IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 June 2022, accepted 13 August 2022, date of publication 19 August 2022, date of current version 1 September 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3200467*

APPLIED RESEARCH

# Realizing Opportunistic Routing in Multi-Channel Environments

**CHE-JUNG HSU[1] AND HUEY-ING LIU[2]**

[1]Graduate Institute of Applied Science and Engineering, Fu Jen Catholic University, New Taipei City 242062, Taiwan
[2]Department of Electrical Engineering, Fu Jen Catholic University, New Taipei City 242062, Taiwan

Corresponding author: Huey-Ing Liu (hiliu@mail.fju.edu.tw)

**ABSTRACT** Opportunistic Routing (OR), which utilizes the broadcast nature of wireless communication to route packets in a dynamic way, significantly improves the transmission quality. Most of the researches focus on the single channel scenario; however, multi-channel (MC) is widely adopted over wireless communication systems. How to apply OR over MC environments is quite challenging and still an unsolved issue. OR operating over single channel environments collects acknowledgements mainly through overhearing data packets. In contrast, MC environments increase the performance by reducing interference and providing more bandwidth. Since users scatter over multiple channels, the chance of overhearing is reduced. Without overhearing acknowledgements (ACKs), the performance of OR degrades due to duplicate transmissions. To better understand the behavior of OR over MC environments, this paper first surveys current OR methods and constructs a simplified scenario to observe how they behave as the number of interfaces and channels increases. Then, analyze and compare their costs and penalties to find out a potential method. Finally, refine the potential method for MC environments. It is observed that token passing is the most potential method for its high efficiency of utilizing control packets. To apply token passing to OR in MC environments, this paper proposed multi-channel token passing (MCTP) to resolve the hazards of token passing paths and dynamic rate for multiple flows. Simulation results show that MCTP has high performance with low and consistent overheads. In addition, MCTP is scalable with increasing interface and channel numbers, which makes it a prototype for the future OR research in MC environments.

**INDEX TERMS** Opportunistic routing, anypath routing, multi-channel, wireless mesh networks.

## I. INTRODUCTION

In recent years, wireless networks have played an important role in our daily life for its convenience and cost efficiency. For multi-hop wireless networks, the broadcast nature, which provides pro and con at the same time, makes routing challenging. The pro is that everyone within the transmission range is able to overhear the packet and become a backup link if the original relay fails to successfully receive the packet. The con is that multiple nodes may transmit packets simultaneously and induce interference. Thus, a collision avoidance or prevention mechanism is required to prevent interference and maintain high performance. Traditional routing (TR) treats wireless links as point-to-point wired links; packets are forwarded to the destination along a given route. At each transmission, the packet is sent to a specified relay and packets overheard by neighbors are dropped. Instead of treating the broadcast nature as a disadvantage, OR, which is also called anypath routing, utilizes those overheard packets to improve performance by dynamically selecting the best relay to forward the packet. With multiple backup links and the possibility of reaching further, OR significantly improves the overall reliability and throughput.

Due to the technology improvement and massive bandwidth requirements of current audio and video rich applications, modern devices are equipped with multiple interfaces to obtain higher gain from multi-channels. With multiple channels available, performance increases as more active links are available and interference is reduced. However, the chance of overhearing is also reduced, which is essential for OR. When a data packet is heard by multiple relays, only the best relay forwards the data packet to downstream

The associate editor coordinating the review of this manuscript and approving it for publication was Hosam El-Ocla.

relays. Notably, for a referenced node, the previous nodes close to the source denote as upstream nodes of it and the nodes close to the destination are downstream nodes. The forwarded data packet also acts as an acknowledgement for other relays to prevent duplicate transmissions without additional control packets. In other words, OR coordination methods collect acknowledgements mainly through overhearing data packets. However, in MC environments, relays may forward data packets to downstream relays through channels that other relays cannot hear. Without overhearing acknowledgements, duplicate transmissions occur and degrade the system performance for multi-channel OR (MCOR).

This paper studies the issue of applying OR over MC environments. The major contributions of this paper are as follows:

1) Provide a brief survey of ORs for readers to know the current status of ORs in MC environments.
2) Analyze coordination overheads in terms of duplicate data packets and additional control packets for various OR methods in a simplified MC environment. The analysis also provides a guide for future design of OR methods for MC environments.
3) Illustrate the efficiency of OR coordinated by token passing and resolve its hazards in MC environments; neither of them is discussed in the previous works. This is also a state-of-art solution for OR in MC environments.

The rest of this article is organized as follows. Sections II gives the background of OR and surveys OR methods. Section III analyzes and compares overheads of OR methods. Then, a coordination method based on token passing is proposed for MC environments in section IV. Section V presents the performance evaluations for different ORs and finally, conclusions are included in section VI.

## II. BACKGROUND AND RELATED WORKS

### A. OR AND ITS COMPONENTS

Opportunistic routing (OR) [1], [2], [3] improves performance of wireless networks by utilizing overheard packets; it broadcasts packets to multiple candidates and dynamically selects the best candidate to send the packet. Generally, the basic operation of ORs consists of the following two steps:

1) Broadcast: broadcast a packet,
2) Coordination: the candidates, which successfully receive the packet, coordinate to determine a winner to relay the packet.

These two steps repeat until the packet reaches the destination.

Consider the example as shown in Fig. 1, where $r_1$ is sending data to $r_4$ through relays $r_2$ and $r_3$. First, $r_1$ broadcasts a packet and $r_2$ and $r_3$ overhear this packet. Then, $r_2$ and $r_3$ need to run a coordination method to decide the best relay. Finally, the packet is forwarded to $r_4$ by the best relay ($r_3$ in this example).

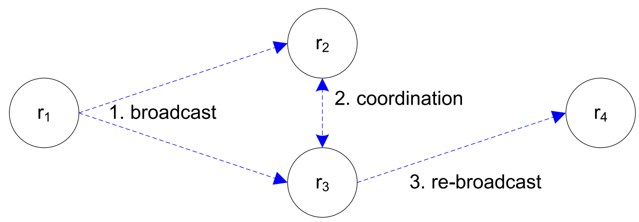


**FIGURE 1.** Basic steps of OR.

By utilizing the broadcast nature of the wireless medium, OR processes two advantages: higher reliability and reaching farthest. With multiple candidates available, they act like multiple backup links and form a virtual link with high reliability. Besides, long-ranged links with poor quality, which cannot be utilized in TR, are taken into consideration all the time; OR catches every good luck and directly jumps to the farthest relay, which receives the packet.

Though ORs bring those advantages, there are two major challenges of implementing ORs: *coordination method* and *candidate selection*. Candidate selection defines a list of prioritized relays and coordination methods pick the best relay from the list. Candidate selection is similar to routing tables in TR; it considers what is essential in the scenario, such as transmission cost or interference, and a whole picture of them is given in the final subsection. Once a packet is broadcasted, a coordination method selects the relay with the highest priority among the relays that successfully received the data packet to forward the packet to downstream relays. The coordination method selects the relay in distributed mode and prevents duplicate transmissions with the lowest overhead. Several researches have been proposed for coordination methods and they are classified into three categories: overhearing [4], [5], network coding [13], [14], [15], [16], [17], [18] and token passing [19]. The operation details for each class of coordination methods are stated in the following subsection.

### B. COORDINATION METHODS

#### 1) COORDINATION BY TIMER AND OVERHEARING

Coordination by overhearing, is the most straightforward and common approach; it utilizes overheard data packets as acknowledgements and does not induce additional control packets. In coordination by overhearing, timer is the primary means to determine priority. After receiving a data packet, each candidate sets a timer to delay its response; a shorter timer represents a higher priority, which is based on the candidate selection. The successfully received candidates with the shortest timer (i.e. the highest priority) broadcasts the packet first. Once the packet transmission is overheard, the other relays treat it as an acknowledgement and simply drop their data packets.

Consider the example in Fig. 2, src transmits packets to dst through relays $r_1$, $r_2$ and $r_3$ and the priority is $r_3 > r_2 > r_1$ with timer setting as $t_3 < t_2 < t_1$, respectively. After src broadcasts a data packet, $r_3$ with the highest priority responds while $t_3$ is timeout. Because $r_3$ failed to receive the packet,

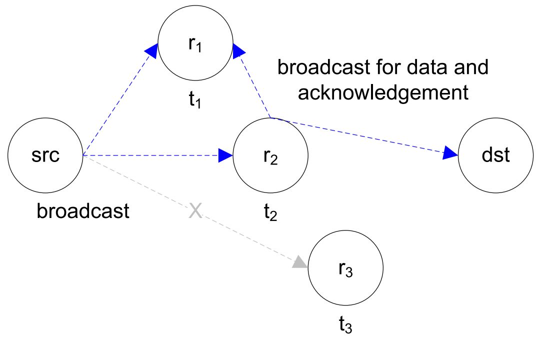


**FIGURE 2. Example of OR coordinated by timer and overhearing.**

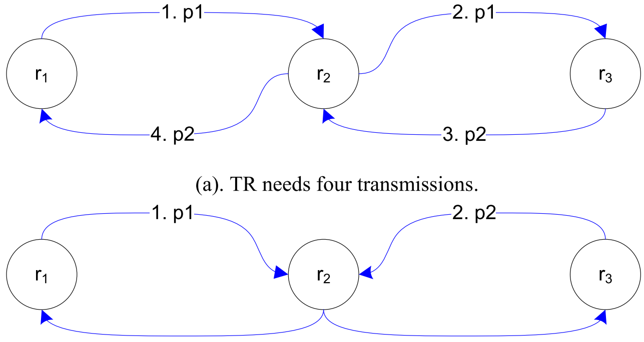


**FIGURE 3. An illustrated example of IXNC.**

nothing was sent. Without acknowledgements from higher priority relays, $r_2$ considers it as the best relay and forwards the data packet to dst while $t_2$ expires. The data packet sent from $r_2$ to dst also acts as an acknowledgement and prevents $r_1$ from sending duplicates. Coordination by overhearing does not induce extra control packets; however, the main challenge of this method is the duplicate transmissions when relays fail to overhear each other (ex: $r_1$ failed to overhear $r_2$ in Fig. 2) and the condition gets severe in the MC environment.

