Analysis of General Network Coding Conditions and Design of a Free-Ride-Oriented Routing Metric

Bin Guo, *Student Member, IEEE*, Hongkun Li, *Student Member, IEEE*, Chi Zhou, *Senior Member, IEEE*, and Yu Cheng, *Senior Member, IEEE*

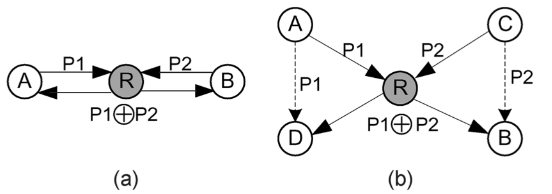


Fig. 1. Examples of information exchange with network coding. (a) Chain topology, in which coding is performed without opportunistic listening. (b) X topology, in which coding is performed with opportunistic listening.

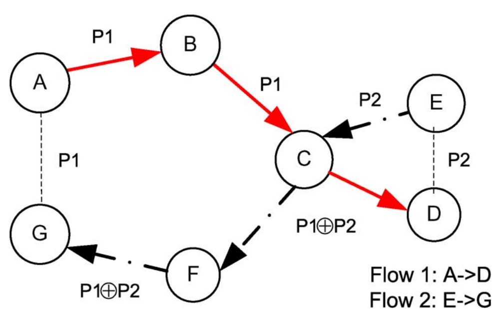


Fig. 2. Single coding node in the k-hop model [12].

***Abstract*—Coding-aware routing is an effective approach for incorporating network coding into practical network protocols. The main challenge in exploiting the network coding is how coding opportunities can accurately be identified and utilized. In this paper, we formally establish coding conditions for a very general scenario: Multiple coding nodes may exist along a path, and multiple flows may intersect at one node. Specifically, we systematically analyze possible coding scenarios and develop generalized coding conditions to ensure the decoding ability at the destinations. In particular, we develop an extended coding graph to make the conditions implementable. With the help of the extended coding graph, we examine the coding relationship between flows. Our coding conditions also include a coding flow selection procedure to guarantee the coding capability of each coding node, in which not only a native packet, but an encoded packet as well, could be involved. Furthermore we propose a novel coding-aware routing metric, the free-ride-oriented routing metric (FORM), with the objective of exploiting the coding opportunities so that a new flow can free ride on the existing traffic and can be supported with a smaller number of transmissions. NS-2 simulations are conducted to demonstrate the advantage of FORM and the proposed coding conditions over well-known COPE scheme in terms of throughput.**

***Index Terms*—Coding conditions, coding-aware routing, network coding , routing metric.**

Manuscript received January 29, 2010 ; revised August 18, 2010 and January 23, 2011; accepted February 09, 2011. Date of publication February 28, 2011; date of current version May 16, 2011. This work was supported in part by the Air Force Office of Scientific Research under Grant FA9550-09-0630 and by the National Science Foundation under Grant CNS-0832093. The review of this paper was coordinated by Prof. B. Hamdaoui.

The authors are with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: bguo2@iit.edu; hli55@iit.edu; zhou@iit.edu; cheng@iit.edu).

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

I. Introduction

NETWORK coding [1]–[3] has attracted much interest in wireless networks, because it could leverage the inherent broadcast characteristic of wireless channels and, thus, yield better throughput, even for unicast applications [4]–[11]. Fig. 1 illustrates the basic idea of network coding in wireless networks. Fig. 1(a) shows the *chain topology* , and Fig. 1(b) shows the X *topology*. In both cases, R is the common relay node for two flows. After receiving both P_1 and P_2, R encodes two packets and then broadcasts $P_1 \oplus P_2$. Upon the reception of the encoded packet, the destination can derive the expected native packet by decoding the encoded packet. The only difference is that A and B in the chain topology are the source nodes, whereas D and B perform "opportunistic listening" [6] in the X *topology*, i.e., D and B can overhear the transmission from A and C, respectively. The number of total transmissions is reduced from four to three in both scenarios.

Recently, coding-aware routing [12], [13] has been regarded as an effective approach to actively creat more coding opportunities. Its basic idea is that the coding opportunities are taken into account and are deliberately created during route discovery phase rather than passively waiting for the appearance of the coding opportunities. The main issues for efficient coding-aware routing are how to identify coding nodes and how to choose coding flows that traverse on each coding node. The solutions rely on accurately defined coding conditions by which the coding opportunities are discovered to reduce the total number of transmissions. Both coding structures shown in Fig. 1 are restricted within a two-hop region. In such structures, the coding opportunities occur within the region that includes only the relay node and its one-hop predecessors and successors.

Le *et al.* [12] exploit more coding opportunities in the k-hop ($k > 2$) model, which means that both opportunistic listening and decoding could occur several hops away from the relay node rather than one hop. The k-hop model is illustrated in Fig. 2. Suppose that two flows— $A \rightarrow D$ and $E \rightarrow G$ —intersect

Digital Object Identifier 10.1109/TVT.2011.2121097

at C and G, D can overhear the transmission from A, E, respectively. Although F cannot decode, it forwards the encoded packet to G, which can decode, because G has overheard P_1. This type of coding structure exploits coding opportunity at C, whereas the two-hop structure ignores it. The aforementioned coding structure only considers the scenario that two flows intersect at one node. However, in practice, each flow may intersect with other flows at multiple nodes, and multiple flows may intersect at one node. For these practical scenarios, we show in [14] that the coding condition indicated in [12] cannot guarantee the decoding ability at a destination node, and some preliminary discussions are provided in [14] to show that the coding condition in [12] should be augmented to accurately identify coding opportunities in practical scenarios.

The coding opportunities play a critical role in realizing the network-coding benefit, and the validity significantly depends on the effectiveness of the coding conditions. However, it is challenging to optimally define the coding conditions to discover all the coding opportunities in a given network, because the coding opportunities are obtained based on not only network topology but also traffic patterns. In this paper, we resort to an alternative suboptimal two-step approach. First, we generalize the coding conditions by focusing on one flow, with other flows given. This view is inspired by the fact in practice when a new flow joins the network, there normally exist other flows already in the network sharing the network resources. We try to discover a route for the new flow to efficiently exploit the network resources through coding opportunities. Second, we can iteratively recalculate the route for each existing flow that attempts to reach the global optimality for the entire network if the traffic engine control is allowed.

In particular, we systematically analyze possible coding scenarios and develop coding conditions for a k-hop coding structure model, in which the proposed conditions ensure that the destination of a coded flow can decode its packets. Our generalized coding conditions (GCCs) have the following two important properties.

- *Availability*. If the packets are encoded at the coding node, the destination can obtain the corresponding native packet, either by decoding the encoded packet itself or through some decoding-capable nodes that have retrieved the native packet for it.
- *Compatibility*. It guarantees that the encoding function of the coding nodes introduces no negative impact on other flows, which are overlooked by some related work. We will discuss the details in later sections.

Our coding conditions also include a coding flow selection procedure, which is more accurate and more effective, to guarantee the coding capability of each coding node, because the received packets at the coding node could be either native or encoded. According to the procedure, the coding node can decide which flows should be encoded together.

We further apply the general coding conditions to guide the design of our routing metric and protocol, which can be applied in static multihop wireless networks, e.g., wireless mesh backbone networks. In particular, we propose the novel coding-aware routing metric (CRM), i.e., free-ride-oriented routing metric (FORM), and its corresponding routing protocol. FORM quantitatively describes the "free-ride" transmission opportunities that a new flow can take advantage of through coding with the existing flows. The goal of our routing metric is to effectively explore coding opportunities and maximize the benefit of "free ride" to reduce the total number of transmissions and consequently increase the network throughput. The simulation results conducted in NS-2 show that our FORM-based routing protocol can achieve a higher throughput gain than the well-known COPE scheme.

In summary, this paper mainly contributes in the following three aspects.

- We develop GCCs, which are more accurate and more effective in detecting coding opportunities along a path in practical scenarios.
- Our coding conditions also provide a selection mechanism, by which the coding node can decide which flows should be encoded together.
- We design the novel CRM FORM to discover the path with more benefits riding on current network traffic patterns, utilizing the proposed coding conditions.

