Optimal Data Caching and Forwarding in Industrial IoT With Diverse Connectivity

Zenghua Zhao, *Member, IEEE*, Yiyao Shi, Bingxue Diao, and Bin Wu

***Abstract*—Many real-world wireless networks for industrial internet of things applications have diverse connectivity characteristics, which makes routing protocol design challenging. Although adaptive routing protocols have been emerging to deal with connectivity diversity, there is still lack of a unified routing framework in well-connected and intermittently connected networks. In this paper, we present a unified routing metric for wireless networks with diverse connectivity, and formulate the adaptive routing problem as an integer linear programming optimization one. We then propose a heuristic routing protocol, caching-optimized adaptive routing protocol (COARP). Extensive simulation results show that COARP is near optimal in simple static network scenarios, and performs well across all the range of connectivity spectrum in dynamic network scenarios.**

***Index Terms*—Diverse connectivity, industrial IoT, routing protocol, wireless networks.**

I. Introduction

INDUSTRIAL internet-of-things (IIoT) is emerging recent years, which incorporates communication techniques, fog computing, and cloud computing into industrial applications [1], [2]. Wireless networks are primary communication techniques to connect devices in IIoT. However, many real-world wireless networks in IIoT applications have temporal or spatial diversity in connectivity characteristics. In other words, the underlying network topologies vary dynamically from well connected to intermittently connected due to node mobility. For example, the DieselNet-Hybrid bus network's connectivity varies spatially [3]. Connectivity diversity is also observed in other IIoT applications, such as intelligent transportation using vehicular ad hoc networks [4] and farmland monitoring using micro aerial vehicle networks (MAVs) [5]. With the proliferation of IIoT applications, we believe more and more wireless networks underlying them will exhibit connectivity diversity.

The network connectivity diversity makes routing protocol design challenging. There are two primary challenges. First, it is nontrivial to propose a unified routing metric characterizing various network connectivity. In well-connected wireless networks, routing protocols assume there always exist end-to-end paths between source and destination. Thus, they usually adopt routing metrics characterizing end-to-end delay or the similar. For example, expected transmission time (ETT) [6], a well-known routing metric used in well-connected networks, predicts the total amount of time it would take to send a packet along a route. However, ETT-like routing metrics cannot be used in intermittently connected networks, since a persistent end-to-end path is unavailable in such networks. Two mobile nodes contact each other when their distance is less than their transmission range. The inter-contact time (time interval between two consecutive contacts) of two nodes is then leveraged to make routing decisions in intermittently connected networks [7]. Similarly, inter-contact time is meaningless in well-connected networks. Therefore, routing metrics used in well-connected networks are ill-suited in intermittently connected networks, and vice versa. A unified routing metric is still open for industrial wireless networks with diverse connectivity.

Second, it is challenging to unify packet forwarding schemes in diverse connectivity networks. In well-connected networks, packets are forwarded as fast as possible in a *store-and-forward* way. Whereas in intermittently connected networks, packets are forwarded in a *store-carry-and-forward* way. Packets have to be carried/cached at intermediate nodes during network disruption, and wait for opportunities to be forwarded later. To address this problem, R3 [3] identifies packet replication as a key structural difference between protocols designed for opposite ends of the connectivity spectrum. Packet replication levels are adapted to the extent of uncertainty in network path delays. However, when and where to cache packets is unconsidered in [3].

To address the above challenges, we first propose a unified routing metric considering both delay and caching cost. The unified routing metric consists of two parts: hybrid link delay and node caching cost. Hybrid link delay fuses link delivery delay used in well-connected networks with link disruption delay in intermittently connected networks. Link delivery delay denotes time to deliver a packet across a link. Whereas link disruption delay describes the expected inter-contact time. Furthermore, node caching cost captures the feature of caching packets at intermediate nodes during network disruption.

Manuscript received May 18, 2018; revised August 4, 2018 and October 20, 2018; accepted November 30, 2018. Date of publication December 27, 2018; date of current version April 3, 2019. This work was supported in part by the Key project from the National Natural Science Foundation of China under Grant 61832013, and in part by the Natural Science Foundation of Tianjin under Grant 17JCYBJC15500. Paper no. TII-18-1250. *(Corresponding author: Zenghua Zhao.)*

Z. Zhao, Y. Shi, and B. Wu are with the Tianjin Key Laboratory of Advanced Networking, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: zenghua@tju.edu.cn; shiyiyao@tju.edu.cn; binw@tju.edu.cn).

B. Diao was with Tianjin University, Tianjin 300350, China. She is now with the Beijing Jingdong Century Trading Co., Ltd., Beijing 100176, China (e-mail: bingxue_diao@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2018.2889880

Based on this unified routing metric, we then formulate the routing problem in diverse connectivity industrial wireless networks into an integer linear programming (ILP) optimization one. The rationale behind the formulation is that when the network is well connected, the routing protocol selects the path with the minimum end-to-end delay. While the network is in disruption, nodes along the path with minimum expected inter-contact delay and caching cost are selected to cache packets. Since ILP solving is time consuming, we further design a heuristic routing protocol, i.e., caching-optimized adaptive routing protocol (COARP). COARP estimates the unified routing metric on-line and can be run in real time.

We evaluate COARP in simulations via NS-2 [8]. We first compare the performance of COARP with the ILP-based optimization solution in three simple but typical network scenarios: well connected, partially connected, and intermittently connected one. We then verify COARP in more complicated network scenarios varying from well connected to intermittently connected dynamically. The performance of COARP is compared with that of three representative routing protocols: probabilistic routing protocol using history of encounters and transitivity (PROPHET) [9], dynamic source routing (DSR) [10], and delay tolerant transport protocol (DTTP) [11].