ExOR [4] is the first study to demonstrate the idea of OR; it runs coordination by overhearing for a batch of packets. When a timer of a relay expires, the relay broadcasts all unacknowledged packets in the batch. Unlike ExOR, SOAR [5] runs a coordination for each packet. To cope with unsuccessful overhearing, SOAR triggers an additional standalone-ACK after a pre-defined duration timeout.

### 2) COORDINATION BY NETWORK CODING

Network coding [6], [7] utilizes the idea of mixing packets for transmissions to improve performance. Generally, there are two types of network coding: inter-flow network coding (IXNC) and intra-flow network coding (IANC) [8]. IXNC improves performance by mixing packets of different flows. Consider the example of Fig. 3(a), there is a packet p1 from $r_1$ to $r_3$ and another packet p2 from $r_3$ to $r_1$. With traditional routing, it takes four transmissions overall to accomplish the task. With IXNC, one transmission is saved by broadcasting p1+p2 from R2 as shown in Fig. 3(b). However, IXNC in wireless networks is not practical due to computational complexity [8]. IXNC needs additional efforts to recalculate for coding opportunities for any change of traffic patterns. Besides, it relies on high quality links and needs to handle conditions that nodes may fail to receive packets for decoding. Though, there are researches that improve performance by combining IXNC with OR [9], [10], [11], [12], they do not resolve the coordination issue with IXNC and use timer or IANC for coordination instead.

In contrast, the coordination issue can be resolved by IANC. MORE [13] proposed an OR coordination method by random linear network coding. Due to the nature of network coding, packets are transmitted batch by batch. Native packets are coded before each transmission and decoded only at destination. Source and relays broadcast random linear combinations of native packets. When the destination has collected sufficient linearly-independent packets, the batch of packets is restored with Gaussian elimination. Finally, an end-to-end batch-ACK is sent from destination to source using traditional routing and a new batch is then triggered by the source.

Consider the example of Fig. 4, where src delivers a batch of two packets, a and b, to dst through relay $r_1$ and $r_2$. src broadcasts coded packets p1 and p2, which are random linear combinations of native packets a and b. Though $r_1$ missed p2 and $r_2$ received both, dst is still able to decode the batch and restore the native packets a and b with two received linearly-independent packets p3 and p4.

Though IANC eliminates duplicate transmissions, it may induce redundant transmissions. Only linearly independent coded packets are meaningful and provide helpful information. Even though linearly dependent coded packets are not duplicate, they provide no additional information and are therefore redundant. Consider the example of Fig. 4, if $r_1$ sends one more coded packet, the packet is linearly dependent and redundant. The core of the redundant issue is when relays should generate coded packets. MORE handles the issue by a credit system. it predicts how often a relay should trigger a coded packet in terms of incoming packets. On each packet arrival, relays receive credits, which is calculated based on packet delivery ratio (PDR) between relays. When the credit counter reaches a threshold, the relay resets the counter and generates a coded packet. However, the prediction is inaccurate and still causes redundant transmissions [14]. As a result, the issue of when to generate coded packets is the main challenge to keep coordination by network coding efficient.

After MORE is proposed, there are several researches enhancing the work. Different from MORE with only one batch under transmissions, CodeOR [15] allows more batches in flight and eliminates the idle time of transmitting batch-ACK from destination to source. SlideOR [16] further allows adding new packets to batches dynamically so that OR coordinated by network coding is no longer limited by fixed batches. It is also observed that OR coordinated by network coding induces more transmissions. To reduce the

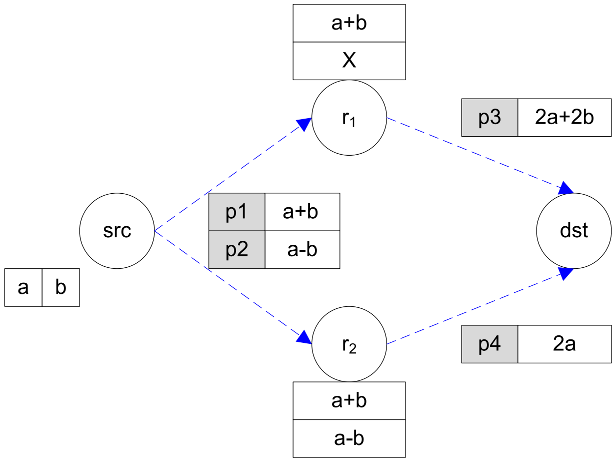


**FIGURE 4.** An illustrated example of IANC.

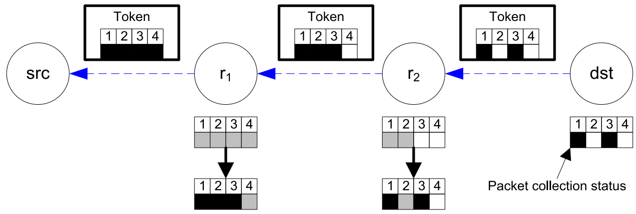


**FIGURE 5.** An illustrated example of coordination by token passing.

number of transmissions, CCACK [14] adds acknowledgements in header; relays learn to stop transmissions when downstream relays have collected sufficient information. ONCR [17] picks the optimal candidate set with minimum transmission cost by linear programming. The rate control issue is addressed in [18], it estimates the initial rate based on expected transmissions of relays and dynamically adjusts rate by feedback.

#### 3) COORDINATION BY TOKEN PASSING

Token-based coordination prevents duplicate transmissions by a control packet, denoted as token; and it is first proposed in ECONOMY [19]. Though coordination by token passing is not as common as timer/overhearing and network coding, it significantly reduces duplicates with only the cost of token. Acknowledgements are also piggybacked in tokens. The token is generated by destination and passed to source through all candidates along the route. During the procedure, the token is passed strictly from high priority relays to low priority relays; a relay is permitted to send packets only when a token arrives. With the information included in the token, the relay has a clear view of what packets higher priority relays have received and transmits only unacknowledged packets.

In the example in Fig. 5, tokens flow from dst to src through $r_1$ and $r_2$. The box next to each relay shows the collected packet status: white for unacknowledged, gray for collected, and black for acknowledged. When a token is sent from $r_2$ to $r_1$, $r_1$ learns not only the status of $r_2$ but also the status of dst. Suppose packet p3 sent from $r_1$ is heard by dst but missed by $r_2$. Then, the acknowledgement of p3 is piggybacked in the token delivered from dst to $r_2$ and arrives at $r_1$. With acknowledgements in token passing, $r_1$ knows that p1, p2 and p3 have been received by higher priority relays, such as $r_2$ or dst. As a result, $r_1$ transmits unacknowledged packets (p4) without duplicates.

### C. CANDIDATE SELECTION

Candidate selection and their priorities are mainly determined by the routing metrics. There are various considerations in the metric design. Initially, ETX [20] is widely used for the candidate selection in OR. ETX is originally designed for TR and it considers expected transmission count of all links of a route. However, link cost of ETX considers only one relay (i.e., one candidate) but OR utilizes multiple backup links (i.e., multiple candidates). As a result, ETX cannot fully capture the transmission count of OR and is not suitable for OR. Then, expected anypath transmission (EAX) [21] is developed to capture the expected transmission count of OR. EAX takes multiple candidates of OR into consideration and re-calculates the expected transmission count based on the probability that at least one candidate receives the data packet. For MC environments, EAD [22] extends EAX and considers intra-flow interference; it adds penalties to overlapping channels. However, due to the nature of MC environments, finding the optimal solution is time consuming and providing some heuristic algorithms to efficiently reduce the time complexity is still an issue for EAD.

Metrics above need to build routing tables in advance using distance vector or link-state protocols. In contrast, TSOR [23] is an online learning algorithm that builds OR routes without out-of-band packets. It assumes that link conditions are unknown in advance and learns link costs from each data transmission; therefore, the algorithm is suitable for highly dynamic networks. Furthermore, the candidate set is also affected by transmission power. A larger transmission power covers more relays, increases node degrees and improves link qualities; on the other hand, it also consumes more energy and induces higher coordination overheads. ERTO [24] formulates an optimization problem considering packet delivery probability, energy consumption and node degree; and finds the transmission power that maximizes link quality and minimizes energy consumption.