The rest of this paper is organized as follows. Section II reviews more related work. Section III describes the system model. In Section IV, we reinvestigate the coding condition developed in [12] and reveal its inefficiency when multiple coding nodes exist. In Section V, we develop the generalized conditions for multiple coding nodes along one specific path and a selection mechanism for choosing coding flows on each coding nodes. In Section VI, we develop the new CRM FORM and present the design of the FORM-based routing protocol. The simulations are conducted to evaluate the performance in Section VII. Section VIII gives the concluding remarks.

II. Related Work

Network coding was first proposed by Ahlswede *et al.* [1] , and it initially targeted at multicasting in traditional wired networks. The studies in [2] and [3] focus on the construction of efficient codes for network coding. A theoretical study on the efficiency of network coding in multihop wireless networks is given in [4] and [5]. Recently, there have been several works [6]–[11] on multiple unicast sessions by applying wireless network coding. These approaches are categorized as interflow network coding, which encodes multiple packets destined to different next hops and broadcasts them together. In particular, COPE [6] takes advantage of the broadcast nature of the wireless channel to perform *opportunistic listening* and *opportunistic coding* so that the coding opportunities are potentially explored to increase the throughput. In contrast to interflow network coding, intraflow network coding, i.e., routers mix packets that are headed to the same destination, is another category of network-coding application and is proved to provide reliability [24]. MORE [25], a state-of-the-art opportunistic routing scheme, exploits intraflow network coding by using random linear combinations while transmitting to give each transmitted packet unique information. In this manner, the possibility of useless duplicated transmissions is removed.

Network coding can also be applied in the physical layer, which is termed as physical-layer network coding (PNC) [26], [27]. Different from coding packets in bits, PNC performs coding in the physical wireless radio signal level by combining either the amplitudes or the phases of the signals. The receiver can decode the desired signal if it knows the other combined signal. In PNC, because two senders simultaneously transmit, the network throughput is further improved. However, the synchronization and decoding from interfered signal are still challenging issues. Similar to PNC, Katti *et al.* proposed MIXIT [28], which is a symbol-level network coding that combines opportunistic routing and symbol-level intraflow network coding. MIXIT performs random network coding across correct received symbols in different packets and provides end-to-end error recovery.

Coding-aware routing, which is another family of network-coding applications, is also proved to improve the network performance by actively creating coding opportunities. The authors of [15] and [16] proposed coding-aware routing schemes to maximize the overall coding opportunities in the whole network by using linear programming. However, their works are centralized approaches, which are difficult to implement in practice.

On the contrast, the authors of [12] and [13] proposed practical coding-aware routing mechanisms to actively create more coding opportunities. In the coding-aware routing protocol design, there are two essential aspects: 1) coding condition, which is used to effectively discover the coding opportunities, and 2) routing metric, which should carefully be designed to facilitate the routing protocol in a simple effective manner. In [17], the authors provide the condition for possible coding, but the condition is limited in a "two-hop" scenario. A novel coding condition is presented in [12], which can be applied in a k -hop coding structure aside from the popular "two-hop" scenario. However, the coding condition may be invalid when there are multiple intersecting nodes along a path. After reviewing some well-known routing metrics, we place different metrics into the following two categories—the coding-oblivious routing metric and the coding-aware routing metric (CRM)—and compare them to indicate the benefit of the CRM.

A. Coding-Oblivious Routing Metrics

1) Hop Count: The concept of the hop count metric is simple: Every link counts as one equal unit, independent of the quality or other characteristics of the link. The ease of implementation makes it the most widely used metric in multihop wireless networks.

2) ETX: The expected transmission count (ETX) [18] is defined as the number of transmissions required to successfully deliver a packet over a wireless link. This metric is based on bidirectional link loss ratios. Although ETX is based on the packet delivery ratio, which is directly related to the throughput, ETX does not try to route around congested links. Moreover, because the transmission rate of probe packets is typically low, it does not accurately reflect the loss rate of actual traffic.

3) ETT: The expected transmission time (ETT) [19] is an extension of ETX by considering different link routes or capacities. ETT is the expected time to successfully transmit a packet at the medium access control (MAC) layer. The ETT has the goal of incorporating link rate into its calculation, whereas the ETX may not obtain good performance. Furthermore, the weighted cumulative expected transmission time (WCETT) [19] is an extension of ETT and is designed for multichannel networks. This metric gives higher value for a path with a larger number of links that operate on the same channel, i.e., it favors channel diversity and low intraflow interference.

TABLE I
DEFINITIONS OF SOME TERMS USED IN THIS PAPER

Terms	Definition
Native packet	Original non-coded data packet.
Encoded packet	The packet after encoding operation (XOR).
Coding nodes	The nodes that satisfy the coding conditions and thus are able to perform the encoding function.
Coding flows	The flows that can be encoded together at one specific node.
Decoding-capable nodes	The nodes that are capable to decode the encoded packet and retrieve the corresponding native packet.
Acquisition nodes	The nodes that are not destinations but capable to obtain those packets which are useful to decode, either by overhearing or receiving.

B. Coding-Aware Routing Metrics

1) ECX: Ni *et al.* [15] proposed a new CRM, i.e., the expected number of coded transmissions (ECX), to capture the expected transmission for a successful exchange of packets between two nodes through an intermediate node for the chain topology. ECX is designed based on ETX, considering information exchange scenario for network coding. However, ECX only considers the information exchange scenario and does not fully exploit network-coding opportunities.

2) CRM: The CRM [12] facilitates the comparison between coding-possible and coding-impossible paths and is implemented in the proposed distributed coding-aware routing (DCAR) protocol. The key idea of CRM is to consider the queue length based on coding, i.e., if a new packet can be encoded with a previous packet m in the queue, it does not incur additional workload to the queue. In other words, if two flows with an average queue length Q_1 and Q_2 can be encoded together, the total effective queue length will be $\max(Q_1, Q_2)$. CRM corresponds to the expected number of transmissions to successfully transmit the existing and the new incoming packets.

III. System Model

Before delving into details of our coding conditions, we first define some important terms that are summarized in Table I and briefly introduce our system model.

We consider a stationary multihop wireless network, e.g., an infrastructure/backbone wireless mesh network [23], supporting multiple unicast sessions. Each node can overhear all the transmissions from its one-hop neighbors, and each node can obtain the information of the entire path for each flow it relays. The path information includes all the hops of the path and the neighbors of each node along the path. Furthermore, we define the encoding and decoding functions at each node as follows to properly leverage the coding opportunities.

— **Encoding**. The coding node will encode the ready-to-transmit packet with other nodes in its output queue based

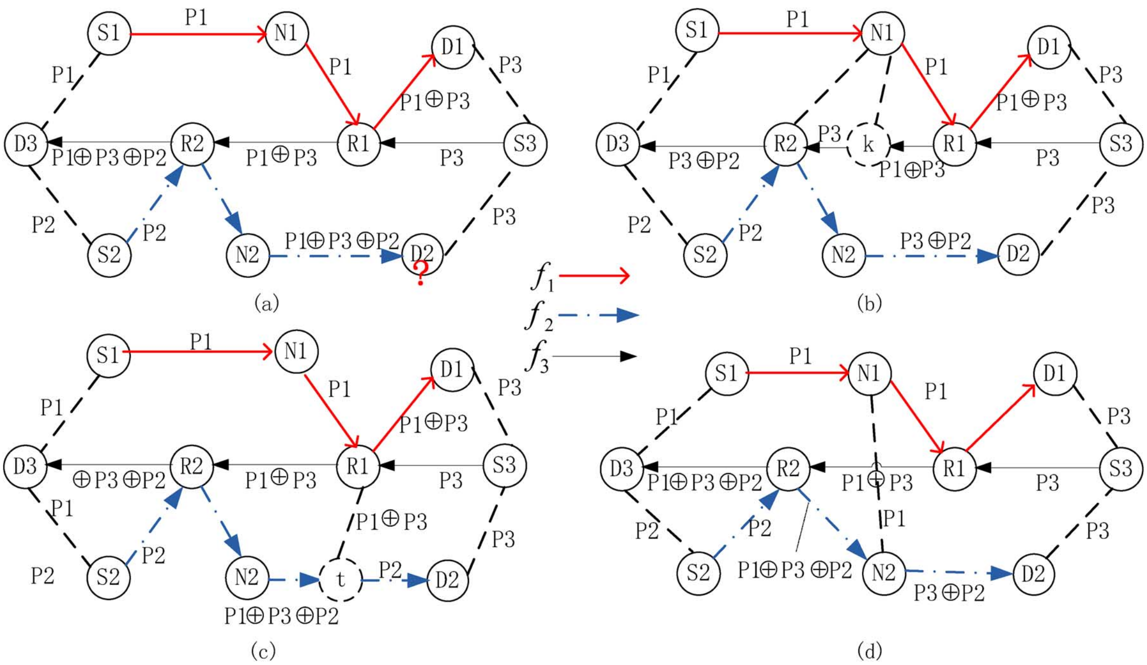


Fig. 3. Examples to illustrate multiple coding nodes along one path.

on the coding conditions. If there is no packet available to encode with it, the node will send the native packet.