In summary, the main contributions of this paper are the following.

(1) We propose a unified routing metric to deal with connectivity diversity, and formulate the adaptive routing problem as an ILP optimization one. To the best of our knowledge, this is the first work to formulate routing problem for industrial wireless networks with diverse connectivity.

(2) We design a heuristic routing protocol, COARP, to tackle with the computational complexity in solving an ILP problem.

(3) We implement and evaluate COARP in extensive simulations via NS-2. In simple network scenarios, COARP is near optimal compared to the ILP-based optimal solutions. In complicated scenarios, COARP performs well across all the connectivity spectrum, compared with other routing protocols.

The rest of the paper is organized as follows. In Section II, we briefly introduce the state-of-the-art routing protocols adaptive to diverse connectivity. We present the unified routing metric in Section III and formulate the adaptive routing problem as ILP optimization one in Section IV. Section V illustrates the design of COARP. We evaluate COARP comprehensively in Section VI. Section VII concludes the paper.

II. Background and Related Work

A. Background

The IIoT is a key enabling technology to realize the vision of Industry 4.0 [1]. Many types of wireless networks and standards serve IIoT as device connectivity solutions, such as 802.11, ZigBee, WirelessHART, and ISA100.11a [12]. 802.11 and ZigBee have been widely applied in various industrial applications, including smart transportation, environmental monitoring, healthcare service, and surveillance. Meanwhile, WirelessHART and ISA100.11a are developed for industrial automation and process control in harsh environments. Although they are applied in different industrial applications, from technique perspective, all the above wireless networks support multihop mesh network topology. In this case, routing protocol plays an important role for overall network performance.

In this paper, we focus on routing protocol design for industrial wireless networks with diverse connectivity due to node mobility. Although our evaluation is performed based on 802.11, we design COARP without any assumptions on under layer wireless techniques. Our research is motivated by two factors. The first is that most of routing protocols are designed for either well-connected or intermittently connected networks, and work poorly or even fail to work in networks with diverse connectivities [3]. The second is that more and more industrial wireless networks exhibit connectivity diversity characteristics driven by their applications, especially with the usage of mobile robots in industries. We will give some examples here.

The DieselNet-Hybrid bus network is among the first to be found that it's connectivity varies spatially [3]. Buses are well connected in urban centers with high Wi-Fi access point (AP) density, but are poorly connected as they move to less urban areas. Connectivity diversity is also found in other industrial applications, such as intelligent transportation [4] and farm land monitoring [5]. Currently, some industrial wireless standards have been designed for networks with fixed devices (e.g., WirelessHART), while a major future demand for wireless industrial systems could be the capability to support mobility with the usage of mobile robots [13]. With the proliferation of IIoT applications, we believe that more and more underlying wireless networks will exhibit diverse connectivity.

B. Related Work

There are huge bodies of routing protocols for each type of wireless networks. We consider wireless networks with diverse connectivity characteristics, and thus introduce only routing protocols adaptive to various network connectivity. We broadly classify such routing protocols into two categories, hybrid routing and adaptive routing. Hybrid routing protocol combines two existing routing protocols together, whereas adaptive routing protocol is a single protocol designed for diverse connectivity environments.

Hybrid Routing Protocol: Since it is hard to find a unified routing metric to characterize various connectivity conditions, many researchers integrate existing routing protocols in well-connected networks and that in intermittently connected networks into one framework, making a switch between these two routing protocols according to connectivity conditions [11], [14]–[20]. For example, [11] integrates optimized link state routing protocol (OLSR) [21] and PROPHET [9]. It is a straightforward yet efficient way to use a hybrid routing protocol in wireless networks with diverse connectivity. However, it is hard to decide the switching point, which hurts the network performance. Moreover, maintaining two routing protocols results in more overhead than using a single routing protocol. Therefore, COARP is designed to be a single unified routing protocol rather than a hybrid routing protocol.

Adaptive Routing Protocol: Adaptive routing protocol is a unified routing protocol adaptive to various network connectivity. It is an appealing solution to wireless networks with diverse connectivity for its efficiency and concise compared with hybrid routing. Tie *et al.* [3] propose an adaptive routing protocol R3. R3 dynamically replicates packets according to the network condition. It collects path delay distributions by calculating the convolutions of individual link delay distributions obtained through probings. The shortest path in terms of expected path delay is first to select and then the second path which minimizes the combined two-path delay. To some extent, it is a multipath source routing protocol. Yang *et al.* [22] improve R3 by choosing the number of packet replications dynamically, and takes delay correlations into account for better decisions.

Asadpour *et al.* [5] design a location-aware DTN/geo-routing algorithm for MAVs, which experience various connectivity. The algorithm basically routes a packet along the spatially shortest path to the destination, if an end-to-end path exists. Otherwise the packet is physically carried closer to the destination. The algorithm makes use of location information of mobile nodes available in MAVs. Since location information is not pervasive in most IIoTs, we assume it is unavailable in our work.

Our work is inspired by the above routing protocols. The primary difference from them is that we propose a unified routing metric to deal with the connectivity diversity, and build an ILP model.