### D. OR IN OTHER SCENARIOS

In addition to wireless mesh networks, OR is applied to wireless sensor networks (WSN) and underwater acoustic sensor networks (UASN) for its potential to improve reliability and throughput. They have various applications such as environmental monitoring, undersea explorations and disaster prevention. Generally, OR in WSN and UASN adopts timer and overhearing for coordination and selects routing metrics for different purposes.

For OR in WSN, researches focus on energy saving to prolong the lifetime of WSN. LORA [25] proposed a metric for load balancing which considers energy and distance in three dimensions. Nodes with greater remaining energy are given higher priority to avoid earlier energy depletion of some nodes. Expected transmission cost (ETC) [26] is a metric to include rendezvous costs (time for nodes to wake up and align with others). By choosing routes with less idle time, it reduces latency and energy consumption.

Regarding OR in UASN, depth obtained from pressure sensors is a common routing metric; however, routing by depth may get lost and enter a void region (unable to find a valid forwarder to destination) or lead to detour forwarding. Some UASN researches focus on metrics to avoid void regions. DVOR [27] builds distance vectors by hop-count; and NA-TORA [28] uses normalized advancement (NA), which considers ETX and energy consumption, to detect and avoid void nodes. Furthermore, nodes with multi-modal (each node is equipped with a set of acoustic modems) have been proposed in recent years. OMUS [29] considers multi-modal and proposes a heuristic method to pick the modem that consumes the lowest energy to meet the desired link quality.

## III. CONSIDERATIONS FOR MULTI-CHANNEL ENVIRONMENTS

To compare the efficiency of coordination methods and how they perform over MC environments with fewer chances of overhearing, this section first introduces a simplified two-hop scenario and analyzes the behaviors of various coordination methods over such a simplified scenario. Then, overheads of each OR coordination method, including costs for coordination and penalties for failure, are presented. Finally, compare and conclude the most potential coordination method for the MC environments.

### A. GENERAL ASSUMPTIONS

To better understand the influence factors in the MC environment, some assumptions are made in advance. Assuming there are $N$ nodes in the wireless network and each node is equipped with $I$ interfaces. There are $C$ orthogonal channels and a randomly selected channel is assigned to each channel. Notably, no two or more interfaces of a node are assigned to the same channel. To route packets with OR, $R$ nodes are selected from $N$ nodes as relays in the OR route by a metric such as ETX/EAX/EAD. Relays are ordered by the metric, where $r_R$ has the highest priority and $r_1$ has the lowest priority.

### B. SIMPLIFIED TWO-HOP SCENARIO

For clarity and simplicity, the following paragraphs analyze overheads of different coordination methods under a simplified scenario as shown in Fig. 6. In the scenario, a source, denoted as src, sends a batch of $B$ packets to a destination, denoted as dst, through a set of $R$ relays: $r_1$ to $r_R$. The packet delivery ratio (PDR) of a link from $r_i$ to $r_j$ is denoted as PDR($r_i$, $r_j$). For simplicity, the PDR of all links are assumed to be $p$ in this scenario. The src and dst are assumed to be outside of transmission range of each other so that they cannot reach one another; however, all the relays are reachable by the src, the dst and all other relays. Moreover, the transmission of packets is strictly two-hop transmission; data packets sent from relays to relays are eliminated. This strict assumption makes the analysis focus on the behavior of a single coordination.

Notably, in MC environments, relays may transmit packets at different channels that other relays cannot hear. To focus on the condition of coordination failure, the following assumptions are given:

1) The src and dst are equipped with $C$ interfaces to hear all channels.
2) Relays transmit data/control packets at random channels.

According to the assumptions, a relay may fail to hear data or control packets from the other relays with the probability of (1 - $I/C$).

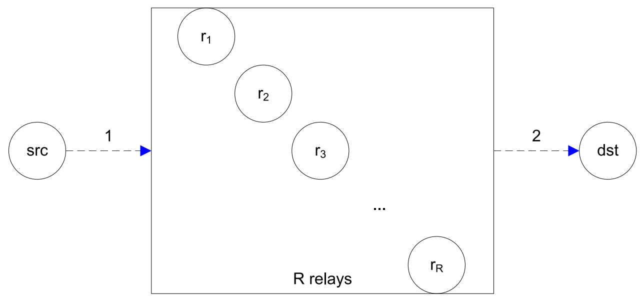


**FIGURE 6.** The simplified two-hop scenario.

### C. CALCULATION OF SUCCESSIVE FAILURE

Before analyzing overhead, a formula to calculate successive failure is first introduced. Function Failure(AN, $p$) stands for the chance that a relay failed to receive AN acknowledgements, where each acknowledgement is successfully received with a probability of $p$. The main concern of the calculation comes from the condition when AN is an integer or not. When AN is an integer, it is intuitive that the chance of successive failure of integer AN, denoted as Failure_I, is defined as (1). However, when AN is a fraction, it is more appropriate to calculate successive failures of fraction AN, denoted as Failure_F, by expectation as (2).

$$\text{Failure\_I}(\text{AN}, p) = (1-p)^{\text{AN}} \tag{1}$$

$$\text{Failure\_F}(\text{AN}, p) = \text{AN} \times (1-p) + (1-\text{AN}) \tag{2}$$

For example, consider AN = 0.7 and $p = 0.5$. The AN = 0.7 stands for 0.7 acknowledgement sent from other relays, in other words, an acknowledgement arrives with a probability of 70%. The $p = 0.5$ stands for the packet delivery ratio is 50%, which means that a node has 50% to hear an arrived acknowledgement. With (2), the relay is expected to fail to receive acknowledgements with a probability of 65% = 0.7× (1-0.5) + 0.3 × 1. The correct probability is 0.65 rather than 0.615 = $(1-0.5)^{0.7}$ according to (1). The gap between Failure_I and Failure_F is 5.7% = (65% / 61.5%) −1 and it increases as $p$ increases as shown in Fig. 7.

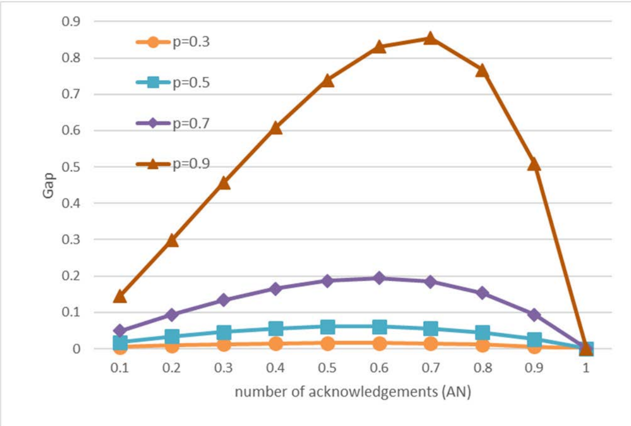


**FIGURE 7. The gap between Failure_I and Failure_F with different PDRs.**

To precisely calculate the probability of successive failure, a new equation (3) is defined by combining (1) and (2) to handle the integer and fraction parts of AN. Take AN = 3.7 for example, Failure(3.7, $p$) = Faiulre_I(3) × Failure_F(0.7).

$$\begin{aligned}\text{Failure}(\text{AN}, p) &= [\text{fraction}(\text{AN}) \times (1-p) + (1-\text{fraction}(\text{AN}))] \\ &\quad \times ((1-p)^{\text{floor}(\text{AN})}) \end{aligned} \tag{3}$$

### *D. COST AND PENALTY ANALYSIS*

In the above simplified two-hop scenario, a packet is sent by the src to a set of relays. Then, one or multiple relays forward the packet to the dst depending on coordination results. Various coordination methods are compared by overhead, which is measured by the number of additional transmissions required to deliver a batch of $B$ packets from the src to the dst. The overhead comes from the cost and the penalty as (4). The coordination costs are caused by control packets such as tokens or standalone-ACKs. As OR utilizes overheard data packets as acknowledgements; standalone- ACKs are initiated when nodes fail to hear them due to interference or on different channels. The penalties are caused by data packets such as duplicate packets or coded packets without new information.

$$\text{Overhead} = \text{Cost}(\text{control packet}) + \text{Penalty}(\text{data packet}) \tag{4}$$

Regarding the fundamental transmission number, TR requires a number of $1/p + 1/p$ transmissions to send a packet from the src to reach the dst. For the ideal case of OR, there is no coordination cost and penalties. When src broadcasts a packet, the chance that at least one relay received the packet is $1-(1-p)^R$. It requires $1/(1-(1-p)^R)$ transmissions to reach the relay set and another 1/p transmission from the relay set to dst. Therefore, it requires $1/(1-(1-p)^R) + 1/p$ transmissions overall in the ideal case of OR.

Consider an example with 3 relays ($R = 3$) and PDR $p = 0.7$. TR requires 1/0.7 = 1.428 transmissions from src to relay set and another 1.428 transmissions from relay to dst, thus 1.428 + 1.428 = 2.856 transmissions overall. The ideal case of OR requires 1/0.973 = 1.027 transmissions from src to relay set and 1.428 transmission from relay set to dst as TR; therefore, OR requires 1.027 + 1.428 = 2.455 transmissions overall. In the simplified two-hop scenario, the transmission number of OR is (1 − 2.45/2.85) = 14% fewer than that of TR. When hop count increases and most links are as efficient as OR, the difference may reach up to (1 − 1.027/1.428) = 28%.