— **Decoding**. In case that the node is the destination and receives an encoded packet, if it cannot decode, then it drops the packet; otherwise, it retrieves the expected native packet. On the other hand, if the node is not the destination and overhears an encoded packet, then it applies best effort decoding, which means decoding as much as possible.

The coding conditions play an important role in wireless network-coding applications. The validity of the coding node significantly depends on the effectiveness of the coding conditions. However, it is challenging to find all the coding nodes in a given network, because the coding opportunities are achieved based on not only network topology but on traffic patterns as well. Instead, we focus on one specific flow, with others as existing flows, and then examine the coding nodes along the considered flow. This approach is reasonable, because we aim at leveraging the coding opportunities to help design the routing protocol for one considered flow. On the other hand, every flow in the network can be treated as a considered flow, with others as background flows.

IV. Condition for a Single Coding Node

To better present our general coding conditions, we first reinvestigate the coding condition developed in [12], which assumes a single coding node along a path, and demonstrate its possible invalidity when multiple coding nodes exist.

Let a denote a node and $\mathcal{N}(a)$ denote the set of one-hop neighbors of node a. Let F denote the path of a flow f. If a belongs to the path F, we say that $a \in F$. Let $\mathcal{U}(a, F)$ and $\mathcal{D}(a, F)$ denote the set of all upstream and all downstream nodes of a on path F, respectively. For example, in Fig. 2, we have $\mathcal{U}(C, F_1) = \{A, B\}, \mathcal{U}(C, F_2) = \{E\}$; $\mathcal{D}(C, F_1) = \{D\}$, and $\mathcal{D}(C, F_2) = \{F, G\}$. G overhears P_1 from the previous transmissions of A, whereas D overhears P_2 from the previous transmissions of E. After C encodes P_1 and P_2 together, D and G will obtain the encoded packet $P_1 \oplus P_2$ and decode P_2 and P_1, respectively, because they have overheard the other packet that is involved in the encoded packet from previous transmissions. Therefore, the coding opportunity occurs at C. In general, supposing that the flow f intersects with f_x at node c, the coding conditions at c can be specified as follows.

1) There exists $d1 \in \mathcal{D}(c, F)$ such that $d1 \in \mathcal{U}(c, F_x)$ or $d1 \in \mathcal{N}(s2)$, $s2 \in \mathcal{U}(c, F_x)$.
2) There exists $d2 \in \mathcal{D}(c, F_x)$ such that $d2 \in \mathcal{U}(c, F)$ or $d2 \in \mathcal{N}(s1)$, $s1 \in \mathcal{U}(c, F)$.

We describe this coding condition as the *primary coding condition* (PCC). Note that $s1$, $s2$ and $d1$, $d2$ are not necessarily sources or destinations; they could be any node along the path. To illustrate the condition, let us take Fig. 2 for example. Obviously, we have $D \in \mathcal{D}(C, F_1)$ such that $D \in \mathcal{N}(E)$, $E \in \mathcal{U}(C, F_2)$ and also $G \in \mathcal{D}(C, F_2)$ such that $G \in \mathcal{N}(A)$, $A \in \mathcal{U}(C, F_1)$. According to the condition, C is a coding node. C sends out the encoded packet, and both D and G can obtain the intended native packets.

The PCC, indeed, relaxes the limitation of the *two-hop* coding structure and extends the chain and X topologies (as illustrated in Fig. 1) into a multihop context. However, such a coding condition might be invalid when there are multiple intersecting nodes along the path. Let us take Fig. 3(a) as an example. Suppose that two flows f_1 and f_2 already exist in a wireless network, following the paths F_1 and F_2, respectively. Then, a new flow f_3 arrives, following the path F_3. Based on the PCC, both $R1$ and $R2$ will be labeled as coding nodes, where R_1 may encode packets from paths F_1 and F_3 and R_2 may encode packets from paths F_2 and F_3. However, we find that $R2$ cannot be used as a coding node. When $R1$ performs the encoding function, $P_1 \oplus P_3$ is transmitted to both $D1$ and $R2$. $D1$ can decode the encoded packet, because it has overheard the packet P_3. Upon receiving the encoded packet $P_1 \oplus P_3$ along path F_3, $R2$ will further encode the packets $P_1 \oplus P_3$ and P_2 and send out the encoded packet $P_1 \oplus P_3 \oplus P_2$ to $D3$ and $N2$ along paths F_3 and F_2, respectively. There would be no problem at $D3$, because $D3$ can overhear both P_1 and P_2 from previous transmissions and thus obtain P_3 from the received $P_1 \oplus P_3 \oplus P_2$; however, when $D2$ receives the encoded packet $P_1 \oplus P_3 \oplus P_2$, it cannot decode its native packet P_2, although it has overheard P_3. In other words, we obtain a benefit for flow f_3 through the encoding function of $R1$ and $R2$ at the cost of failed transmission to the existing flow f_2. In this case, the node $R2$ should not be used as a coding node, and we describe this fact as the *false-coding effect*. Indeed, we must impose more constraints to avoid the false-coding effect.

V. Generalized Coding Conditions

According to the PCC, we can identify a set of coding nodes, which are defined as the *potential coding nodes*. In this section, we generalize the PCC by adding more constraints to avoid the false-coding effect. With the GCCs, we can identify the *true coding nodes* from the potential ones to perform the encoding function and ensure the availability of each packet at destinations.

A. Coding Conditions Generalization

We have shown that $R2$ is not a true coding node in the scenario shown in Fig. 3(a), because the encoded packet transmitted by $R2$ contains packets from other flows (i.e., f_1) incurred by coding at an upstream node. Such a scenario due to multiple coding nodes exceeds the decoding ability implied by the PCC. Therefore, we need to enhance the coding conditions to identify true coding nodes. In this section, we discuss the conditions for ensuring that the destination of each flow can obtain its own native packets.

Intuitively, if we can ensure that the coding operations at other coding nodes do not impact the availability associated with a potential coding node being considered, the potential node could then be a true coding node. We still consider these two potential coding nodes $R1$ and $R2$, as shown in Fig. 3(a) . There are three situations that the decoding ability of $R2$ can be ensured. The first situation is shown in Fig. 3(b), where there exists a *decoding-capable* node k between $R1$ and $R2$ (or $R2$ itself), which can overhear P_1 and, thus, can decode the encoded packets forwarded by $R1$. In this case, $R2$ can work well by first retrieving P_3 (possibly with the help of node k) from the coded packet $P_1 \oplus P_3$ and then encoding $P_3 \oplus P_2$ according to the PCC. The second situation is shown in Fig. 3(c) , where there exists an *acquisition* node downstream of $R2$ on F_2 , e.g., t (or the destination node $D2$), which can overhear the encoded packet $P_1 \oplus P_3$ and, thus, can retrieve P_2. In such a situation, the coding operation at R_2 according to the PCC is also valid, in the sense that the destination node $D2$ can decode P_2. The third situation is shown in Fig. 3(d), in which $D2$ can decode the intended packet, and hence, $R2$ is a true coding node. Suppose that there exists a downstream acquisition node $k \in \mathcal{D}(R2, F_2)$ that can obtain P_1. Then, k will decode the encoded packet $P_1 \oplus P_3 \oplus P_2$, extract $P_3 \oplus P_2$ (partially decoding), and forward $P_3 \oplus P_2$ to the next hop. Because $D2$ has overheard P_3 , it can obtain P_2 through decoding.