III. Network Model and Unified Routing Metric

A. Network Model

We consider a wireless network with diverse connectivity due to node mobility. The network is multihop and is composed of mobile nodes. Two nodes transfer data packets to each other when within communication range. A node can be a source to generate packets or a destination to receive packets from others. Every node is willing to forward packets as an intermediate one. The network is assumed to have high node density as in most IIoT application scenarios [1]. Therefore, the network is well connected most of the time, and is broken into one or several partitions intermittently due to node mobility. That is to say, the network has diverse connectivity characteristics. The network topology varies dynamically but in a slow manner. Nodes have finite storage capacity. We also assume the storage is exclusively used for holding in-transit data as in [7]. The storage for holding data consumed by applications is sufficient and not considered here.

Formally, we denote such a wireless network as a graph $G(V, E)$, where V is the set of network nodes, and E is the set of links. The graph is bidirectional, i.e., if $ij \in E$, then $ji \in E, \forall i, j \in V$. If node i and node j can hear each other, then $ij \in E$, and $ji \in E$.

B. Unified Routing Metric

The goal of a routing algorithm in diverse-connectivity networks is to work efficiently across various connectivity scenarios. That is to say, it should achieve minimum delay during full connectivity by forwarding packets as fast as possible, and achieve minimum loss rate during intermittent connectivity by caching packets at intermediate nodes. To this end, we define a unified metric to meet requirements in various connectivity status.

Hybrid Link Delay: Link delay is usually used in well-connected networks as a basic component of routing metric, as well as in intermittent-connected networks [7]. However, the meanings of link delays in the two connectivity scenarios are different from each other[3]. In the former case, link delay refers to the time to deliver a packet across a link, denoted as *link delivery delay* here. In the latter case, since there is no persistent path between a source and a destination, link delay instead indicates the expected delay of inter-contact time of two nodes in the network, denoted as *link disruption delay*. To unify the delay expression, we define *hybrid link delay* as the sum of link delivery delay and link disruption delay.

Caching Cost: Caching capability is also an important metric, especially in intermittently connected networks. While a link is broken, packets should be cached at intermediate nodes and forwarded later. Which nodes to cache packets impact the network performance. For simplicity, we define *caching cost* of a node to be inversely proportional to its available buffer size. This is reasonable, since the candidate node should have enough buffer to cache the packets.

Unified Routing Metric Definition: Based on the above analysis, we define the unified routing metric as the weighted sum of hybrid path delay and the caching cost of nodes along the path. We denote the unified routing metric on a path p as C_p. C_p can be written as

$$\begin{aligned} C_p &= \alpha C_p^D + \beta C_p^B \\ &= \alpha \sum_{ij \in E_p} C_{ij}^D + \beta \sum_{k \in V_p} C_k^B \end{aligned} \tag{1}$$

where, C_p^D is the sum of hybrid link delay of links along the path p. C_p^B denotes the caching cost of nodes along path p, if packets need to be cached in the network. α and β are weights of C_p^D and C_p^B, respectively, $\alpha + \beta = 1$. E_p and V_p denote the link set and the caching node set of path p, respectively. In addition, since delay and caching cost have different units, they are normalized before use.

IV. ILP Formulation

Based on the unified routing cost, we present an ILP formulation that determines the optimal routing for minimizing routing cost in the network. Notations and variables used in formulating the ILP are as follows.

Notations

s	Source node.
d	Destination node.
d_{ij}	Hybrid link delay of link ij.
T_d	Link delay threshold.
B_k	Available buffer size in node k.
F	Total packets need to be cached in the network.

C_k^B Cost of caching packets in node k.
C_{ij}^D Hybrid delay of link ij.
T A predefined constant.
α, β Weights.

Variables

e_{ij} Binary integer variable. It takes 1 if link ij on the path and 0 otherwise.
b_k Binary integer variable. It takes 1 if b_k caches packets and 0 otherwise.

We express ILP formulation as follows.

$$\text{P: Minimize}\left(\alpha \sum_{ij\in E} e_{ij} C_{ij}^D + \beta \sum_{k\in V} b_k C_k^B\right). \tag{2}$$

s.t.

$$\sum_{j\in V} e_{sj} = 1, sj \in E \tag{3}$$

$$\sum_{j\in V} e_{js} = 0, js \in E \tag{4}$$

$$\sum_{j\in V} e_{dj} = 0, dj \in E \tag{5}$$

$$\sum_{j\in V} e_{jd} = 1, jd \in E \tag{6}$$

$$\sum_{j\in V} e_{ij} \leq 1 \quad \forall i \in V, ij \in E, i \neq s, i \neq d \tag{7}$$

$$\sum_{j\in V} e_{ji} \leq 1 \quad \forall i \in V, ij \in E, i \neq s, i \neq d \tag{8}$$

$$\sum_{j\in V} e_{ji} = \sum_{k\in V} e_{ik} \quad \forall i \in V,\ ji \in E, ik \in E, \tag{9}$$

$$i \neq s, i \neq d \tag{10}$$

$$b_k = \prod_{i\in V} \left\lfloor \frac{d_{ki}}{T_d} \right\rfloor \quad \forall ki \in E \tag{11}$$

$$b_k \leq \sum_{i\in V} e_{ki} \quad \forall ki \in E \tag{12}$$

$$\sum_{k\in V} b_k \times B_k \geq F. \tag{13}$$

The objective function (2) is to minimize the routing cost. The first term is the delay cost of all hybrid link delays along a path. The second is the caching cost of nodes which are on the path and cache packets.

Constraints (3)–(10) adhere to network flow conservations [23]. Constraints (3) and (4) define a source node s with one link from this node. Constraints (5) and (6) indicate d is a destination node with one link to this node. Here, we consider a unicast single-path routing: one source and one destination with only one path between them.