Notably, the transmission number of OR is evaluated under the ideal case and it increases with overhead. The following paragraph analyzes the number of extra transmissions induced by various coordination methods.

#### 1) COORDINATION BY TOKEN PASSING

The simplified behavior of token-based coordination, denoted as TP, is listed below:

1) The src broadcasts a batch of $B$ packets until each packet is received by at least one relay.
2) A token carrying acknowledgements sent from the dst flowing through all relays in decreasing order by unicast. The token is generated by the dst and passed to $r_R$. Then, the token is passed from $r_i$ to $r_{i-1}$ and finally arrives at the src. The latest acknowledgments of $r_i$ are added in the token, while $r_i$ passes the token to $r_{i-1}$.
3) Relays forward all of received but unacknowledged packets to the dst, while the token is in hand.

As the token flows through all high priority relays, a relay $r_i$ can obtain the latest acknowledgements from $r_R$ to $r_{i+1}$ and forward packets to dst without duplicates. Therefore, with the cost to transmit $R$ tokens, which costs $R/p$ transmissions, token-based coordination obtains zero penalties. The condition holds in multi-channel environments as well as long as a token passing path is found.

#### 2) COORDINATION BY TIMER AND OVERHEARING

The simplified behavior of timer/overhearing-based coordination, denoted as TOH, is listed below:

1) The src broadcasts a batch of $B$ packets until each packet is received by at least one relay.
2) Relays forward received but unacknowledged packets to the dst starting from $r_R$ to $r_1$.
3) When overhearing a packet sent by higher priority relays, lower priority relays treat it as an acknowledgement and relinquish to send the same packet.

No extra cost is required for the TOH-based coordination because acknowledgements are fulfilled by overhearing data packets. However, there are penalties that come from duplicate transmissions when relays fail to overhear. When a packet is forwarding to the dst by the best relay $r_{best}$, a lower priority relay, say $r_i$, has a probability of PDR(src, $r_i$) to hold the same packet and a probability of (1-PDR($r_{best}$, $r_i$)) to miss the acknowledgement from the best relay. Therefore relay $r_i$ has a chance of PDR(src, $r_i$) × (1-PDR($r_{best}$, $r_i$)) to trigger a duplicate transmission. Depending on the design of the coordination method, multiple duplicate transmissions from multiple relays are possible. For the best case that only one relay triggers a duplicate transmission, the penalty is $p \times (1-p)$ transmissions. This is the lower bound of the penalty for TOH-based coordination to deliver one packet from src to dst.

In the MC environments, relays may not listen at the channel that the other relay transmits. The relay $r_i$ has a probability of $I/C$ to hear data packets from src and acknowledgements from $r_{best}$. Similar to the calculation of that in single channel scenario, relay $r_i$ has a probability of $I/C \times p$ to receive a packet from src and a probability of Failure$(I/C, p)$ to miss an acknowledgement from $r_{best}$. Therefore, the least penalty of TOH-based coordination over MC environments, denoted as mc_penalty$_{TOH}$ is generalized as (5).

$$\text{mc\_penalty}_{TOH} = I/C \times p \times \text{Failure}(I/C, p) \quad (5)$$

Consider an example of $p = 0.7$ and $R = 3$, the penalty is at least $0.21 = 0.7 \times (1 - 0.7)$ duplicate transmissions for the single channel scenario. For a multi-channel environment with $C = 3$ and $I = 2$, the penalty increases to $0.249 = 2/3 \times 0.7 \times (2/3 \times 0.3 + 1/3 \times 1)$.

### 3) COORDINATION BY STANDALONE-ACKS

Coordination purely by standalone-ACKs, denoted as SACK, is too costly so that it generally acts as a support for other coordination methods. As shown, the probability of overhearing over MC environments is significantly decreased. The SACK plays an important role in MC environments; thus, the method is included for comparison and the simplified behavior is listed below:

1) The src broadcasts a batch of $B$ packets until each packet is received by at least one relay.
2) Each relay broadcasts AF standalone-ACKs to all interfaces; the parameter AF is defined by the frequency of standalone-ACKs.
3) Relays forward unacknowledged packets to the dst.

There is a trade-off between cost and penalty with coordination by the SACK and is controlled by the frequency AF. To deliver acknowledgements to all neighbors, it costs additional $R \times \text{AF} \times I$ transmissions per batch for all $R$ relays to broadcast AF standalone-ACKs over $I$ interfaces. Regarding penalties per packet, the best relay $r_{best}$ transmit $\text{AF} \times I$ standalone-ACKs and a low priority relay $r_i$ can receive $\text{AF} \times I \times (I/C)$ on average. The relay $r_i$ has a chance of $I/C \times p$ to receive the packet from src and triggers a duplicate transmission when all acknowledgements from $r_{best}$ fail, which is Failure$(\text{AF} \times I \times I/C, p)$. Though multiple duplicate transmissions from multiple relays are possible, this study considers the best case that only a single relay triggers duplicate transmissions. Therefore, the least penalty of TOH-based coordination, denoted as mc_lpenalty$_{TOH}$, is generalized as (6).

$$\text{mc\_lpenalty}_{TOH} = I/C \times p \times \text{Failure}(I/C \times I \times \text{AF}, p) \quad (6)$$

Consider an example with $p = 0.7$, $R = 3$, $C = 3$, and $I = 2$, the results with different AF and batch size are listed in Table 1. To measure overhead per packet, cost per batch is further divided by $B$. The trade-off between cost and penalty is controlled by AF and the best overhead is marked by gray background. Take $B = 20$ for example, its overhead is the best when AF = 1 and overhead is $0.40733 = 6/20 + 0.10733$. Compared with coordination with overhearing (penalty = 0.249), penalty of coordination with standalone-ACK is much fewer with the price of large coordination cost. Because the cost is fixed for a batch, increasing the batch size (such as 80) can even the price.

**TABLE 1. The Cost, penalty and overhead for SACK with different batch sizes under different AF with $p = 0.7$, $R = 3$, $C = 3$, and $I = 2$.**

| AF | Cost | Penalty | Overhead | | |
|---|---|---|---|---|---|
| | | | $B$=20 | $B$=40 | $B$=80 |
| 0 | 0 | 0.46667 | 0.46667 | 0.46667 | 0.46667 |
| 1 | 6 | 0.10733 | 0.40733 | 0.25733 | 0.18233 |
| 2 | 12 | 0.02240 | 0.62240 | 0.32240 | 0.17240 |
| 3 | 18 | 0.00378 | 0.90378 | 0.45378 | 0.22878 |
| 4 | 24 | 0.00087 | 1.20087 | 0.60087 | 0.30087 |
| 5 | 30 | 0.00018 | 1.50018 | 0.75018 | 0.37518 |

### 4) COORDINATION BY OVERHEARING WITH SUPPORT OF STANDALONE -ACKS

To overcome the loss of overhearing in multi-channel environments, it is straightforward to combine overhearing-based coordination with standalone-ACKs. The simplified behavior of overhearing-based coordination supported by standalone-ACKs, denoted as T&S, is listed below:

1) The src broadcasts a batch of $B$ packets until each packet is received by at least one relay.
2) Each relay broadcasts AF standalone-ACKs to all interfaces; the parameter AF is defined by the frequency of standalone-ACKs.
3) Relays forward received but unacknowledged packets to the dst starting from $r_R$ to $r_1$.
4) When overhearing a packet sent by higher priority relays, lower priority relays treat it as an acknowledgement and relinquish to send the same packet.

This method is a variant of coordination by SACKs. It has the same cost but relays can receive one extra acknowledgement from overheard data packets. With the cost of $R \times \text{AF} \times I$ transmissions, the penalty for T&S, denoted as penalty$_{T\&S}$ and listed in (7).

$$\text{penalty}_{T\&S} = I/C \times p \times \text{Failure}(I/C \times (I \times \text{AF} + 1), p) \quad (7)$$

Consider the same example of Table 1 ($p = 0.7$, $R = 3$, $C = 3$, and $I = 2$), the results with different AF and batch size are listed in Table 2. With an additional acknowledgement in the data packet, the combination is better than that of SACKs.

**TABLE 2.** The cost, penalty, and overheads for T&S with different batch sizes under different AF with $P = 0.7$, $R = 3$, $C = 3$, and $I = 2$.

| AF | Cost | Penalty | Overhead | | |
|---|---|---|---|---|---|
| | | | $B$=20 | $B$=40 | $B$=80 |
| 0 | 0 | 0.46667 | 0.24889 | 0.24889 | 0.24889 |
| 1 | 6 | 0.10733 | 0.34200 | 0.19200 | 0.11700 |
| 2 | 12 | 0.02240 | 0.60966 | 0.30966 | 0.15966 |
| 3 | 18 | 0.00378 | 0.90202 | 0.45202 | 0.22702 |
| 4 | 24 | 0.00087 | 1.20034 | 0.60034 | 0.30034 |
| 5 | 30 | 0.00018 | 1.50008 | 0.75008 | 0.37508 |

#### 5) COORDINATION BY NETWORK CODING

The simplified behavior of coordination by network coding is listed below:

1) The src broadcasts coded packets from a batch of $B$ packets.
2) Relays broadcast coded packets.
3) After receiving $B$ linearly independent packets, dst can decode the batch of $B$ packets and sends a batch-ACK back to src using TR.
4) The src starts the next batch when receiving or overhearing the batch-ACK.
5) Relays stop broadcasting the current batch when receiving or overhearing the batch-ACK of the current batch or data packets of the next batch.