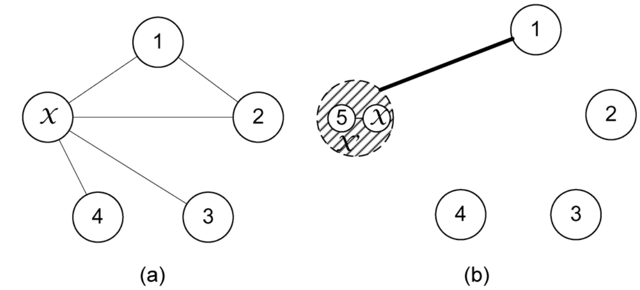


Fig. 4. Examples of a coding graph of a potential coding node p. (a) Native case. (b) Coded case.

According to all the aforementioned scenarios, the false-coding effect can be eliminated. To summarize, we generalize coding conditions for a potential coding node that consists of the following scenarios.

1) There exist upstream decode-capable nodes, which can extract the intended native packet for the node, on the considered flow.
2) There exist downstream acquisition nodes, which can overhear enough packets (either native or encoded) to decode, on other flows associated with the encoding function at this node.

B. Extended Coding Graph

The generalized conditions cannot directly be applied, because decode-capable nodes and downstream nodes may not be easily recognized. Therefore, we further investigate more details to make our conditions implementable. In practice, we would like each node to locally collect information and to individually determine whether it can play the role of a coding node. In [12], the use of a coding graph was proposed to identify whether a node is a potential coding node according to the PCC, but the coding graph cannot be directly applied to the general scenarios when a node may have incoming coded packets. In this section, we extend the coding graph to identify the true coding nodes.

We denote the flow under consideration as f_x, and there are multiple potential coding nodes along this flow. Let p denote a specific potential coding node, and we assume that p has one or more potential coding flows with f_x. A coding graph associated with a node is defined as an undirected graph, where each vertex represents a flow that traverses the node, and an edge between two vertices implies that these two flows can be encoded together. Based on the coding graph, we can figure out which flows should be encoded together by computing cliques in the graph. In [12], each node in the coding graph represents a native flow. However, when the flow under consideration is a coded flow, the approach is no longer applicable. Therefore, we extend the coding graph to address the concern of how we can qualify a node as a true coding node if it receives coded packets. In fact, the false-coding effect occurs when the incoming flow is a coded flow. For example, such a situation happens to $R2$ in Fig. 3(c) and (d) .

According to two scenarios of our conditions, we differentiate the following two cases on the coding graph: 1) the *native case*,

Flow ID	Downstream		Upstream	
	Node ID	Overheard flows	Node ID	Overheard flows
1	-	-	-	-
	...	...	...	...
	-	-	-	-
...	...	...	...	...
n	-	-	-	-
	...	...	...	...
	-	-	-	-

Fig. 5. Flow data structure on each node.

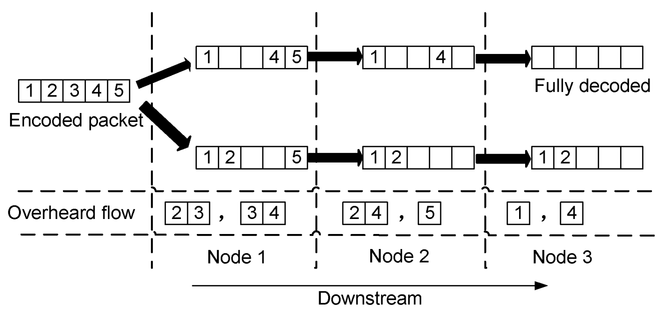


Fig. 6. Example to show the procedure of checking the decoding ability node by node.

shown in Fig. 4(a) and 2) the *coded case*, shown in Fig. 4(b). The first scenario can be realized by the native case, in which an edge between two vertices means that these two flows satisfy the PCC. The second scenario can be realized by the coded case, which implies that the flow under consideration is a coded flow. In the coded case, the vertices associated with the coded flow form a new vertex (e.g., x' in Fig. 4(b)), which is defined as a *coded vertex*. We also define the vertices inside a coded vertex as *interior vertices*, and similarly, the outside vertices are called *exterior vertices*. Note that, in the coded case, the flow under consideration is a coded flow, with other existing flows as native flows; hence, there is only one coded vertex with other exterior vertices. The new challenge here is to identify whether there is an edge between an exterior vertex and the coded vertex (e.g., the bold line between 1 and x' in Fig. 4(b)). Such a decision depends on the second statement of the GCCs.

The node resorts to the flow data structure (see Fig. 5) to locally solve this problem. The flow data structure records the information of each flow that traverses it. Each entry includes the flow ID, each downstream/upstream node on this flow, and the corresponding overheard flows. To cope with this issue in practice, each node obtains its one-hop neighbor information and the corresponding overheard flows piggybacked in the HELLO messages. The information collected could provide the path information during the routing discovery phase (we will discuss the details in the following section).

For each existing flow (an exterior vertex), the node accesses to the flow data structure and finds out the overheard flow information of each node on this flow. Recalling the second scenario of our conditions, we need to check whether there exist downstream acquisition nodes that can overhear enough packets (either native or encoded) to decode the coded flow (the coded vertex). To this end, we resort to a peeling-onion-like approach, and we show in Fig. 6 one example to better illustrate this case. The coded flow traverses the downstream path of the existing flow in the flow data structure to check whether it can be decoded (peeled) node by node (layer by layer). At each node, if there is no overheard flow that can decode, then the encoded flow goes to the next node. If there are several overheard flows that can decode, the coded flow is "peeled" as different new versions of coded flows (peeled coded flows) and then goes to the next node. The next node continues trying to "peel" each "peeled coded flow" by using its overheard flows. This procedure continues until the coded flow is "totally peeled," i.e., fully decoded. In this case, the existing flow can encode with the coded one, and consequently, there is an edge between the corresponding exterior vertex and the coded vertex. Note that both the number of involved nodes and the number of the overheard flows on each node are only few, and on each coding node, the available coding flows are also bounded by a small number [10]; hence, the overhead of this "peeling" procedure is insignificant. For example, $R2$ in Fig. 3 is a potential coding node; correspondingly, f_1, f_3 are the interior vertices, and f_2 is the exterior vertex. $R2$ will check the downstream nodes on f_2 and their relevant overheard flows. If there exists a node that can overhear the coded version of f_1 and f_3 [see Fig. 3(c)] or there exist nodes that can separately overhear f_1 and f_3 [see Fig. 3(d)], then the encoded flow can be decoded along the downstream path of f_2, and consequently, f_2 can join the coded flows.

Based on the extended coding graph, we propose the coding flows selection procedure. In coding graphs, either in the native or the coded case, if a subgraph is a clique, then vertices (flows) in this subgraph can be encoded all together. This procedure is analog to the maximum clique problem, which is NP-complete. In practice, however, it has been observed and proved in [6] and [10] that the number of maximum encoding flows at one node is only few in most cases; therefore, the computational overhead is insignificant. Furthermore, a flow may have several options to choose flows that can be encoded together, and we adopt opportunistic encoding to cope with this issue in implementation. For instance, we suppose that $\mathbb{C}$ is the set of true coding flows, and if $\mathbb{C}$ is null, then p is not a true coding node. In Fig. 4(a), $\mathbb{C} = \{\{f_1, f_2\}, f_3, f_4\}$ indicates that f_1, f_2, and f_x can be encoded together and that f_x can also be packed with f_3 or f_4. In practice, the node can encode P_x with $P_1 \oplus P_2$, P_3 , or P_4, depending on which packet is ready to transmit.

C. Discussions on Coding Conditions

Note that our coding conditions are sufficient conditions rather than optimal ones to find all the coding opportunities. The issue on how we can identify all the coding opportunities, i.e., to figure out the sufficient and necessary conditions, is still an open issue, even by using centralized approaches.

We aim at leveraging the coding conditions to facilitate the design of routing protocol, which requires acceptable complexity and distributed implementation in finding coding opportunities. We conservatively design the conditions to guarantee that any coded packet can be decoded and any destination can retrieve the native packet, and the price to be paid is that some coding opportunities may be missed. Although we cannot achieve the optimality, our approach can further be improved by iteratively recalculating the route for each existing flow to look for more coding opportunities, and we will discuss the details in the following section.

VI. Design of the Coding-Aware Routing Metric

In this section, we propose a new CRM, i.e., FORM, and develop the corresponding routing protocol. Although coding-aware routing actively exploits the coding opportunities, the path with more coding opportunities is not necessarily the path with better packet delivery performance. Because a coding-aware path may traverse more hops to reach the destination, it may suffer from interferences or bad link quality with higher probability.