Constraints (7)–(10) limit an intermediate node i. For an intermediate node, (7) and (8) denote that there are at most one link in and at most one link out, respectively. Constraint (10) requires if there is one (or no) link in to node i, then there must be one (or no) link out of it.

We use b_k to select nodes which cache packets when the network is in disruption. If b_k is 1, then node k is enabled to cache packets. Node k caches packets, only if all of its out links are broken. In this case, hybrid link delay of every out link d_{ki} is larger than a delay threshold T_d. T_d is set to a constant value in time unit. For simplicity, we let $T_d < 2 \times \text{maximum}\{d_{ij}\}$, thus $\lfloor \frac{d_{ki}}{T_d} \rfloor$ is 0 or 1. Therefore, (11) meets the above condition. Constraint (12) requires that if node k caches packets, then k should be on the path. Constraint (13) ensures that total caching capacity of selected nodes is enough to cache all the packets in the network.

The caching cost of node k is inversely proportional to its available buffer size B_k, i.e., the more the available buffer size, the lower the cost. On the other hand, the caching cost is proportional to the size of packets buffered. We normalize it as

$$C_k{}^B = \frac{F}{B_k + F}, k \in V. \tag{14}$$

We also normalize the hybrid link delay of d_{ij} by dividing a large constant T. $T = \text{maximum}\{d_{ij}\}$. Thus, the cost of hybrid link delay $C_{ij}{}^D$ is

$$C_{ij}{}^D = \frac{d_{ij}}{T}, ij \in E. \tag{15}$$

V. COARP: A Heuristic Protocol

The ILP formulation can give an optimal solution. However, it is hard to solve an ILP problem in real time in large-scale networks due to its complexity and centralized nature. We thus design COARP, a heuristic routing protocol based on the insights gained from the ILP formulation. COARP runs at each node in real time, and is feasible in practical network settings.

A. COARP Overview

COARP is an adaptive routing protocol in wireless networks with dynamic connectivity. It adopts the unified routing metric to determine the optimal path as in the ILP formulation. The unified routing cost is calculated by estimating hybrid link delay and node caching cost periodically. Then COARP runs Bellman–Ford shortest path algorithm to select the path with minimum routing cost. In order to improve network performance further, we also design a packet scheduling scheme to manage packets cached at a node.

In a nutshell, COARP consists of three main components: first, route cost estimation, second, path selection, and third, packet scheduling. Each component is illustrated as follows.

B. Route Cost Estimation

According to the ILP formulation in (2), in order to calculate the route cost, we have to estimate the hybrid link delay and the node caching cost.

1) Hybrid Link Delay Estimation: Hybrid link delay consists of two parts: link delivery delay and link disruption delay. COARP estimates both link delivery delay and link disruption delay by probings. Each node sends probing packets to its one-hop neighbors periodically. When a neighbor node receives a probing packet, it feeds back an acknowledgment packet immediately. Link delivery delay is estimated as half of the round-trip time of a probing packet. If a probing packet is not acknowledged, the inter-contact time is the time interval between two consecutive acknowledged probing packets' arrival time. Link disruption delay is then derived by exponential moving average of the inter-contact time. Hybrid link delay is calculated by summarizing link delivery delay and link disruption delay.

2) Caching Cost Estimation: Each node estimates its caching cost according to (14) dynamically. When a node fails to deliver a packet to the next hop, it caches the packet, and updates its caching cost correspondingly.

C. Path Selection

Each node computes the path to each destination by running Bellman–Ford shortest path algorithm. In particular, when the hybrid link delay or the caching cost is updated, each node calculates the unified route costs between itself and all other nodes in the network based on the hybrid link delay and the caching cost. This routing information is stored as a table. When the routing cost changes, the routing information table is then broadcasted to its neighbor nodes. When a node receives the routing information table from its neighbors, it calculates the shortest paths to all other nodes and updates its own routing information table to reflect any changes.

D. Packet Scheduling in the Cache

As the buffer size is limited in a node, it is possible that the buffer overflows when too much packets need to be cached. In this case, which packet should be dropped first? Similarly, which packet should be forwarded first when there is a forwarding opportunity? We design a packet scheduling scheme to manage packets in the cache.

COARP schedules packets according to their utilities. The utility of a packet i is defined as $U_i = -D(i)$, where $D(i)$ is the average delay of the packet [24]. The average delay of a packet is the sum of hybrid link delays of links on the path from the source to the destination. Packets sharing the same source and destination have the same utilities. Therefore, COARP schedules packets according to a tuple (source, destination, and utility). When the buffer overflows, the packets with the lowest utility are dropped first. Similarly, the packets with the highest utility are forwarded first when there are forwarding opportunities. The packets with the same utility are scheduled according to first-in-first-out rule.

E. Implementation

We implement COARP in NS-2. As a routing protocol, COARP is implemented at network layer and maintains the routing table. Each node broadcasts probing packets periodically to estimate routing cost. The period time is set to 0.5 s in our simulations. The probing packets are only sent to the one-hop neighbors to avoid broadcast storm. We use a double linked list to manage packets stored at a node.

Since Bellman–Ford algorithm cannot prevent routing loop, there is count-to-infinity problem. We adopt split horizon with poison reverse technique to solve count-to-infinity problem as in celebrated routing information protocol (RIP) [25]. For more details, please refer to RIP.