The cost of this method is the batch-ACK from dst to src, which is (hop count)/$p$. Depending on node density, the cost is proportional to $R/p$ and similar to the cost of TP. However, the analysis of penalties from redundant data packets is the main challenge. Relays are unaware of the status of downstream relays, they keep generating redundant data packets until receiving the signal to stop. The signal comes from three sources: 1) batch-ACK, 2) data packets of a new batch and 3) acknowledgements in header (CCACK). When the duty of a relay is done (its downstream relay has collected sufficient coded packets), there is no guarantee when a batch-ACK or data packets of a new batch arrive; its penalty is therefore unpredictable.

With CCACK, relays can obtain special vectors in header for acknowledgements by overhearing data packets and stop generating redundant packets as soon as possible. However, in multi-channel environment, some relays can never collect acknowledgements from downstream relays with CCACK. They need to wait for batch-ACK or data packets of the next batch as the traditional way. If standalone-ACKs are applied, there is an additional cost of $R \times \text{AF} \times I$ as other methods; and the overall cost is (hop count)/$p + R \times \text{AF} \times I$, which is the sum of two methods and is too expensive.

Consequently, coordination by network coding has difficulties in multi-channel environments. Without CCACK, it has similar cost as TP and unpredictable penalties. With CCACK, it is similar to T&S, which relies on overhearing and needs additional standalone-ACKs for multi-channel environments. As a result, network coding is not included in the comparison.

### E. PRACTICAL ISSUES

The above analysis shows ideal behaviors of different coordination methods. Some issues caused by practical implementations are not included. There are two major issues: 1) the last mile issue for TOH and 2) the ACK-timing issue for SACK.

The last mile issue is how to get acknowledgements from destination nodes. TOH collects acknowledgements by overhearing data packets of downstream relays; however, a destination does not broadcast data packets. In other words, relays close to the destination must collect acknowledgements by other methods rather than overhearing. In practice, destination nodes periodically broadcast standalone-ACKs to resolve the last mile issue. Therefore, pure TOH does not exist and its practical behavior is closer to T&S.

For the analysis of SACK and T&S, low priority relays are assumed to wait for acknowledgements from high priority relays; however, acknowledgements from high priority relays may not arrive in time. In practice, relays send additional standalone-ACKs by a timer or packet counter. Standalone-ACKs are sent periodically controlled by a timer or relays trigger a standalone-ACK for a predefined number of data packets received. Relays set timers for sending data packets; if the timer expires before these extra acknowledgements arrive, those acknowledgements are useless and unworthy.

Consequently, the last mile issue forces TOH to include SACK and becomes unscalable. The ACK-timing issue weakens the effect of SACK. These practical concerns are shown in simulations in later sections.

**TABLE 3.** Comparisons of different coordination methods in terms of cost and penalty.

| method | cost per batch | penalty per packet |
|---|---|---|
| TOH | 0 | $I/C \times p \times \text{Failure}(I/C, p)$ |
| SACK | $R \times \text{AF} \times I$ | $I/C \times p \times \text{Failure}(I/C \times I \times \text{AF}, p)$ |
| T&S | $R \times \text{AF} \times I$ | $I/C \times p \times \text{Failure}(I/C \times (I \times \text{AF} + 1), p)$ |
| TP | $R/p$ | 0 |
| Network coding | (hop cout)/p | unpredictable |

### F. COMPARISON AND DISCUSSION

The results of the overhead analysis are summarized in Table 3. To make it easier to observe, three figures are presented to show trends with specific parameters. In Fig. 8, the overheads of different coordination methods with $p = 0.7$ are shown under different $I/C$ settings. In Fig. 9 and Fig. 10, we fixed the $I/C = 2/2$ and $I/C = 2/3$ to observe overhead under different PDR settings. In all figures, $R = 3$ and $B = 40$; and AF is set to be the value with the best overhead.

Fig. 8 shows the trend of overheads with increasing interfaces and channels. Both TOH and SACK have high overheads. The overhead of TOH increases when interface numbers are insufficient (ex: $I/C = 2/3$). SACK suffers from the scalability issue; the overheads increase as the interfaces increase. T&S is the combination of TOH and SACK; it has lower overheads but still suffers from the scalability issue. When the interface number is high, the cost to reduce the penalty by standalone-ACKs is too high and unworthy. As a

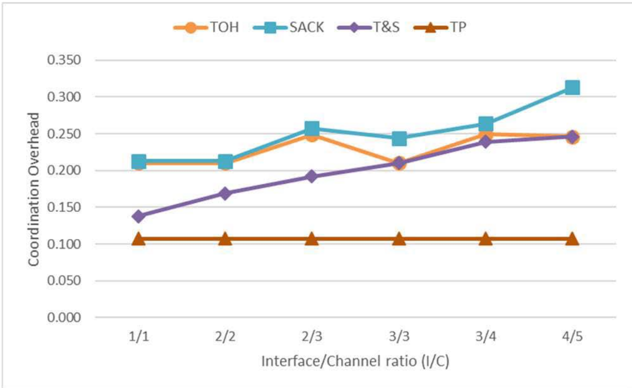


**FIGURE 8.** **Overheads for different coordination methods under different interface/channel settings.**


**FIGURE 9.** **Overheads for different coordination methods under different PDR settings (I/C = 2/2).**

result, overheads of TOH and T&S are similar when $I/C =$ 3/3 or higher. With consistent overheads, coordination by TP outperforms among all coordination methods especially in MC environments.

Fig. 9 and Fig. 10 shows the trend of overheads under different PDRs. Fig. 9 shows the results of sufficient interfaces ($I/C$ = 2/2); the results reveal that reducing duplicates by extra standalone-ACKs is unworthy especially when PDR is extremely high or low as shown. When PDR is high, collecting acknowledgements by overhearing is sufficient. When PDR is low, only one relay receives the data packet and coordination is not required. Therefore, overheads of TOH and T&S are similar when PDR = 30% and 90%.

Fig. 10 shows the results of insufficient interfaces ($I/C$ = 2/3); overheads of TOH increase with PDRs. Without sufficient interfaces, some relays receive data packets but cannot collect acknowledgements by overhearing. When PDRs increase, those relays have a higher probability to receive packets and trigger duplicate transmissions. In other words, TOH needs the support of standalone-ACKs when interface number is insufficient, which is also observed in Fig. 8.

According to the above analysis, TOH needs the support from standalone-ACKs for three conditions: 1) last mile issue, 2) interface number is insufficient and 3) near middle PDRs. However, the combined T&S suffers from the scalability


**FIGURE 10.** **Overheads for different coordination methods under different PDR settings (I/C = 2/3).**

issue. When interface number increases, the cost to reduce penalty by standalone-ACKs is too high and unworthy. This makes TOH limited to low interface and channel numbers such as $I/C$ = 1/1 or 2/2. For scenarios with more interfaces, the efficiency of control packets is essential.

TP outperforms in two aspects: low overheads and high scalability. The key is to utilize control packets effectively. Consider it as a game for all $R$ relays to collect information from high priority relays. The best solution is $R$ transmissions because each relay has to reveal its information at least once. With token passing from high to low priority relays, relays can deliver aggregated information and the overall cost is $R/p$ transmissions. For a relay $r_i$, all information of high priority relays (from $r_{i+2}$ to $r_R$) is collected by the acknowledgement of its immediate higher relay $r_{i+1}$. Without delivering acknowledgements in order, there is no guarantee to obtain the aggregated information and more standalone-ACKs are required.

Consequently, coordination methods that rely on overhearing, such as TOH, T&S and network coding with CCACK, need to include standalone-ACKs for the last mile issue and the condition with insufficient interfaces. However, standalone-ACKs induce high cost and suffer from the scalability issue; these methods are therefore limited to low interface/channel numbers. Coordination by TP can produce consistent and low overheads for all interface/channel settings by utilizing control packets more effectively. All these findings convince us that TP is the most potential coordination method for MC environments. The challenges and required refinements to apply TP to MC environment are presented in the next section

## IV. THE MULTI-CHANNEL TOKEN PASSING

The previous analysis demonstrates that token pass is the most promising OR coordination solution for MC environments. Therefore, this paper proposed the multi-channel token passing (MCTP) to resolve the hazards of token passing in MC environments. MCTP handles two major challenges: 1) how to ensure the token passing path and 2) how to adjust the token rate to reach high throughput for dynamic traffic.

The details of the challenges and the proposed solutions are described in the following subsections.