A. FORM

We propose FORM to take advantage of the free-ride transmission opportunities through network coding. Suppose that a new flow is injected into a wireless network with several existing flows. If it is encoded with the existing flows, the packets could be transmitted along with the existing packets, i.e., the new flow can "free ride" on the existing flows. The network-coding gain in this scenario depends mainly on the existing traffic pattern rather than the link status. The goal of our routing metric is to utilize the coding opportunities discovered by the coding conditions and maximize the benefit of "free ride" to reduce the total number of transmissions and consequently increase the network throughput.

FORM consists of the following two parts: 1) *modified benefit*, which incorporates both the path length and the number of coding nodes along this path, and 2) *degree of free ride*, which indicates the abundance of coding opportunities of each path. Consider a set of available paths for the new flow, $\mathcal{F} = \{F_i, 1 \leq i \leq k\}$, where k is the total number of the available paths. Recalling the free-ride property, when a packet traverses along one path, it may encounter some coding nodes, on which it can be transmitted together with other packets. In other words, the coding nodes do not need extra transmissions for this packet. In this manner, one transmission is saved when the packet goes through one coding node.

1) Modified Benefit: First, we define the *benefit* $\beta(F_i)$ for path F_i as the number of coding nodes on path F_i. It denotes how many transmissions can be saved along this path for the new incoming packet. Moreover, we define the path length $H(F_i)$ as the number of hops when the packet traverses from the source to the destination along F_i. The minimum length among the set of all the available paths is

$$L_{\min} = \min_{1 \leq i \leq k} H(F_i). \tag{1}$$

It is observed that the packet needs additional transmissions along the path, which is longer than $L_{\min}$. This condition implies that the benefit from coding achieved on that path might be weakened. Based on this fact, we define *modified benefit* to measure the actual number of transmissions that can be saved on each path as follows:

$$\beta_m(F_i) = \beta(F_i) - [H(F_i) - L_{\min}]. \tag{2}$$

Indeed, $\beta_m(F_i)$ considers both the benefit brought by the coding and the cost due to the longer transmission path. When $\beta(F_i) > (H(F_i) - L_{\min})$, $\beta_m(F_i)$ is positive, the benefit from coding is larger than the cost spent over a longer path, which means that it is worth to take the longer path to achieve benefit from coding. When $\beta(F_i) < (H(F_i) - L_{\min})$, $\beta_m(F_i)$ is negative, which means that the benefit is not enough to take the longer path or there is no coding opportunity at all, and the path with $L_{\min}$ will be the choice. $\beta_m(F_i) = 0$ means that the benefit is exactly counteracted by the longer path. In this case, the path with $L_{\min}$ will be chosen. From this perspective, the modified benefit considers both coding-possible and coding-impossible paths.

2) Degree of Free Ride: Now, we discuss the abundance of the coding opportunities on each coding node and each path. Once a new packet arrives at a coding node, the coding node has to choose the proper packets from its buffer to encode with the new packet, according to the coding condition. However, its buffer may not have such packets for encoding if the flow rate (or bandwidth) of the existing flows are lower than the flow rate of the incoming flow or the life time of the existing flows are not sufficiently long to support the coding. In this case, the coding node has to initiate a new transmission for this packet. In other words, the packet cannot free ride on the ongoing transmissions. To avoid this situation, we should choose those coding nodes with abundant encoding resources, i.e., there are more encoding packets in the buffer to ensure that the new packet has the opportunity to free ride on the ongoing flows. We define the *degree of free ride* $(DegFR)$ to indicate the available encoding resource of each path. Supposing that there are m coding nodes along the path F_i, the number of available encoding packets in each coding node's buffer is EP_j, $1 \leq j \leq m$, and $DegFR$ of path F_i is defined as

$$DegFR(F_i) = \begin{cases} 1, & \text{no coding nodes exist} \\ \sum_{1 \leq j \leq m} EP_j, & \text{otherwise.} \end{cases} \tag{3}$$

Finally, we combine these two aspects and define FORM as follows:

$$FORM(F_i) = DegFR(F_i) * \beta_m(F_i). \tag{4}$$

Intuitively, FORM reflects the level of benefit that the new flow could obtain from the existing traffic pattern and the current network resource. Note that the FORM value of one path may be negative, which means that this path consumes more network resources than the path with the shortest length. The path with larger FORM value will be selected for the incoming flow to maximize the benefit of "free ride" on the existing flows and, hence, to increase the throughput of the entire network and the new flow as well. If all the paths are coding-impossible paths, the shortest path has the largest FORM value, which is 0, and the shortest path will be chosen.

Remarks: We note the following remarks based on the performance of FORM.

1) FORM is genaral for both coding-possible and coding-impossible paths. In the scenario of all coding-impossible paths, FORM reduces to a hop-count based metric.
2) The source node calculates the FORM of each path according to the general coding conditions and selects the best path with the largest FORM value. Because the general coding conditions are based on the information of the entire path, when the routing request is sent to the destination, the destination cannot see which set of nodes are coding nodes. When the reply packet traverses back to the source along the reverse path, the information about coding opportunities would be collected at the source. The details will be discussed in the next section.
3) FORM is based on the hop-count metric, and it can be extended with other coding-oblivious metrics. We use the ETX metric as an example to show how we can extend. We assume that the outgoing links of one node to its neighbors have a similar ETX value, which is a reasonable assumption when the nodes are located in the similar environment and are close to each other. In fact, it is complicated to measure how many transmissions could be saved by coding if the outgoing links have different levels of link quality, because the encoded packets are broadcast and will undergo different links. Similarly, the ETX for the path F_i and the minimum ETX among all the potential paths are defined, respectively, as

$$ETX(F_i) = \sum_{l \in F_i} ETX(l) \tag{5}$$

$$e_{\min} = \min_{1 \le i \le k} ETX(F_i) \tag{6}$$

where l denotes the link on F_i. Correspondingly, the benefit and the modified benefit of path F_i are expressed as

$$\beta_{ETX}(F_i) = \sum_{j \in \mathbb{C}, j \in F_i} ETX(j) \tag{7}$$

$$\beta_{mETX}(F_i) = \beta(F_i) - \left[ETX(F_i) - e_{\min}\right]. \tag{8}$$

The modified benefit under the ETX metric denotes how many expected transmissions could be saved on one specific path compared with the minimum ETX value.

B. Routing Protocol Design

Our routing protocol is designed based mainly on the Dynamic Source Routing (DSR) protocol [29], which is a simple and efficient routing protocol designed for wireless networks. The routing protocols discover the routes, collect the information, and choose the best route based on the predefined metric. Moreover, every node learns more information to check whether it is a coding node and further provides information to make the routing decision. First, each node is required to store a list of neighbor information, including its one-hop neighbors and the corresponding overheard flow information. This information can be obtained by broadcasting hello messages. This technique has been used in several on-demand routing protocols, e.g., ad hoc on-demand distance vector (AODV) and DSR. In addition, similar to DCAR [12], every node must store the information for every existing flow that it processes, i.e., the entire path information for each flow or every node that the flow goes through and the length of the path. Upon collecting all this information, the source can make the routing decision. In this section, we discuss the design details, which can be summarized as the following four major steps.

. *Step 1: RREQ broadcasting*. When a new flow arrives, the source first broadcasts a route request (RREQ) message to initialize the route discovery. Several RREQs reach the intended destination through different paths, and RREQs will record every relay node as any source routing does and its neighbor information. Upon receiving a RREQ, the intermediate node will first check whether it has already received this RREQ message. If so, it discards the RREQ to prevent routing loop. Then, it will temporarily store the RREQ, append its who-can-overhear information into a RREQ, and rebroadcast the RREQ until the RREQ reaches the destination. Let us consider the illustrative example shown in Fig. 3(b). Suppose that f_1 and f_2 are two existing flows and that a new flow f_3 is coming. The RREQ message along the path $S3 \rightarrow R1 \rightarrow R2 \rightarrow D3$ is recorded as shown in Fig. 7(a) hop by hop. Note that the predecessor and successor nodes of each node along this path would be eliminated in their one-hop neighbors. For example, $R1$ is a one-hop neighbor of $S3$. When the RREQ is received by $R1$ from $S3$, it will be removed from $S3$'s one-hop neighbor list.