VI. Performance Evaluation

A. Methodology

We first compare the performance of COARP with the optimal solution obtained by ILP solver CPLEX [26]. Since ILP-based formulation is too complex to solve in real time under large-scale networks or dynamic network topologies, we use static network topologies with small-scale nodes. Three typical topologies are considered: well connected, partially connected, and disrupted. Although the topologies are simple, they represent various connectivity in real-world wireless networks. We compare the solutions of COARP and ILP-based optimal routing in all the three network topologies.

We then evaluate COARP in more complicated network scenarios comprehensively. The network topologies are dynamic and show diverse connectivity during simulations. We investigate the impact of various parameters on the performance, including node density, node mobility, and traffic load.

The performance of COARP is compared with that of three routing protocols: PROPHET [9], DSR [10], and DTTP [11]. PROPHET is designed specifically for intermittently connected networks, whereas DSR is for well-connected networks. They work only in their target environments. To make them work in other connectivity conditions, we have modified source codes of them in NS-2 without impacting their performance. DTTP is a hybrid routing protocol proposed recently, which supports both DTN and TCP/IP protocol stacks. DTTP is adaptive to network connectivity by switching between two exiting routing protocols: PROPHET and OLSR [21]. We have implemented DTTP in NS-2. We leave for future work the comparison of COARP with other adaptive routing protocols.

B. Comparing the Performance of COARP With the Optimal Solution

We setup three typical network topologies to compare the performance of COARP with ILP-based optimal solutions. The three topologies are well connected, partially connected, and disrupted connected, respectively, as shown in Fig. 1. There are 11 nodes totally in each network scenario: one source, one destination, and nine intermediate nodes. Solid lines are links between nodes.

Using these network topologies, we run our ILP model in CPLEX, and run COARP in NS-2. Their setup and solutions are illustrated as follows.

ILP-Based Optimal Solution: The primary input is link delay and available buffer size at each node. Table I lists available buffer size at each intermediate node. Table II gives hybrid link delay of links in a well-connected network as shown in Fig. 1(a).

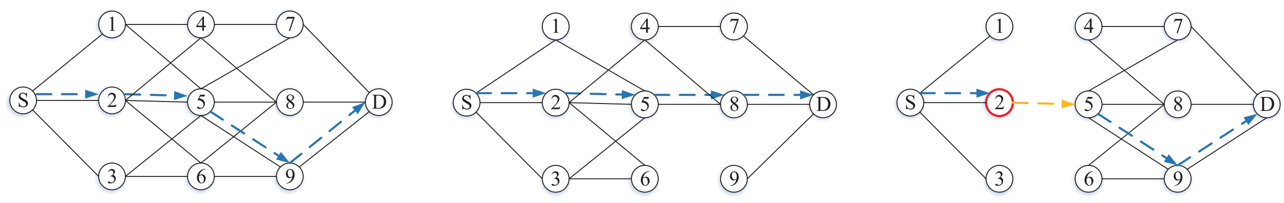


Fig. 1. Static network topologies and ILP-based solutions. (a) Well-connected topology. (b) Partially connected topology. (c) Disrupted topology.

TABLE I
NODE AVAILABLE BUFFER SIZE IN CPLEX

Node	1	2	3	4	5	6	7	8	9
Buffer size	3	5	2	4	1	0	2	0	3

TABLE II
HYBRID LINK DELAY SETUP IN CPLEX FOR WELL-CONNECTED TOPOLOGY

Link	Delay	Link	Delay	Link	Delay
(S,1)	7	(S,2)	3	(S,3)	4
(1,4)	5	(1,5)	1	(2,4)	12
(2,5)	4	(2,6)	6	(3,5)	13
(3,6)	11	(4,7)	1	(4,8)	2
(5,7)	10	(5,8)	9	(5,9)	4
(6,8)	2	(6,9)	7	(7,D)	9
(8,D)	2	(9,D)	3		

TABLE III
HYBRID LINK DELAY SETUP IN CPLEX FOR DISRUPTED TOPOLOGY

Link	Delay	Link	Delay	Link	Delay
(S,1)	7	(S,2)	3	(S,3)	4
(1,4)	100	(1,5)	100	(2,4)	100
(2,5)	100	(2,6)	100	(3,5)	100
(3,6)	100	(4,7)	1	(4,8)	2
(5,7)	10	(5,8)	9	(5,9)	4
(6,8)	2	(6,9)	7	(7,D)	9
(8,D)	5	(9,D)	3		

Hybrid link delay of each link is set to a small value in well-connected conditions. We use a large value, 100, to denote link disruption. Corresponding to the disrupted topology in Fig. 1(c), hybrid link delay of link (1, 4), (1, 5), (2, 4), (2, 5), (2, 6), (3, 5), and (3, 6) is set to 100 each. More details please refer to Table III. In the partially connected network shown in Fig. 1(b), hybrid link delays of link (5, 9), (6, 9), and (6, 8) are set to 100, the others remain the same as in Table II.

In CPLEX experiments, α and β in (2) are set to 0.9 and 0.1, respectively. We assign hybrid path delay more weights, since the network performance is more sensitive to delay than the caching cost. F is total packets which need to be cached in the network. We set F to 1 without loss of generality. T, is set to 30 s to normalize the hybrid link delay, since the maximum link delay is less than 30 s.

Dashed lines with arrows in Fig. 1. show optimal paths of ILP solutions. We can see that in Fig. 1(a), the optimal path is the one with the minimum delay, i.e., (S-2-5-9-D). Packets are not cached in intermediate nodes, since the network is well connected. In Fig. 1(b) with partially connected topology, the optimal solution is (S-2-5-8-D). Packets are not cached either. However, in Fig. 1(c), as the network is disrupted into two partitions, the optimal solution is (S-2-5-9-D). Note that link (2–5) has long hybrid link delay, i.e., it is disrupted. Therefore, packets are cached at node 2.