### A. THE CRITERION FOR THE MULTI-CHANNEL TOKEN PASSING PATH

In the TP, a token passing path from destination to source must be found; it allows tokens with piggybacked acknowledgements to flow from the destination back to the source. While a token passes through the token passing path strictly from high to low priority relays, acknowledgements are distributed effectively with low costs. Finding a token passing path in the single channel environment is trivial; however, in MC environments there might be no common channel(s) among the consecutive relays. To ensure that there is always a common channel between any two nodes, there is a necessary condition for MCTP in MC environments: twice the number of interfaces is larger than the number of channels $C$ as shown in (8).

$$2 \times I > C \tag{8}$$

As each node is equipped with $I$ interfaces and $C$ orthogonal channels are available, there are $2 \times I$ interfaces assigned to $C$ channels for any two relays. According to the pigeonhole principle, at least two interfaces are assigned to the same channel if $2 \times I > C$. Consider an example of parameters $C = 3$ and $I = 2$. For any two relays, there are $2 \times 2 = 4$ interfaces assigned to 3 channels and two interfaces must be assigned to the same channel. In other words, there is a link between relays $\mathrm{r_i}$ and $\mathrm{r_{i+1}}$ and a token passing path always exists if $2 \times I > C$. If the minimum number of interfaces is met (e.g.: 3 channels and 2 interfaces), a token passing path is guaranteed and coordination by TP can work properly. Other conditions when the minimum interface is not met (e.g.: 3 channels and 1 interface), can be ignored because low density is not suitable for OR.

### B. DYNAMIC ADJUSTMENT OF THE TOKEN RATE

In coordination by TP, tokens are generated by the destination of each flow. A relay is allowed to send all the unacknowledged packets, while it receives a token. Token rate is the frequency of the token generation and is highly related to the frequency for relays to send packets and clear their queues. An appropriate token rate is essential to high throughput. For a low token rate with a high traffic arrival rate, queues have a high possibility to be full and therefore drop the incoming packets. However, for a high token rate with a low traffic arrival rate, the overall system performance degrades for high overheads. With multiple flows, traffic arrival rates vary from flow to flow especially in the MC environment. A unified and predefined token rate cannot meet the requirements and variations of all flows. Therefore, adjusting the token rate dynamically is essential for coordination by TP in the MC environments.

To adjust the token rate dynamically for a flow, relays first estimate the minimum token rate according to local information: such as observed throughput, queue length etc. Then, the minimum token rates of relays are calculated and included in the header of data packets and passed through relays to the destination of the flow. Notably, the exchange of these network condition information brings no extra cost. Finally, after collecting the information from relays, the destination is easy to figure out the end-to-end capacity and select a token rate to satisfy all relays. Details are described as follows.

#### 1) ESTIMATE THE MINIMUM TOKEN RATE

The token rate and the traffic arrival rate are the primary two factors for the change of queue length. In addition, the link quality also makes minor effects on it. The idea to estimate the minimum (i.e., lower bound) token rate for each relay is to find the rate to fill their queues. Relays collect packets in a queue and forward them to downstream relays. Then, packets in the queue are removed after receiving acknowledgements included in tokens from downstream relays. Ideally, a token should arrive before the queue is full for perfect link environments. In other words, the ideal minimum token rate of relay $r_i$, denoted as $tr_{im}(r_i)$ is the rate to fill the queue as shown in (9). However, for practical situations, the queue may not clean up after receiving a token due to packet loss (unacknowledged packets) for an imperfect link. Therefore, depending on PDRs, a fraction of packets, denoted as $UA_i$, are still unacknowledged and remain in the queue. The queue length is reduced to $(1-UA)$ and the minimum token rate of relay $r_i$, denoted as $\mathrm{tr_m}(r_i)$, is therefore estimated as (10).

$$tr_{im}(r_i) = \text{Throughput}/q_i \tag{9}$$

$$tr_m(r_i) = \text{Throughput}/((q_i) \times (1 - UA_i)) \tag{10}$$

Relays can estimate the minimum token rate with (10) by monitoring local throughput and $UA$s. Then, the minimum token rates of relays are obtained and piggybacked in the header of data packets and flow to the destination for dynamic token rate setting.

#### 2) DYNAMICALLY ADJUST THE TOKEN RATE

The idea to adjust the token rate dynamically is to observe the current load and set a slightly higher token rate. The destination collects all the relays' loads of the flow and selects the highest token rate as the base rate so that the rate can satisfy all relays. Finally, set a token rate of the flow f, denoted as $tr(f)$, to the base rate multiplied by a control factor, denoted as $\alpha$, as shown in (11). Assume that there are $m$ relays in the flow. The $\alpha$ controls the balance between the maximum throughput achievable (due to higher extra cost) and the speed to reach the maximum throughput. After the token rate is updated, relays observe new throughput and reply to the destination. As the token rate is estimated based on throughput and the token rate is always set higher than current load, the throughput will eventually reach the maximum throughput and reduce the cost of tokens.

$$tr(f) = max\{\mathrm{tr_m}(r_i), i = 1..m\} \times \alpha \tag{11}$$

The proposed dynamic token rate adjust mechanism has the following two advantages:

1) Quick detection of congestion: The traditional congestion detection scheme is usually through the observations of packet loss; however, the reason for packet loss not only comes from congestion, link condition is another primary factor. However, link quality sometimes varies dynamically thus making the congestion detection more difficult or longer time to confirm. Different from the traditional congestion detection schemes, the proposed token rate control scheme is able to determine the congestion accurately through the observation of queue length and throughput.
2) Fast convergence to maximum throughput: The adjustment of the token rate is multiplicatively increasing and thus shortens the time to converge to the maximum throughput. However, the lack of global view of multiple flows, the proposed rate control scheme cannot guarantee the fairness of multiple flows. How to impose a global view to the token rate control scheme will be the future work of our study.

## V. PERFORMANCE EVALUATION

Simulations are made to verify both the analysis in the previous section and the performance of various OR coordination methods under MC environments. The simulated coordination methods include: MCTP, SOAR and TR; SOAR is implemented and adjusted to represent TOH and T&S. The routing metrics for OR and TR are EAX and ETX, respectively. All the simulations are implemented and simulated with the simulation platform – Qualnet. This paper adds custom routing protocols in Qualnet and observes their performance. Behavior of wireless environments (such as interference between devices) and WiFi protocols are simulated by Qualnet. The results of network capacity and overheads are demonstrated to reveal the performance of different coordination methods. Details of the simulation assumptions and results are described below.

### A. SIMULATION SETTINGS

The simulated environment is an area of $4 \times 4 = 16$ square grids with one node in each grid. Grid size is 200(m) $\times$ 200(m) and the position of the node in each grid is randomly distributed. Two flows are simulated in the grid scenario; the first one flows from the upper left corner to the lower right corner and the second one flows from the upper right corner to the lower left corner. Each of the demonstrated simulation results is an average of 20 instances. The same instance is given to all methods by setting the same seed to ensure that they are compared under the same condition and the same routes. Each simulated instance starts from sending probes to obtain link PDRs and finds routes by ETX/EAX. After routes are determined, traffic comes in and the program starts collecting statistics.

SOAR [5] is a classic OR coordination method based on timer and overhearing; it also sends additional standalone-ACK to deal with unsuccessful overhearing; the frequency of standalone-ACKs is dynamic and based on the packet counter. In the simulation, destination and relays trigger standalone-ACKs for every 5 and 10 data packets received respectively. Therefore, the behavior of SOAR can represent T&S and is denoted as T&S(rly). Regarding TOH, the implementation of TOH must send standalone-ACKs at destination nodes for the last mile issue. To simulate TOH, this paper further adjusted SOAR by preventing standalone-ACKs at relays. As a result, SOAR with the standalone-ACKs only from destination can represent TOH and is denoted as T&S(dst).

To observe overheads under different interface/channel settings, four different settings are included: single interface and single channel ($I/C = 1/1$), dual interfaces and channels ($I/C = 2/2$), triple interfaces and channels ($I/C = 3/3$) and two interfaces and three channels ($I/C = 2/3$). Including the setting of $I/C = 2/3$ is mainly to demonstrate the influence of reduced chance of overhearing to different coordination methods of OR because that is the key of OR.

OR coordination methods are compared by capacity and overheads. Capacity is the sum of all flow throughputs and is the main metric to evaluate performance. Overhead is further divided into two categories: cost and penalty; where cost/penalty is determined by the number of control/duplicate packets required per packet arrived at destination. In other words, it divides the number of control/duplicate packets by the number packets arriving at destination. Consequently, a good coordination method should reach high capacity and low overhead as well. The detail of the simulated parameters is listed in Table 4.