. *Step 2: Operations at the destination*. Once the destination receives the RREQs, it will reverse the hop sequence of the received path and insert the reversed path and the corresponding who-can-overhear information into the route reply (RREPs) messages, which are unicast back to the source node.

. *Step 3: Operations at the intermediate nodes*. Each intermediate node that receives the RREP message will perform the following operations.

- It will first compare the information contained in RREP and the information in the temporarily stored RREQ. If they match each other, this node would obtain the entire path information, i.e., the "node" and "one-hop neighbors" parts of RREQ and RREP, as illustrated in Fig. 7(a) and (b). If they do not match, the RREQ should be discarded.

- Then, it will compare the path information obtained from RREP with the stored flow data structure (shown in Fig. 5) for each existing flow to verify whether it is a coding node and which coding case (native or coded) to which it belongs according to the GCCs.
- If it is a coding node, it will update the RREP by inserting the coding-related information, i.e., coding flows and the available encoding packets in the buffer. For instance, when $R1$ receives the RREP and observes that the RREP matches the stored RREQ, it compares with the stored flow information. For example, in Fig. 3(b), $R1$ is a coding node, and f_1 is in its stored flow data structure. Then, it inserts the information that f_1 is a coding flow with the new one at $R1$. In addition, $R1$ counts the number of packets of f_1 in its output buffer, record it as EP_{f1}, and insert it into a RREP. Finally, $R1$ sends out the RREP on the reverse path back to the source by unicast.

. *Step 4: Routing decision made at the source*. Based on the received RREPs, the source node obtains several available paths F_i. On each available path, it calculates the FORM for each path and selects the path with the largest FORM value. The path selection procedure is illustrated in Algorithm 1.

Algorithm 1: Routing decision.

Input: F_i, $1 \le i \le k$ (retrieved from the received RREPs)

Output: Optimal path F^*

$F^* = \phi$; $FORM^* = 0$; $l_{\min} = \infty$;

for $i = 1; 1 \le i \le k; i++$ **do**

  Calculate $H(F_i)$;

  if $H(F_i) \le l_{\min}$ **then**

    $l_{\min} = H(F_i)$;

  end

end

for $i = 1; 1 \le i \le k; i++$ **do**

  Calculate $FORM(F_i)$;

  if $FORM(F_i) \ge FORM^*$ **then**

    $FORM^* = FORM(F_i)$;

    $F^* = F_i$;

  end

end

Return F^*;

RREQ		
Node	One-hop Neighbors	Overheard Flows
S3	D1, D2	f_1, f_2
R1	D1, N1	-
k	N1	f_1
R2	S2, N2	-
D3	**S1**, S2	f_1, f_2

(a)

RREP				
Node	One-hop Neighbors	Overheard Flows	Coding Flows	EP
D3	**S1**, S2	f_1, f_2	-	-
R2	S2, N2	-	f_2	EP_{f2}
k	N1	f_1	-	-
R1	D1, N1	-	f_1	EP_{f1}
S3	D1, D2	f_1, f_2	-	-

(b)

Fig. 7. Example of the data structures used in the routing protocol. (a) RREQ of the illustrative example. (b) RREP of the illustrative example.

C. Implementation Issues and Discussion

In this section, we discuss some other implementation issues involved in the FORM-based routing.

a) Broadcast mechanism: All the nodes are set to a promiscuous mode to leverage the broadcast characteristic of wireless medium. However, because the IEEE 802.11 broadcast mechanism lacks reliability in the absence of ACKs, we use the "pseudobroadcast" technique [6] to maintain the reliability. The technique piggybacks on the 802.11 unicast standard by inserting an additional header into the encoded packet and listing all the next hops of the packet. The MAC address of the encoded packet is set as one of the intended recipients, which will acknowledge the receipt by the standard 802.11 unicast procedure; other nodes listed in the XOR header will be acknowledged in a hop-by-hop manner [6].

b) Encoding and decoding: The coding node does not need to wait for the available coding packet if such a packet does not exist in its output queue. Instead, it encodes the packets in an opportunistic manner. When a packet is ready to transmit, the coding node will search in the output queue for the available packet to be encoded with the ready-to-transmit packet. If there is no such packet, then the packet is directly transmitted without coding. This way, the extra delay incurred by waiting the encoding opportunity can be avoided.

To perform the decoding function, each node maintains an *overhear buffer* to store the overheard packets, and also, each overheard packet is assigned with a *packet timeout*. When a packet is timed out or the buffer is full, the packet will be purged from the buffer. We assume large enough buffer and long-enough packet time out on each node, because it is difficult to estimate the delay of overheard packets in the k -hop coding structure. Let us revisit the example in Fig. 2 . When G overhears the packet P_1, it needs to store P_1 for a while, during which P_1 travels from $A \rightarrow B \rightarrow C$, and then, the encoded packet returns from $C \rightarrow F \rightarrow G$. The total delay is the summation of the delay of this packet on each link. However, it is difficult for G to estimate, because it depends on both link quality and queue status of each involved node.

c) Overhead analysis: We analyze the overhead of the proposed routing protocol compared with DSR and DCAR.

Basically, the overhead lies in the RREQ and RREP messages; therefore, we focus on the extra length of these routing discovery messages. First, we compare it with DSR, the most well-known source routing protocol. Similar to DSR, we have the hop limit $H_{\max}$ in the protocol; therefore, there are at most $H_{\max}$ entries in RREQ and RREP. Let $E(d)$ be the expected value of node degree. The average number of one-hop neighbors listed in RREQ and RREP for each entry is $E(d) - 2$. Note that we do not consider its predecessor and successor nodes. We use 2 bytes to represent the node's ID. In total, the extra length of the RREQ will not be larger than $2 \times H_{\max} \times (E(d) - 2)$, which is approximately dozens of bytes. In the RREP, the coding information occupies 2 B for each entry, because the number of coding flows is restricted, and the output buffer size is limited. Then, the extra length of RREP will not exceed $2 \times H_{\max} \times (E(d)-2) + 2 \times H_{\max} = 2 \times H_{\max} \times (E(d) - 1)$. The overhead that lies in RREQ and RREP is tolerable when the data packet is long enough. Compared to DCAR, the length of RREQ is similar; the extra length of RREP lies only in the additional information about coding, which takes 2 B, at most, for each entry. Hence, the total additional overhead will not exceed $2 \times H_{\max}$.

d) Protocol complexity: The complexity of the proposed routing protocol comes from three aspects. First, we need to identify those coding nodes to leverage the coding opportunities. We have discussed the coding conditions in the previous section. The more the information that each node obtains, the more the coding nodes that can be identified. However, the complexity will increase due to more information exchanges. In our protocol, the coding decision is made based on stored flows information and RREQ and RREP messages during the route establish phase. In this manner, the complexity is greatly reduced compared with the result with searching for the sufficient and necessary coding conditions. Second, we need to consider the encoding and decoding functions. Each node needs extra buffer to store the overheard packets, which occupies additional resources. Third, the control message overhead should be considered, which is associated with any on-demand routing protocols, as aforementioned.

e) Optimality: Our proposed FORM protocol works in a heuristic manner and is suboptimal in the sense that the new incoming flow is routed based on the existing traffic pattern. By using the tools developed in [30], we could formally analyze the optimality. Our routing protocol combines flooding-based route discovery and source routing, and the routing metric FORM is neither left isotonic nor right isotonic; therefore, our protocol cannot achieve optimality. See [30] for more details. As aforementioned in Section II, there have been several works [15], [16], [22] that explore the throughput gain through coding-aware routing by solving optimization problems; however, these approaches are centralized and are difficult to apply in practical protocol designs.

Although we do not obtain the optimal solution, the performance can be improved by recalculating the paths for existing flows. Once the new flow has entered the network, the traffic pattern is changed. The changed traffic pattern provides the existing flows, with the potential ability to have a better path, because each flow deserves a better route based on the changed traffic pattern. To this end, we apply the iterative approach to recalculate the routes for existing flows. Upon the arrival of the new flow, each existing flow recalculates its route, in turn, based on other flows. In this manner, the performance of the entire network may be improved. We show an example in Fig. 8 to demonstrate the validity of path reselection for the existing flows. Fig. 8(b) shows the result after recalculation. It is obtained that the total coding opportunities in the network increase, as long as flows 1 and 2 choose different paths. Indeed, the total throughput of the network can further be improved, and we will show the result in Section VII.