COARP Simulation Results: To verify the performance of COARP, we compare its simulation results with the ILP-based optimal solutions. To this end, we run COARP under three typical network topologies same as in Fig. 1. We set link delays proportional to that used in ILP-based solutions, as listed in Tables II and III, by adjusting the distance of the corresponding links in the simulation.

The simulation results show that COARP selects the same paths as that of ILP-based solutions in well-connected and partially connected topologies. In the disrupted topology, COARP selects the same path, but stores packets at node 1, 2, and 3, instead of only at node 2 as in the ILP-based solution. This is because that in COARP, a node stores a packet when it fails to deliver it to the next hop, and updates its caching cost. It then broadcasts new routing information table to its neighbors. Therefore, the change of the path is delayed. As a result, node 1, 2, and 3 store packets.

In a nutshell, COARP achieves the same solution as ILP-based optimal ones in all the static scenarios performed. We have carried out the comparisons in many other network scenarios, and the same results hold. Therefore, COARP is nearly optimal.

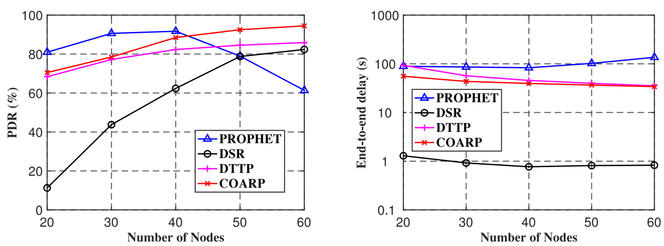
C. Simulation Results in Complex Scenarios

We further evaluate COARP in more complicated scenarios, and verify its performance in terms of Packet Delivery Ratio (PDR) and end-to-end delay. In these scenarios, we check how parameters impact its performance. Parameters we considered are as follows: node density, node movement pattern, and traffic load.

Simulation Setup: We carry out simulation on a rectangular flat space of 1500 m $\times$ 1500 m, with wireless nodes randomly deployed. A node moves according to random waypoint movement model, in which movement scenario files are characterized by pause time and maximum speed. Each node begins the simulation by remaining stationary for pause time seconds. It then

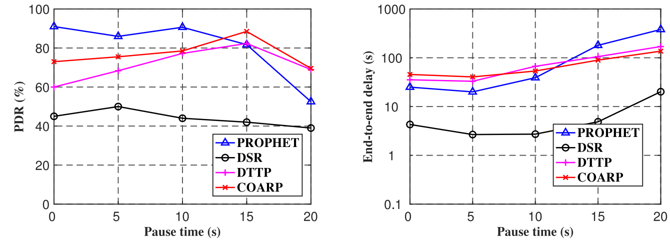
TABLE IV
PARAMETERS USED IN SIMULATION

Parameter	Value
IFQ size	50
Channel model	Wireless Channel
Radio model	TwoRayGround
Network structure	Wireless PHY
MAC type	802.11n
Interface queue	Droptail/PriQueue
Antenna model	OmniAntenna
Node movement model	Random waypoint



(a) (b)

Fig. 2. Performance with various node density. (a) PDR. (b) End-to-end delay.



(a) (b)

Fig. 3. Performance with various node movement. (a) PDR. (b) End-to-end delay.

selects a random destination in the area and moves to that destination at a speed distributed uniformly between 0 and some maximum speed. There are two pairs of source and destination. A source generates packets periodically to the destination. Each simulation ran for 600 s of simulated time. Primary parameters used in simulations are listed in Table IV. We run all four routing protocols on the same simulation setup.

Impact of Node Density: We examine the impact of node density on the performance of COARP. In the same rectangular flat space, we deploy different number of nodes: 20, 30, 40, 50, and 60, respectively. In this way, we simulate different node densities. The random waypoint model is set with pause time of 2 s and with maximum speed of 10 m/s. Each source generates packets in length of 512 Bytes, at a rate of 1 packet/s.

The simulation results are shown in Fig. 2. When node density is low, for example, of 20 nodes, COARP and DTTP have comparable PDR to that of PROPHET. However, the PDR of DSR is much lower, even less than 20%. This is because the network is partially connected or disrupted due to low node density and node mobility, and there is not always a path between the source and the destination. COARP, DTTP, and PROPHET cache packets at intermediate nodes and forward them when a new path is established. Thus they achieve high PDR. Since there is a delay to change the path while failing to deliver a packet, the PDR of COARP is a little lower than that of PROPHET, which always forwards packets to nodes with higher delivery predictability of the destination. With the increase of node density, the PDR of COARP keeps increasing, and is consistently higher than that of DSR. Whereas the PDR of PROPHET decreases after 40 nodes due to its heavy replication.

The PDR of COARP is always a little bit higher than that of DTTP. This is because that COARP is a unified single protocol, it can deal with the connectivity variance smoothly. When a path is broken, packets are stored naturally. In contrast, DTTP has to maintain two routing protocols: PROPHET and OLSR. When the network is well connected, OLSR is running. If packets fail to be sent when paths are broken, OLSR drops them. Packet loss causes DTTP to switch OLSR to PROPHET. PROPHET stores packets until there is a path. However, during the routing protocol switching, packet loss has happened. Therefore, COARP achieves higher PDR in diverse connectivity network, compared with DTTP.