**TABLE 4.** Simulation settings.

| General simulation parameters | |
|---|---|
| PHY | 802.11b |
| Power | 15 dBm (32mW) |
| Data rate | 2Mbps |
| Fading model | Rayleigh |
| Number of instances | 20 |
| Link-quality parameters | |
| Probe size | 1000 Bytes |
| Number of probes | 100 |
| Interval between probes | 10 Sec |
| Traffic parameters | |
| Traffic type | CBR |
| Packet size | 1024 Bytes |
| Interval between packets | 0.01 Sec |
| Duration | 600 Sec |

### B. RESULTS AND DISCUSSION

#### 1) PERFORMANCE UNDER DIFFERENT I/C SETTINGS

Fig. 11 shows the performance obtained by different coordination methods under various $I/C$ settings. Corresponding

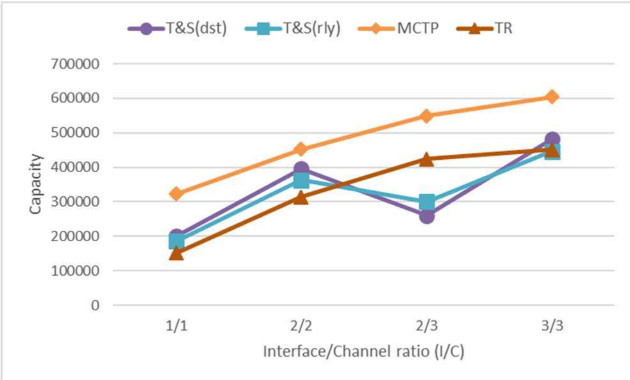


**FIGURE 11.** Capacity comparison under different interface/channel settings.

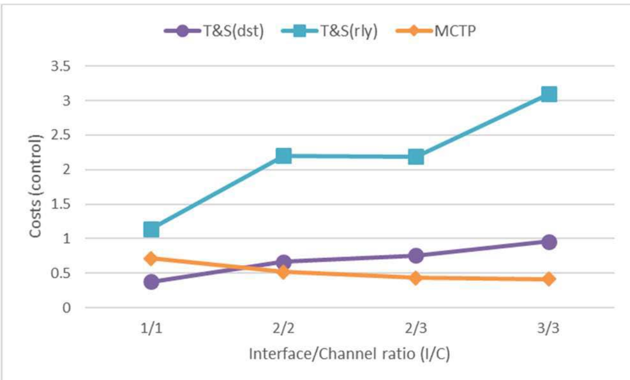


**FIGURE 12.** Cost comparison under different interface/channel settings.

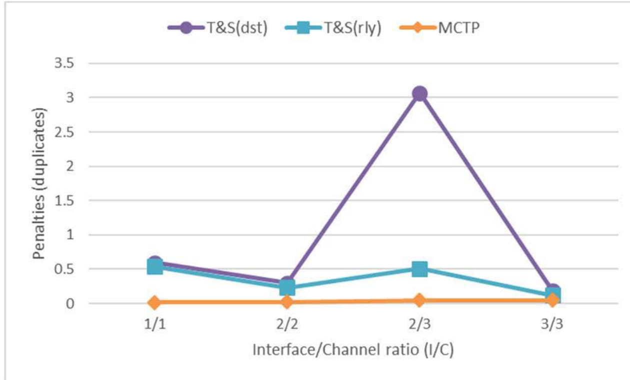


**FIGURE 13.** Penalty comparison under different interface/channel settings.

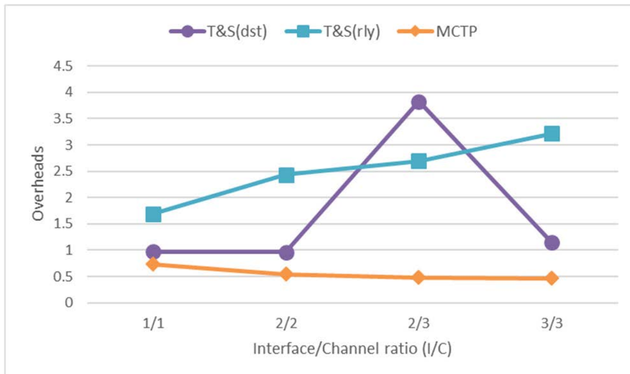


**FIGURE 14.** Overhead comparison under different interface/channel settings.

cost and duplicates are shown in Fig. 12 and Fig. 13. Overheads of Fig. 14 is the sum of Fig. 12 and Fig. 13. As shown in Fig. 11, both MCTP and TR are immune to reduced chance of overhearing under $I/C = 2/3$. In Fig. 12, cost for additional standalone-ACKs of T&S(rly) is high and increases with the number of interfaces and channels. T&S(dst) sends additional standalone-ACKs only at destination nodes; therefore, it has a minor scalability issue while compared to T&S(rly). Fig. 13 shows that T&S(dst) has high duplicates under $I/C = 2/3$ and leads to a poor performance. Though T&S(rly) can reduce duplicates with extra standalone-ACKs, the overall overhead is still high as shown in Fig. 14.

According to previous analysis, TOH needs the support from standalone-ACKs for the last mile issue and the condition of insufficient interfaces. Therefore, T&S(dst) has the scalability issue in Fig. 12 and high duplicates under $I/C = 2/3$ in Fig. 13 as expected. T&S(rly) is expected to reduce duplicates with standalone-ACKs in previous analysis; however, the ACK-timing issue weakens the effect of standalone-ACKs and makes them useless and unworthy. T&S(rly) therefore has low penalties in Fig. 13 and high costs in Fig. 12.

Regarding MCTP, it induces low costs and low penalties as expected. Because of consistent and lower overheads in MC environments, coordination by MCTP outperforms from $I/C = 1/1$ to 3/3. Regarding the challenging scenario of $I/C = 2/3$, where nodes may not overhear each other, it remains solid because the coordination relies on token passing rather than overhearing. As the criterion for the token passing path ($2 \times I > C$) is met, coordination by MCTP performs well in MC environments.

### 2) PERFORMANCE UNDER DIFFERENT PDR ENVIRONMENTS

To observe performance under different PDR environments, interface and channel number is set to $I/C = 2/2$ and the grid size is increased from 200 to 300. As a larger grid size refers to a longer distance between nodes, a larger grid size refers to a lower density and smaller PDR. If the grid size is too small, source nodes can reach destination nodes directly; if the grid size is too large, the wireless network is sometimes disconnected. Based on the test of sending probes in the simulated environment, it is observed that link PDR of 200(m) is about 90%, link PDR of 300(m) is about 70% and link of PDR of 400(m) drops to 30%. Therefore, only grid size of 200, 250 and 300 are considered and scenarios with smaller or larger grid size are excluded. Fig. 15 shows the performances obtained by different coordination methods under various grid sizes. Corresponding costs and penalties are shown in Fig. 16 and Fig. 17. Overheads in Fig. 18 are the sum of cost and penalty in Fig. 16 and Fig. 17, respectively.

Based on previous analysis, reducing duplicates by extra standalone-ACKs is not worthy when PDR is extremely high

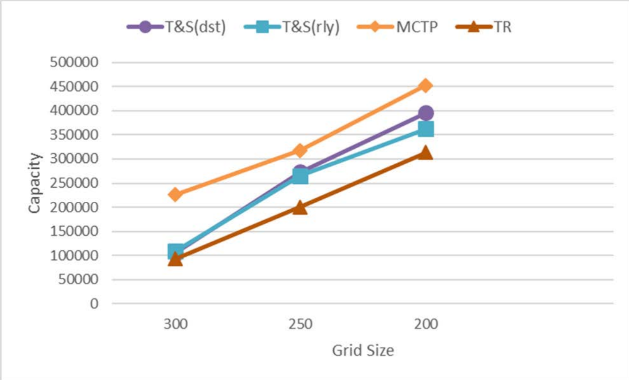


**FIGURE 15.** Capacity comparison under different PDR environments.

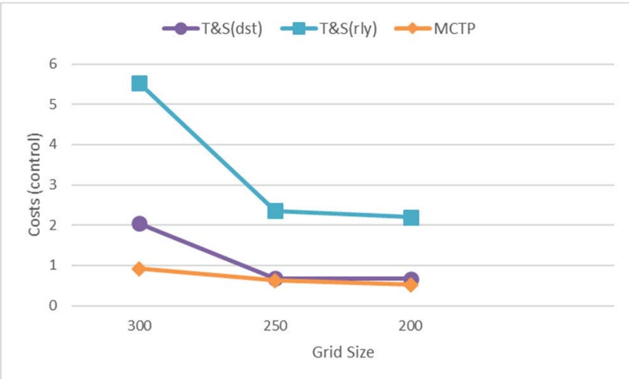


**FIGURE 16.** Cost comparison under different PDR environments.

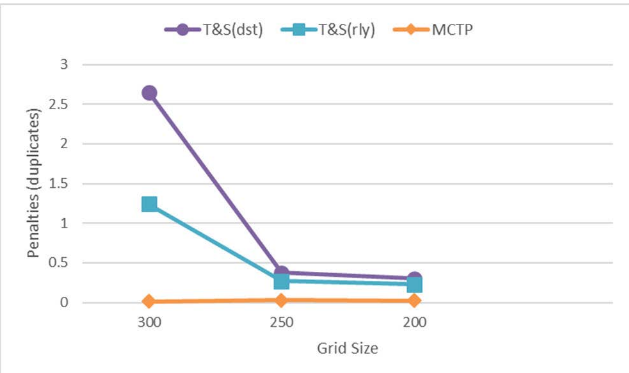


**FIGURE 17.** Penalty comparison under different PDR environments.

or low. Therefore, T&S(dst) performs better than T&S(rly) with smaller grid size in Fig. 15. It is observed in Fig. 17 that penalties of T&S(dst) are as low as T&S(rly) as expected when the grid size is 200 and 250 (better PDR). When the grid size reaches 300, penalties of T&S(dst) increase dramatically. Though the penalty can be reduced by T&S(rly), the cost is too high in Fig.16 and Fig. 18 and therefore unworthy.

#### 3) EFFECT OF DYNAMIC TOKEN RATE

To evaluate the effect of dynamic token rate in MCTP, its performance is compared with fixed token rate in Fig. 19. In the comparison, destinations generate tokens with a predefined interval, including 100, 200, 400 and 800 milliseconds.