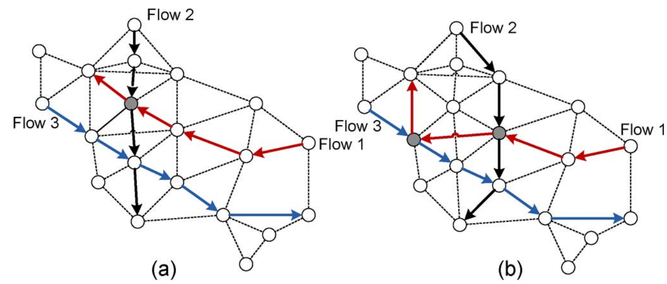


Fig. 8. Example to show the benefit of iteratively recalculating the route for existing flows. The network consists of 20 nodes, and there are three flows in total. The dashed line indicates the connectivity between nodes, and the gray node represents the coding node. (a) Flow 3 is a new flow, whereas flows 1 and 2 are existing flows. (b) After recalculating the routes for flows 1 and 2, the network achieves better performance in terms of more coding opportunities.

In practice, however, recalculating the route for each flow will result in extra overhead in terms of longer delay, because each route discovery procedure needs to exchange control messages. The other issue to be carefully considered is how frequent should each flow recalculate the route. To address these problems, we need to carefully incorporate these concerns into our protocol design, which will be considered as our future work.

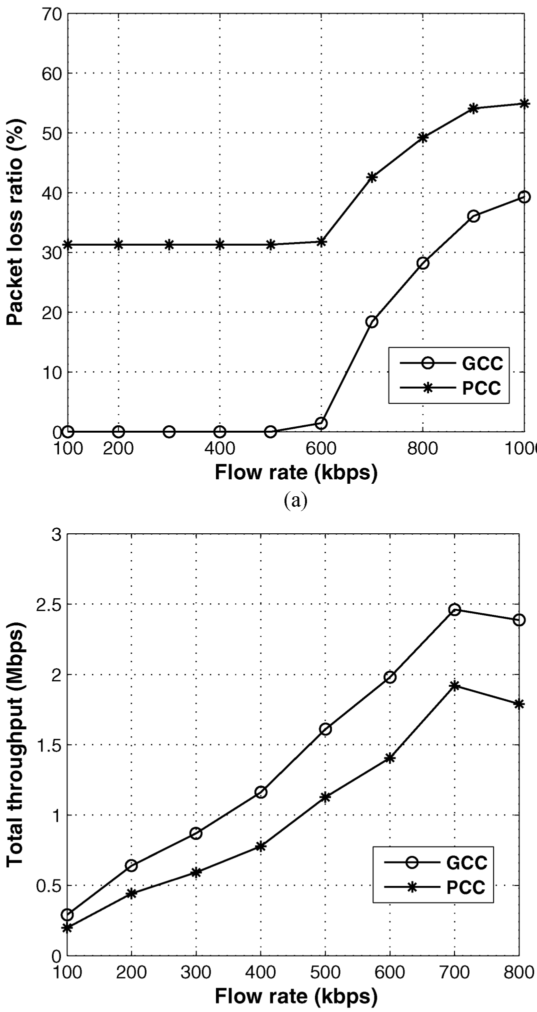
VII. Performance Evaluation

In this section, we present the performance evaluation using NS-2 simulation. The goals of our simulation include the following two aspects: 1) evaluating the validity of our general coding conditions compared with PCC and 2) quantifying the benefit of our new routing metric FORM under GCCs compared with the well-known COPE scheme. We conduct experiments on the illustrative topology to measure the benefit of our general coding conditions and to evaluate the performance of FORM using static grid and random topologies.

Throughout the experiments, all the nodes are set to the promiscuous mode, and the IEEE 802.11 standard is adopted as our MAC protocol. We use User Datagram Protocol (UDP) traffic sources, and all the flows are constant-bit-rate (CBR) flows, with a fixed packet size of 1000 B. The transmission range is set to 250 m, and the interference range is set to 550 m.

A. Benefit on the Illustrative Topology

To demonstrate the effectiveness of our generalized conditions, we adopt the packet loss ratio and the total throughput as



(a)

(b)

Fig. 9. Benefit of the general coding condition. (a) Comparison of the packet loss ratio. (b) Comparison of the total throughput.

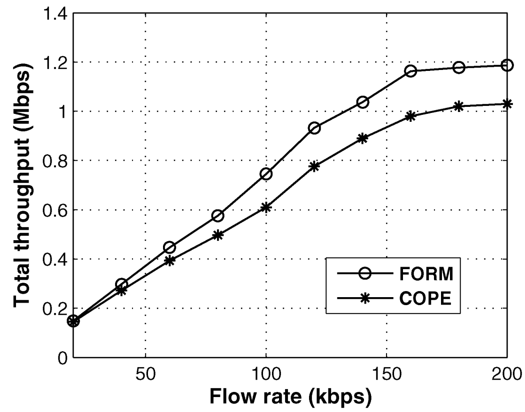
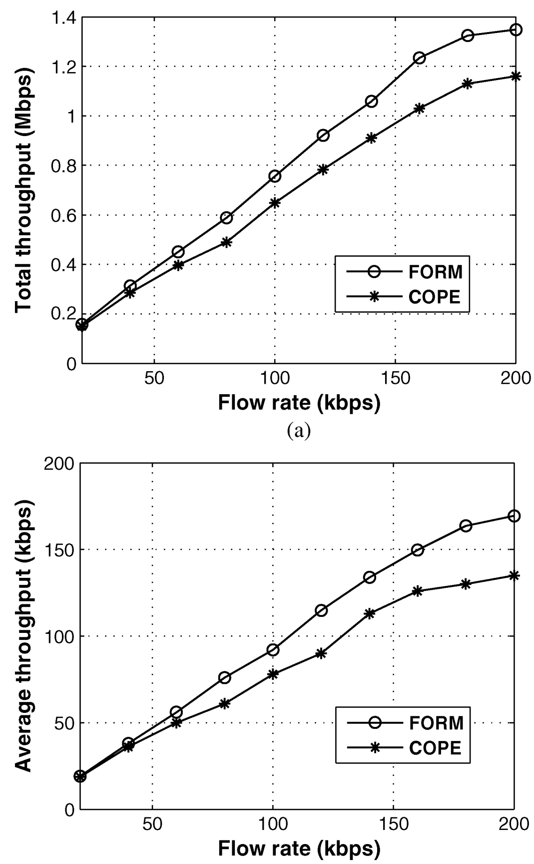


Fig. 10. Total throughput of the grid network (scenario 1).



(a)

(b)

Fig. 11. Throughput of the grid network (scenario 2). (a) Total throughput of the network. (b) Average end-to-end throughput of the new flows.

metrics and investigate the performance under both the PCC and the proposed GCCs. The topology is given in Fig. 3.

In the experiment, we first evaluate the impact of the offered load on the packet loss ratio. We vary the offered load to observe the performance under two coding condition cases. Fig. 9(a) shows the packet loss ratio versus the input data rate. It is clear that our generalized conditions significantly reduce the packet loss ratio. There are two possible reasons for the packet loss: 1) the discarding of packets that are not decoded and 2) the overflow in the buffer. $R2$ is a coding node according to PCC, but it is, in fact, an invalid node. Hence, $D2$ will drop all the received encoded packets, because $D2$ cannot decode them. In contrast, $R2$ will not be labeled as a coding node when applying our condition, and consequently, there is no encoding process at node $R2$. For GCC, only the overflow in the buffer could result in packet loss. Thus, the packet loss ratio of PCC is larger compared with GCC. The overflow in the buffer increases with the increment of the data rate; therefore, the packet loss ratio increases under both conditions. In addition, the two curves get closer when the data rate increases, because the congestion becomes the major factor for the packet loss in GCC with a higher data rate. In Fig. 9(b), we plot the total throughput both under PCC and GCC. Based on the result, it is illustrated that GCC achieves significant throughput gain, which is expected, because several packets are dropped under PCC with the invalid coding node.

B. Throughput on Grid and Random Topologies

We construct a 5×5 grid topology, in which the vertical and horizontal distances between two adjacent nodes are both 200 m. We randomly choose eight pairs of sources and destinations to run the simulations for both the FORM protocol and the COPE scheme. We conduct the simulations for two different scenarios. In scenario 1, eight flows arrive at the network at different time points with the same interval, and in scenario 2, we inject three flows into the network, with five flows already existing. Then, we evaluate the performances in both simulation scenarios.

