynomial Time Approximation Scheme for SPDEAR

For a given positive real number $\theta$ and an instance SPDEAR($G, s, BS, x, Q$), we construct an auxiliary graph $G^\theta = (V, E, b, d^\theta, w, \beta)$ of $G = (V, E, b, d, w, \beta)$. The edge propagation delay is changed as $d^\theta = [d(e) \cdot \theta] + 1$ for each edge $e$. This is the scaling and rounding technique used in [20], [34].

Algorithm 2 FPTAS-SPDEAR($G, s, BS, x, Q$)

1. Construct $G^\theta = (V, E, s, d^\theta, w, \beta)$ by changing the edge delay $d^\theta = [d(e) \cdot \theta] + 1$ on each link $e$;
2. Find the minimum integer $D$, denoted as $D_m$, such that IDEAR($G^\theta, s, BS, x, 0, D$) is feasible;
3. Solve IDEAR($G^\theta, s, BS, x, 0, D_m$) and find the corresponding path set for the SPDEAR problem.

In Algorithm 2, we first update the propagation delay of each link in $G^\theta$ such that each link has an integer propagation delay. Next, we try to find the minimum integer $D$ which can return feasible solution for an instance of IDEAR with $d_{min} = 0$ and $d_{max} = D$. Denote the minimum value of $D$ as $D_m$, we try to solve an IDEAR instance with edge propagation delay $d^\theta$, $d_{min} = 0$ and $d_{max} = D_m$. For the returned path set $P$, we have the following theorem.

Theorem 1. The solution found by Algorithm 2 is an $(1+\alpha)$-approximation to SPDEAR($G, s, BS, x, Q$).

Proof. Assume that the optimal solution for SPDEAR is a set of paths denoted by $Q$. And the optimal value is $d_{o}$. For any path $p$ in the optimal solution $Q$, its delay value in $G^\theta$ is denoted as $d^\theta(p)$. Assuming that $b_{min}$ represents the minimum link bandwidth in the network, we have

$$d^\theta(p) = \sum_{e \in p} d^\theta(e) = \sum_{e \in p} ([\theta \cdot d(e)] + 1) + \sum_{n \in p} \tau(n)$$

$$\leq \sum_{e \in p} (\theta \cdot d(e)) + 1 + \sum_{n \in p} \frac{x\% \cdot Q}{b_{min}}$$

$$= \theta \cdot d(p) + (n - 1)(1 + \frac{x\% \cdot Q}{b_{min}})$$

Since $d_{o}$ is the optimal value for SPDEAR, and $p$ is a path from the optimal solution to SPDEAR, it is easy to see that
0 \leq d(p) \leq d_o. In other words, every path in the optimal solution has its delay no less than 0 and no more than \( d_o \). Let \( Z \) denote \( \sum_{u \in \mathcal{U}} \frac{U}{L \cdot \epsilon} \) which is the upper bound for transmission delay on any node, \( L \) and \( U \) are used to denote lower and upper bounds of \( d_o \) such that \( L \leq d_o \leq U \). For any given \( \epsilon > 0 \), we calculate \( \theta = \frac{d_o}{L \cdot \epsilon} \).

Combining with (4.1), we get that for each path \( p \):

\[
d^O(p) \leq \theta \cdot d_o + (n-1)(1+Z) = \frac{(n-1) \cdot d_o}{L \cdot \epsilon} + (n-1)(1+\frac{1}{\epsilon}) \quad p \in \mathcal{P}
\]

Since \( d^O(p) \) only takes integer values, it implies that

\[
d^O(p) \leq \left\lfloor \frac{(n-1) \cdot d_o}{L \cdot \epsilon} \right\rfloor + (n-1)(1+\frac{1}{\epsilon}), \quad \forall p \in \mathcal{P} \quad (4.3)
\]

This proves that the optimal solution \( \mathcal{Q} \) for SPDEAR \((G,s,BS,x,Q)\) is also a feasible solution for IDEAR \((G^O, s, BS, x, Q, 0, D)\).