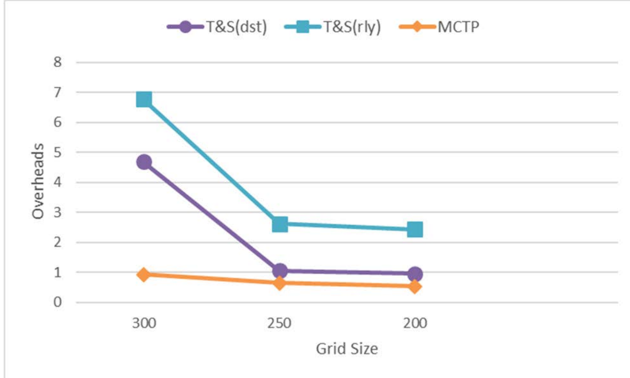


**FIGURE 18.** Overhead comparison under different PDR environments.

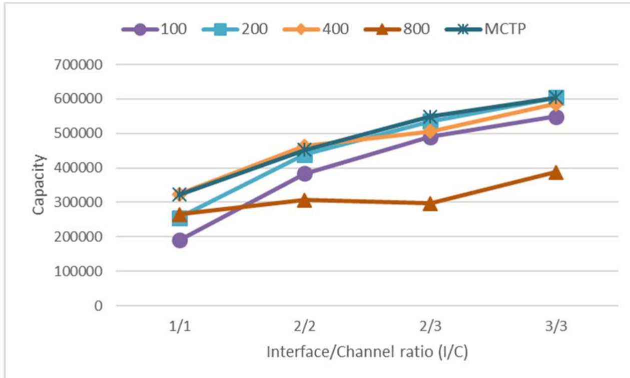


**FIGURE 19.** Capacity comparison under different token interval.

The results showed that picking a proper token interval is crucial. If the interval is set too small (such as 100), the overhead is high. In contrast, if the interval is set too large (such as 800), the token rate cannot saturate the flow. Both conditions lead to bad performances. The results also show that a fixed token interval cannot meet all scenarios. Token interval of 400 is suitable for $I/C = 1/1$ and 2/2. When a larger capacity is available in $I/C = 2/3$ and 3/3, a smaller token interval of 200 is better. MCTP outperforms for all scenarios by setting a proper token interval automatically and dynamically as shown in Fig. 19.

## VI. CONCLUSION

To apply OR to the MC environments, this paper has reviewed current OR coordination methods and built a two-hop simplified scenario to analyze their overheads. The theoretical analysis shows that OR coordination by overhearing (including network coding with CCACK) has high penalties and is unscalable to multi-channel environments. It induces high penalties for the last mile issue (collecting ACKs from destination nodes) and conditions with insufficient interfaces (such $I/C = 2/3$). To handle these conditions, it has to include standalone-ACKs and therefore causes the scalability issue as well.

Coordination by token passing is potential because it can utilize control packets effectively by sending aggregated information from destination to source. To apply token passing to OR in MC environments, this paper proposed MCTP to resolve the hazards of ensuring the token passing path and adjusting token rates dynamically for multiple flows. Both analyzed and simulation results reveal that MCTP outperforms other methods with low and consistent overheads; and it is scalable with increasing interface and channel numbers, which makes it a prototype for the future OR research in MC environments.

## REFERENCES


[1] C.-J. Hsu, H.-I. Liu, and W. K. G. Seah, "Opportunistic routing—A review and the challenges ahead," *Comput. Netw.*, vol. 55, no. 15, pp. 3592–3603, Oct. 2011.

[2] A. Boukerche and A. Darehshoorzadeh, "Opportunistic routing in wireless networks: Models, algorithms, and classifications," *ACM Comput. Surv.*, vol. 47, no. 2, pp. 1–36, Jan. 2015.

[3] N. Chakchouk, "A survey on opportunistic routing in wireless communication networks," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2214–2241, Mar. 2015.

[4] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, Oct. 2005, pp. 133–144.

[5] E. Rozner, J. Seshadri, Y. Mehta, and L. Qiu, "SOAR: Simple opportunistic adaptive routing protocol for wireless mesh networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 12, pp. 1622–1635, Dec. 2009.

[6] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[7] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[8] S. Kafaie, Y. Chen, O. A. Dobre, and M. H. Ahmed, "Joint inter-flow network coding and opportunistic routing in multi-hop wireless mesh networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1014–1035, Jan. 2018.

[9] Y. Yan, B. Zhang, J. Zheng, and J. Ma, "CORE: A coding-aware opportunistic routing mechanism for wireless mesh networks," *IEEE Wireless Commun.*, vol. 17, no. 3, pp. 96–103, Jun. 2010.

[10] S. Kafaie, Y. Chen, M. H. Ahmed, and O. A. Dobre, "FlexONC: Joint cooperative forwarding and network coding with precise encoding conditions," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7262–7277, Aug. 2017.

[11] M. K. Han, A. Bhartia, L. Qiu, and E. Rozner, "O3: Optimized overlay-based opportunistic routing," in *Proc. 12th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, May 2011, pp. 1–11.

[12] K. Chung, Y.-C. Chou, and W. Liao, "CAOR: Coding-aware opportunistic routing in wireless ad hoc networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 136–140.

[13] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*, Aug. 2007, pp. 169–180.

[14] D. Koutsonikolas, C.-C. Wang, and Y. C. Hu, "CCACK: Efficient network coding based opportunistic routing through cumulative coded acknowledgments," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[15] Y. Lin, B. Li, and B. Liang, "CodeOR: Opportunistic routing in wireless mesh networks with segmented network coding," in *Proc. IEEE Int. Conf. Netw. Protocols*, Oct. 2008, pp. 13–22.

[16] Y. Lin, B. Liang, and B. Li, "SlideOR: Online opportunistic network coding in wireless mesh networks," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–5.

[17] Q. Xiang, H. Zhang, J. Wang, G. Xing, S. Lin, and X. Liu, "On optimal diversity in network-coding-based routing in wireless networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 765–773.

[18] C.-J. Hsu and H.-I. Liu, "Source rate control for opportunistic routing," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–5.

[19] C.-J. Hsu, H.-I. Liu, and W. Seah, "Economy: A duplicate free opportunistic routing," in *Proc. 6th Int. Conf. Mobile Technol., Appl. Syst.*, Sep. 2009, pp. 1–6.

[20] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. 9th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Sep. 2003, pp. 134–146.

[21] Z. Zhong and S. Nelakuditi, "On the efficacy of opportunistic routing," in *Proc. 4th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw.*, Jun. 2007, pp. 441–450.

[22] C.-J. Hsu and H.-I. Liu, "Route selection for opportunistic routing in multi-channel scenario," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 6294–6299.

[23] Z. Huang, Y. Xu, and J. Pan, "TSOR: Thompson sampling-based opportunistic routing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7272–7285, Nov. 2021.

[24] N. Li, J. Yan, Z. Zhang, J.-F. Martinez-Ortega, and X. Yuan, "Geographical and topology control-based opportunistic routing for ad hoc networks," *IEEE Sensors J.*, vol. 21, no. 6, pp. 8691–8704, Mar. 2021.

[25] A. Hawbani, X. Wang, Y. Sharabi, A. Ghannami, H. Kuhlani, and S. Karmoshi, "LORA: Load-balanced opportunistic routing for asynchronous duty-cycled WSN," *IEEE Trans. Mobile Comput.*, vol. 18, no. 7, pp. 1601–1615, Jul. 2019.

[26] N.-T. Dinh, T. Gu, and Y. Kim, "Rendezvous cost-aware opportunistic routing in heterogeneous duty-cycled wireless sensor networks," *IEEE Access*, vol. 7, pp. 121825–121840, 2019.

[27] Q. Guan, F. Ji, Y. Liu, H. Yu, and W. Chen, "Distance-vector-based opportunistic routing for underwater acoustic sensor networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3831–3839, Apr. 2019.

[28] Z. Rahman, F. Hashim, M. F. A. Rasid, M. Othman, and K. A. Alezabi, "Normalized advancement based totally opportunistic routing algorithm with void detection and avoiding mechanism for underwater wireless sensor network," *IEEE Access*, vol. 8, pp. 67484–67500, 2020.

[29] R. W. L. Coutinho and A. Boukerche, "OMUS: Efficient opportunistic routing in multi-modal underwater sensor networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 5642–5655, Sep. 2021.



**CHE-JUNG HSU** received the B.S. and M.S. degrees in electrical engineering from Fu Jen Catholic University, Taiwan, in 2004 and 2007, respectively, and the Ph.D. degree from the Graduate Institute of Applied Science and Engineering, Fu Jen Catholic University, in 2022. His research interests include opportunistic routing, peer-to-peer, and wireless networks.



**HUEY-ING LIU** received the B.S. degree from Soochow University, Taiwan, in 1989, and the M.S. and Ph.D. degrees from the National Tsing Hua University, Taiwan, in 1991 and 1996, respectively. She is currently an Associate Professor with the Department of Electrical Engineering, Fu Jen Catholic University. Her main research interests include design, analysis, and implementation of network protocols, wireless networks, and the Internet of Things. Her latest research focuses on wireless sensor networks, opportunistic routing, and the Internet of Things. Currently, she is interested in applying machine learning to natural language processing and networks.