Across various node density, the end-to-end delay of DSR keeps the smallest among the four protocols, even at low node density. This is because that end-to-end delay is calculated only for packets delivered to the destination in NS-2. For COARP, DTTP, and PROPHET, end-to-end delay is longer than DSR, since there are packets cached during path disruption. DTTP switches between PROPHET and OLSR according to network connectivity. It has comparable end-to-end delay with COARP. Since COARP is a single protocol, it has less overhead compared with DTTP. Moreover, COARP selects path with minimum hybrid link delay and caching cost, it outperforms PROPHET and DTTP at all node densities. In summary, COARP achieves high PDR and small end-to-end delay in diverse connectivity scenarios.

Impact of Node Mobility: We run our simulations with movement patterns generated for five different pause time: 0, 5, 10, 15, and 20 s. A pause time of 0 s corresponds to a continuous movement, and a pause of 20 s corresponds to a slow movement. We generate scenario files with 50 different movement patterns, 10 for each value of pause time. The maximum node speed is set to 10 m/s. All four routing protocols are run on the same movement patterns. We set the node number to 20. In this case, the network is intermittently connected.

The simulation results are shown in Fig. 3. We can see that when nodes move continuously, COARP achieves higher PDR than that of DSR, yet lower than PROPHET. Since COARP switches between forwarding and caching frequently, it sacrifices some PDR compared with PROPHET, but it improves the end-to-end delay at the same time. With node movement becoming slower, PROPHET gets less encounter probability, and thus lower PDR. Its end-to-end delay increases correspondingly for caching packets longer. However, the PDR of COARP keeps

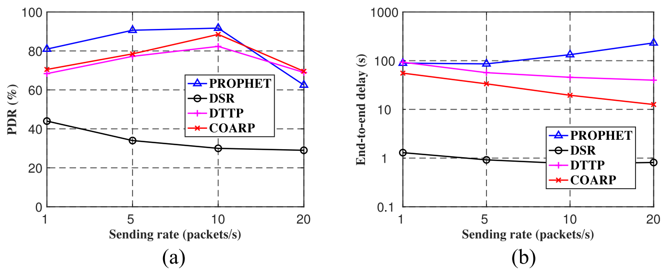


Fig. 4. Performance with various traffic load. (a) PDR. (b) End-to-end delay.

a high value across a variety of node movement patterns, and the end-to-end delay is smaller than that of PROPHET at slow node movement. Although the end-to-end delay of COARP is smaller than that of DSR, it is still at a receptive level. COARP consistently performs a little bit better than DTTP due to its less overhead.

Impact of Traffic Load: In this simulation, we vary traffic load in the network by changing packet sending rate at source nodes. The packet sending rate is set to 1, 5, 10, and 20 packet/s, respectively, in length of 512 Bytes. We deploy 20 nodes randomly in 1500 m $\times$ 1500 m flat space. The pause time and maximum speed used in the random waypoint model are 2 s and 10 m/s, respectively.

Fig. 4 shows the PDR and the end-to-end delay of four protocols under the simulation. The PDR of COARP increases with the sending rate from 1 to 10 packets/s, but drops when sending rate reaches 20 packet/s. This is because that COARP caches packets at intermediate nodes when the network is disrupted. The buffer at intermediate nodes will overflow when there are too much packets cached. PROPHET and DTTP show the same trend. The PDR of DSR is low through the variety of sending rates. The end-to-end delay of COARP is smaller than that of PROPHET. The end-to-end delay of DSR keeps around 1 s from beginning to end. The end-to-end delay of COARP is between that of PROPHET and DSR. Again COARP consistently performs a little bit better than DTTP.

In summary, COARP performs well in all cases we test. It achieves a good balance between routing protocols in well-connected and in intermittently connected networks. COARP have benefits of both types of routing protocols. In future work, we will compare the performance of COARP with other adaptive routings, for example, R3 [3].

VII. Conclusion

Malfunction of routing protocols hurts performance of delivering industrial data to the cloud/fog for further smart computing, thus hinders IIoT applications growth prospect. With the proliferation of IIoT, more and more wireless networks in real-world IIoT applications feature diverse connectivity. Most of previous work in routing protocols for diverse connectivity networks integrate two existing routing protocols, one for well-connected and the other for intermittently connected environment. In this paper, we have proposed a unified routing metric to capture the connectivity diversity. Base on the unified metric, we have presented an ILP formulation to model the adaptive routing problem and implemented a heuristic routing protocol COARP for practical considerations. The performance of COARP have been verified by extensive simulations. In future work, we will implement COARP in our testbed and explore its feasibility in practical wireless networks with connectivity diversity for IIoT applications.