In Line 2 in Algorithm 2, \( D_m \) is found as the minimum value of \( D \) which guarantees that IDEAR\((G^O, s, BS, x, Q, 0, D)\) has a feasible solution. Since IDEAR\((G^O, s, BS, x, Q, 0, \lfloor \frac{(n-1) \cdot d_o}{L \cdot \epsilon} \rfloor + (n-1)(1+\frac{1}{\epsilon}) \) has a feasible solution \( \mathcal{Q} \), it is clear that \( D_m \leq \left\lfloor \frac{(n-1) \cdot d_o}{L \cdot \epsilon} \right\rfloor + (n-1)(1+\frac{1}{\epsilon}) \). This implies that

\[
D_m \leq \left\lfloor \frac{(n-1) \cdot d_o}{L \cdot \epsilon} \right\rfloor + (n-1)(1+\frac{1}{\epsilon}) \leq \theta \cdot d_o + L \cdot \epsilon \cdot \theta \cdot (1+\frac{1}{\epsilon}) \quad (4.4)
\]

In Line 3 of Algorithm 2, we solve IDEAR\((G^O, s, BS, x, 0, D_m)\). Let us assume that the returned feasible solution is \( \mathcal{P}^\theta \). For each \( p \in \mathcal{P}^\theta \), we have

\[
d^O(p) \leq D_m \leq \theta \cdot d_o + L \cdot \epsilon \cdot \theta \cdot (1+\frac{1}{\epsilon}) \quad (4.5)
\]

Meanwhile, we know that:

\[
d^O(p) = \sum_{e \in p} (\theta \cdot d(e)) + 1 + \sum_{\tau(n) \geq \tau(p)} \theta \cdot d(p) + \sum_{\tau(n) > \tau(p)} \theta \cdot d(p) \geq 0
\]

Combine (4.6) with (4.5), it is clear that

\[
\theta \cdot d(p) \leq d^O(p) \leq D_m \leq \theta \cdot d_o + L \cdot \epsilon \cdot \theta \cdot (1+\frac{1}{\epsilon}) \Rightarrow d(p) \leq d_o + L \cdot \epsilon \cdot (1+\frac{1}{\epsilon}) \leq 1 + \epsilon \cdot (1+\frac{1}{\epsilon}) \cdot d_o = (1 + \alpha) \cdot d_o, \quad \forall p \in \mathcal{P}^\theta \quad (4.8)
\]

Meanwhile, it is obvious that the reliability and energy requirement will be satisfied for a SPDEAR instance if it is satisfied for the corresponding IDEAR instance. This proves that \( \mathcal{P}^\theta \) returned by IDEAR\((G^O, s, BS, x, 0, D_m)\) is an \((1+\alpha)\)-approximation to SPDEAR\((G, s, BS, x, Q)\).

B. Discussion

The most important part of the FPTAS to SPDEAR is how to find \( D_m \) in Line 2 of Algorithm 2. In equation (4.4), we proved that \( D_m \leq \left\lfloor \frac{(n-1) \cdot U}{L \cdot \epsilon} \right\rfloor + (n-1)(1+\frac{1}{\epsilon}) \). Therefore, we know:

\[
0 \leq D_m \leq \left\lfloor \frac{(n-1) \cdot U}{L \cdot \epsilon} \right\rfloor + (n-1)(1+\frac{1}{\epsilon})
\]

To calculate \( D_m \), we first need to find a pair of lower and upper bounds of \( d_o \), denoted by \( L \) and \( U \). Then, using binary search, we can find \( D_m \) by testing \( \log(\frac{U}{d_o}) \) instances of IDEAR in polynomial time. Scaling and testing technique [20] [34] is adopted to calculated \( L \) and \( U \). Solving each instance takes \( O((m \cdot \frac{n \cdot U}{L})^{3/2} \cdot \frac{n \cdot U}{L}) \) time (proved in Section III-B) [20] [34].

C. Approximation Algorithm for ODEAR

In Section IV-A, we presented an \((1+\alpha)\)-approximation scheme to SPDEAR, a special ODEAR problem with \( d_{\text{min}} = 0 \). In this section, we will propose an \((1+\alpha)\)-approximation algorithm to solve the ODEAR problem with \( d_{\text{min}} \geq 0 \).

Algorithm 3 Approximation to ODEAR\((G, s, BS, x, Q, d_{\text{min}})\)

1: Solve SPDEAR\((G, s, BS, x, Q)\);
2: if (NO feasible solution is found) then
3: Drop the connection request.
4: end if
5: Assume the found path set is \( \mathcal{P} \);
6: for each path \( p \in \mathcal{P} \) do
7: if \( d(p) < d_{\text{min}} \) then
8: Remove \( p \) from \( \mathcal{P} \); q(\( \mathcal{P} \)) = q(\( \mathcal{P} \)) - q(\( p \));
9: end if
10: end for
11: if q(\( \mathcal{P} \)) \geq Q then
12: Output \( \mathcal{P} \) as the solution.
13: else
14: Drop the request.
15: end if

Our approximation algorithm, Algorithm 3, is based on the FPTAS for SPDEAR. Given an instance of ODEAR\((G,s,BS,x,Q,d_{\text{min}})\), we first solve an instance of SPDEAR\((G,s,BS,x,Q)\) (Line 1). This is a necessary condition check for the ODEAR problem. As we proved in Lemma 1, we can drop an instance of ODEAR if its corresponding SPDEAR instance is not feasible. Then if there is a feasible solution \( \mathcal{P} \) for SPDEAR\((G,s,BS,x,Q)\), we check every path \( p \) in \( \mathcal{P} \) (Lines 6-11). If \( p \) does not satisfy the differential delay constraint, we remove this path from \( \mathcal{P} \), and reduce the provided aggregated packets (Line 9). If all the remaining paths, which all satisfy the differential delay constraint, can still provide enough packets (no less than \( Q \)), we return \( \mathcal{P} \) as the solution. Otherwise, we drop the request.

Theorem 2. Path set \( \mathcal{P} \) found by Algorithm 3 is an \((1+\alpha)\)-approximation solution for ODEAR\((G,s,BS,x,Q,d_{\text{min}})\). \( \square \)

Proof. It is obvious that reliability and energy consumption requirements will be satisfied for the returned paths. Meanwhile, it is ensured that each path has its delay no less than \( d_{\text{min}} \) (for-loop from Line 6 to Line 9). Assume that the optimal value of the instance ODEAR\((G,s,BS,x,Q,d_{\text{min}})\) is \( d_o \). What we need to prove is that \( d(p) \leq (1 + \alpha) \cdot d_o, \forall p \in \mathcal{P} \).

Since \( \mathcal{P} \) is a feasible solution to SPDEAR, it has been proven in Theorem 1 that

\[
0 \leq d(p) \leq (1 + \alpha) \cdot d_o, \forall p \in \mathcal{P} \quad (4.9)
\]

where \( d_o^S \) is the optimal value of SPDEAR\((G,s,BS,x,Q)\).

From Lemma 1, we know that any feasible solution for ODEAR\((G,s,BS,x,Q,d_{\text{min}})\) is a feasible solution for SPDEAR\((G,s,BS,x,Q)\). Therefore, \( d_o \geq d_o^S \). Otherwise (if \( d_o < d_o^S \)), \( d_o^S \) cannot be the optimal value of...
SPDEAR\((G, s, BS, x, Q, d_{min}, d_{max})\). Then it is easy to see that \(d(p) \leq (1+\alpha)d_e, \forall p \in \mathcal{P}\). Therefore, the solution found by Algorithm 3 is an \((1+\alpha)\)-approximation to the ODEAR problem.

V. EFFICIENT HEURISTIC FOR DEAR

We have presented an FPTAS for SPDEAR and an approximation algorithm for ODEAR. Though both are polynomial time solutions, they still can be time consuming in large-sized network due to the scaling and testing process [34]. In practice, it is still necessary to provide fast and efficient heuristics for DEAR problem.

Algorithm 4 H-DEAR\((G, s, BS, x, Q, d_{min}, d_{max})\)

1: for each link \(e \in G\) do
2: Change the propagation delay \(d(e)\) as \([d(e)]\);
3: end for
4: \(d_{min}^R = [d_{min}]; d_{max}^R = [d_{max}];\)
5: Construct a layered graph \(G^R\) based on the graph transformation method in Section III-A;
6: Find a restricted maximum flow \(\mathcal{F}\) from \(s[0]\) to \(BS\) \([d_{min}^R, d_{max}^R]\) by solving the LP formulation in Section III-B;
7: if the flow value \(|\mathcal{F}| \geq Q\) then
8: Based on the flow values on all the links in \(G^R\), find the corresponding path set \(\mathcal{P}^R\);
9: Find the corresponding path set \(\mathcal{P}\) in \(G\);
10: return \(\mathcal{P}\);
11: else
12: Drop the request.
13: end if

A heuristic algorithm is listed in Algorithm 4. In this algorithm, we first round the propagation delay of each link to integer by taking ceiling function. Similarly, we round \(d_{min}\) to an integer using ceiling function, and round \(d_{max}\) to an integer using floor function. This ensures that the solution found by this algorithm is a feasible solution for the DEAR problem. Then, a restricted maximum flow \(\mathcal{F}\) on \(G^R\) is calculated by solving the LP formulation. If the flow value \(|\mathcal{F}|\) is no less than the packet demand \(Q\), the corresponding path set is output as the result. Otherwise, we drop this connection request.