References

[1] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Inform.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.
[2] C. C. Byers, "Architectural imperatives for fog computing: Use cases, requirements, and architectural techniques for fog-enabled IoT networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 14–20, Aug. 2017.
[3] X. Tie, A. Venkataramani, and A. Balasubramanian, "R3: Robust replication routing in wireless networks with diverse connectivity characteristics," in *Proc. 17th Ann. Int. Conf. Mobile Comput. Net.*, 2011, pp. 181–192.
[4] S. Latif *et al.*, "Industrial internet of things based efficient and reliable data dissemination solution for vehicular ad hoc networks," *Wireless Commun. Mobile Comput.*, vol. 2018, 2018, Art. no. 1857202.
[5] M. Asadpour, K. A. Hummel, D. Giustiniano, and S. Draskovic, "Route or carry: Motion-driven packet forwarding in micro aerial vehicle networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 843–856, Mar. 2017.
[6] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. 10th Ann. Int. Conf. Mobile Comput. Netw.*, 2004, pp. 114–128.
[7] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proc. Conf. Appl., Technol., Architectures, Protocols Comput. Commun.*, 2004, pp. 145–158.
[8] The Network Simulator NS-2. [Online]. Available: https://www.isi.edu/nsnam/ns/
[9] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," in *Proc. 14th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2003.
[10] D. B. Johnson, D. A. Maltz, and J. Broch, "Ad hoc networking," in *DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001, pp. 139–172.
[11] X. Chang, Z. Zhao, B. Diao, and T. Li, "Experimental study of disruption-tolerant transport protocol for mobile ad-hoc networks with connection diversity," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2016, pp. 1613–1620.
[12] S. Petersen and S. Carlsen, "WirelessHART versus ISA100.11a: The format war hits the factory floor," *IEEE Ind. Electron. Mag.*, vol. 5, no. 4, pp. 23–34, Dec. 2011.
[13] S. Montero, J. Gozalvez, M. Sepulcre, and G. Prieto, "Impact of mobility on the management and performance of WirelessHART industrial communications," in *Proc. IEEE 17th Int. Conf. Emerg. Technol. Factory Autom.*, Sep. 2012, pp. 1–4.
[14] J. Ott, D. Kutscher, and C. Dwertmann, "Integrating DTN and MANET routing," in *Proc. SIGCOMM Workshop Challenged Netw.*, 2006, pp. 221–228.
[15] I. chakeres, "Dynamic MANET on-demand (DYMO) routing," 2008.
[16] J. Whitbeck and V. Conan, "HYMAD: Hybrid DTN-MANET routing for dense and highly dynamic wireless networks," *Comput. Commun.*, vol. 33, no. 13, pp. 1483–1492, 2010.
[17] F. Esposito and I. Matta, "Preda: Predicate routing for DTN architectures over MANET," in *Proc. Global Telecommun. Conf.*, 2010, pp. 5018–5023.
[18] Y. Kawamoto, H. Nishiyama, and N. Kato, "Toward terminal-to-terminal communication networks: A hybrid MANET and DTN approach," in *Proc. IEEE 18th Int. Workshop Comput. Aided Model. Des. Commun. Links Netw.*, Sep. 2013, pp. 228–232.
[19] M. Ito, H. Nishiyama, and N. Kato, "A novel routing method for improving message delivery delay in hybrid DTN-MANET networks," in *Proc. IEEE Global Commun. Conf.*, 2013, pp. 72–77.
[20] L. Delosieres and S. Nadjmtehrani, "Batman store-and-forward: The best of the two worlds," in *Proc. IEEE PerCom Workshops*, 2012, pp. 721–727.
[21] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," *Manet Working Group*, vol. 527, no. 2, pp. 1–4, 2003.

[22] C. Yang and R. Stoleru, “Hybrid routing in wireless networks with diverse connectivity,” in *Proc. 17th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2016, pp. 71–80.
[23] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.
[24] A. Balasubramanian, B. Levine, and A. Venkataramani, “DTN routing as a resource allocation problem,” in *Proc. Conf. Appl., Technol., Architectures, Protocols Comput. Commun.*, 2007, pp. 373–384.
[25] C. Hendrik, *Routing Information Protocol*, RFC 1058, Jun. 1988. [Online]. Available: https://www.rfc-editor.org/info/rfc1058
[26] IBM ILOG CPLEX optimization studio. [Online]. Available: https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer

Zenghua Zhao (M’06) received the B.S., M.S., and Ph.D. degrees in computer applied technology from Tianjin University, Tianjin, China, in 1995, 2000, and 2006, respectively.

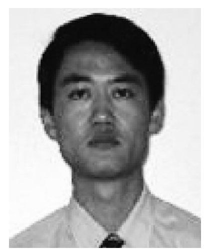
She is an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. She is a member of Tianjin Key Laboratory of Advanced Networking. Her research interests include mobile computing, long-distance wireless mesh networks, software defined networking, and Internet of Things.

Yiyao Shi received the B.S. degree in network engineering from Chongqing University, Chongqing, China, in 2017. She is currently working toward the master’s degree in computer applied technology from the College of Intelligence and Computing, Tianjin University, Tianjin, China.

Her research interests include wireless networking and mobile computing.

Bingxue Diao received the B.Eng. degree in software engineering from Tianjin Polytechnic University, Tianjin, China, in 2015, and the M.Eng. degree in computer applied technology from Tianjin University, Tianjin, China, in 2018.

She is currently an Operation and Maintenance Support Engineer with the Beijing Jingdong Century Trading Co., Ltd., Beijing, China. Her research interests include wireless networking and mobile computing.



Bin Wu received the B.Eng. degree in power electronics from ZheJiang University, Hangzhou, China, in 1993, the M.Eng. degree in signal processing from the University of Electronic Science and Technology of China, Chengdu, China, in 1996, and the Ph.D. degree in computer systems and networking from the University of Hong Kong, Pokfulam, Hong Kong, in 2007.

He is currently a Professor at the College of Intelligence and Computing, Tianjin University, Tianjin, China. He was a Research Associate from 2007 to 2012 with the ECE Department, University of Waterloo, Waterloo, ON, Canada. His research interests include computer systems and networking, wireless/wireline communications, datacenter networks, and cloud computing.

Prof. Wu was a recipient of the Postdoctoral Research Fellowship from the Natural Science and Engineering Research Council of Canada.