VI. NUMERICAL RESULTS

In this section, we presented numerical results to evaluate the performances of our solutions. We implemented the Approximation Algorithm 3 and Heuristic Algorithm 4, which were denoted as DEAR and H-DEAR in the figures. All our simulation tests were performed on a 2.8 GHz Linux PC with 2G bytes of memory. We used different network topologies in an 100 \times 100 sq. units playing field to evaluate our solutions. All the sensor nodes were randomly distributed in the playing field. The power of each sensor node was randomly distributed in \([16, 20]\). The bandwidth, propagation delay, and transmission energy consumption of each communication link was randomly distributed in \([6, 10], [1, 5],\) and \([1, 3]\).

First, we aim to test the impact of the number of sensors on the performance of our solutions. As Figs. 5(a) and 5(b) showed, Scenario 1 and 2 were designed to compare the performances in terms of delay ratio and running time, where delay ratio is defined as \(\frac{d(P)}{d_{max}}\). If there is no feasible solution, then this delay ratio is defined as \(\infty\). In these two scenarios, the reliability requirement \(x\%) and the packet delivery request \(Q\) were set to 50\% and 5, respectively. The source node and the base station were randomly chosen from all the sensors. We chose 0.5 and 1 as the values of \(\epsilon\) to test the solutions of DEAR. Gurobi Optimizer [15] was used to solve the LP formulation in Section III-B. Scenario 1 illustrated that as the number of nodes increased, the delay ratio also increased. When the number of nodes were 5 and 10, the delay ratio performances of DEAR \((\epsilon = 1)\), DEAR \((\epsilon = 0.5)\) and H-DEAR were close to each other. When the number of nodes was greater than 10, DEAR \((\epsilon = 0.5)\) could not provide a solution due to the memory limitation. Similarly, DEAR \((\epsilon = 1)\) could not find solutions when the number of nodes was greater than 20. Based on above observations, H-DEAR is more suitable for large network sizes.

Next, we tested the impact of the reliability requirement on the performances of our solutions. In these two scenarios, the packet delivery request was set to 5, and the number of sensors was 10. In Fig. 6(a), Scenario 3 showed that the delay ratio performances of DEAR \((\epsilon = 1)\), DEAR \((\epsilon = 0.5)\) and H-DEAR were close to each other. As the reliability requirement \(x\%) increased, the delay ratio also increased. From the simulation results, we observed that the reliability requirement not only has impact on the packets loss of link failure (as we expected), but also affected the delay from the source node to the base station. In Fig. 6(b), the running time of H-DEAR was much less than the ones of DEAR \((\epsilon = 1)\) and DEAR \((\epsilon = 0.5)\). Another observation is that for the performance of DEAR, the greater the value of \(\epsilon\), the less the running time, which confirms our theoretical analysis in Section IV.

Finally, we tested the impact of the packet demand on the performances in terms of delay ratio and running time. The packet demand was increased from 1 to 5 in these two scenarios. In Scenarios 5 and 6, we set the reliability requirement.
requirement as 50%. Scenario 5 in Fig. 7(a) shows that as the packet requirement increased, the average delay ratio was also increased. On the other hand, we find that the delay ratio of DEAR (ε = 1), DEAR (ε = 0.5) and H-DEAR were close to each other. Fig. 7(b) shows the running times of DEAR (ε = 1), DEAR (ε = 0.5) and H-DEAR. We observed that the running time performances of DEAR (ε = 1), DEAR (ε = 0.5) and H-DEAR had similar trends with the ones in Scenario 4.

VII. CONCLUSIONS

In this work, we studied the Delay-bounded Energy-constrained Adaptive Routing (DEAR) problem. This paper is the first work which jointly studies the adaptive multipath routing, the energy constraint, and the differential delay problem. Moreover, this work considers and analyzes the impact of transmission delay in multipath routing, which is largely ignored in most of the previous works. We proposed a novel pseudo-polynomial time algorithm for a special case of DEAR where edge delays and differential delay bounds are integers. Next, we presented a fully polynomial time approximation scheme for the SPDEAR problem and an (1 + ϵ)-approximation algorithm for the ODEAR problem, the first to the DEAR problem. We also provided an efficient heuristic for the DEAR problem. Numerical results confirm the advantage of our solutions.

REFERENCES