We plot the result in Fig. 10. Based on the simulation, FORM gains obvious benefits over COPE. In general, the gain tends to be larger in accordance with higher offered load, and FORM can achieve at most more than 30% throughput gain. Fig. 11 shows the simulation results for the second scenario. We plot the total throughput achieved by FORM and COPE in Fig. 11(a). It is illustrated that FORM obtains higher throughput than COPE. The average end-to-end throughput of the new incoming flows is plotted in Fig. 11(b). Similar to the performance of the total throughput, FORM provides better performance than COPE because FORM optimally utilizes current traffic flows to reduce the number of transmissions, and consequently, the throughput is increased.

Then, we evaluate and compare the performance of FORM and COPE in a random topology. In this experiment, a 30-node topology is randomly generated, and we randomly pick eight pairs of nodes as the sources and destinations. We repeat the experiment multiple times, with the source-destination pairs reselected at each round, and estimate the average values and variances of the throughput. The 90% confidence intervals associated with each throughput estimation are also presented in Figs. 12 and 13 . We have run enough number of rounds to achieve a small confidence interval for accurate estimation results. Similar to the experiments in the grid topology, the simulations consist of two scenarios. In addition, we provide the result of MPATH-network-coding-aware multipath routing (NETCOD) [16] as the theoretical bound of coding-aware routing with COPE-type coding structures for comparison. The total throughput of the first scenario is plotted in Fig. 12 . In the experiment of the second scenario, the total throughput is plotted in Fig. 13(a), and the average end-to-end throughput of the new incoming flows is plotted in Fig. 13(b). Based on the simulation results, the performance is in line with the expectation. Because the random topology creates more coding opportunities, FORM certainly achieves better performance. However, MPATH-NETCOD produces better throughput performance than FORM, because MPATH-NETCOD is a centralized approach and is realized by the optimization formulation. MPATH-NETCOD integrates both multipath routing and scheduling and, hence, achieves better throughput gain. MPATH-NETCOD provides the theoretical basis on the additional gain of combining coding- and interference-aware routing in practical protocol design, which can be considered to be an interesting research direction.

Finally, we show the throughput performance after recalculating the route for each flow based on the first scenario in Fig. 14. When all the flows enter the network, each flow recalculates its route, in turn, based on the new traffic pattern, because the new traffic patterns provide each flow with the potentiality to achieve more coding opportunities and consequently produce a better performance in terms of higher throughput, as shown in Fig. 14.

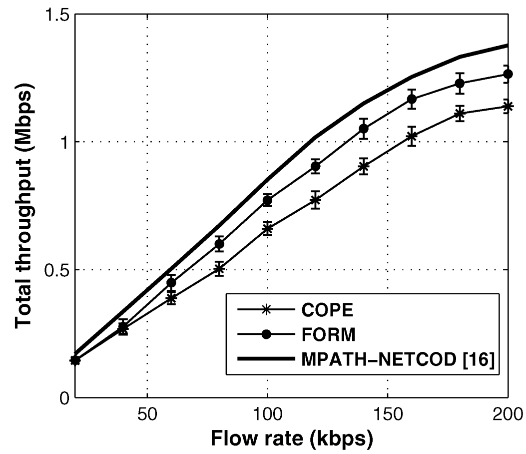
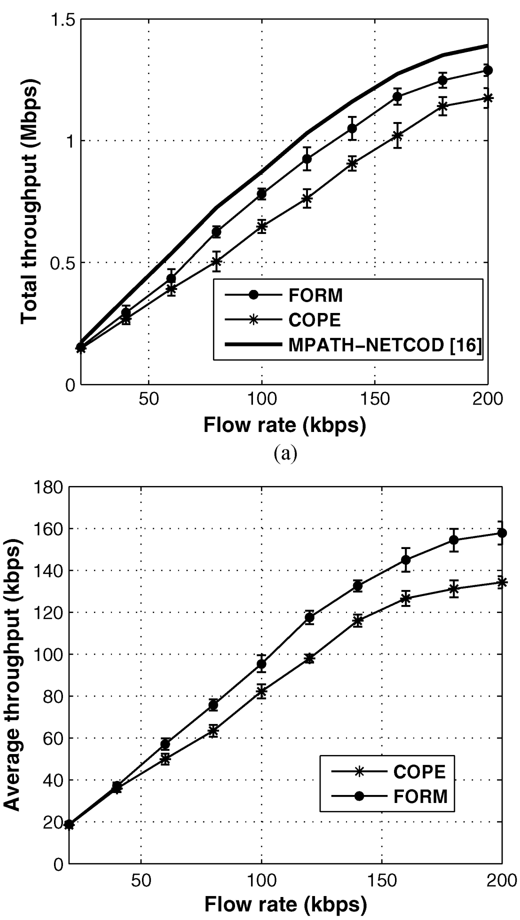


Fig. 12. Total throughput of the random network (scenario 1).



(a)

(b)

Fig. 13. Throughput of the random network (scenario 2). (a) Total throughput of the network. (b) Average end-to-end throughput of the new flows.

C. Delay Performance on the Random Topology

We also evaluate the performances of the average end-to-end packet delay of both FORM and COPE for scenario 1 in the aforementioned random topology. All the parameters are adopted as the same as we have illustrated. In Fig. 15, we provide the performance comparison with different flow rates. Not surprisingly, the FORM scheme yields higher packet delay than COPE. Our protocol is coding aware, and the new flow tries

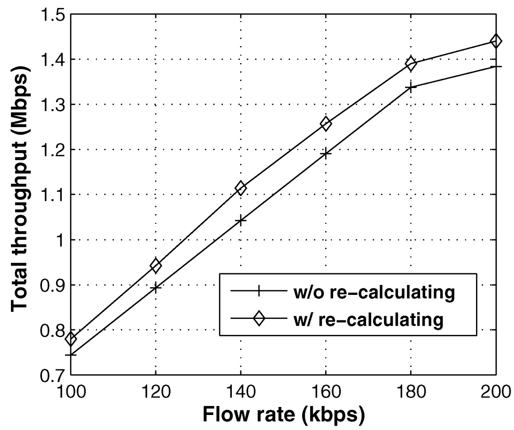


Fig. 14. Performance improvement by recalculating the routes for existing flows.

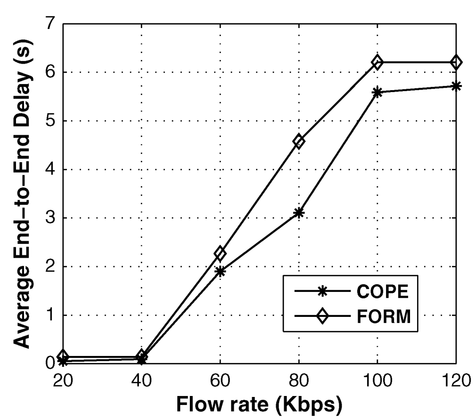


Fig. 15. Comparison of the average end-to-end delay.

to "free ride" on the existing flows. Based on our protocol, we prefer to find a path with more benefits of coding opportunities, i.e., more available coding packets in the buffer. However, a node with more available coding packets is more likely to suffer a longer queue delay. The results imply that our protocol does not consider load balancing in the routing discovery. How we can balance the traffic load and coding opportunities could be an interesting research direction in future works.

VIII. Conclusion

In this paper, we have first thoroughly analyzed multiple coding scenarios and proposed a general coding condition, considering the fact that one flow may intersect with multiple flows at different nodes. Our coding conditions allow exploiting more coding opportunities while eliminating the potential negative impact on other existing flows. Moreover, our coding conditions include a coding flow selection procedure to guarantee the coding capability at each coding node, considering that the received packets could be either native or coded. According to the procedure, the coding node can decide which flows should be encoded together. Then, we propose the novel routing metric FORM and design the corresponding routing protocol to improve the throughput performance of the wireless networks. Extensive evaluations under NS-2 illustrate that our FORM-based routing protocol under the proposed coding conditions achieves significant gain over the well-known COPE scheme. The simulation results also reveal that the integration of coding-aware routing with interference-aware routing and/or load balancing may lead to an improved performance.

References

[1] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 1204–1216, Jul. 2000.

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

[3] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Allerton*, 2003, pp. 40–49.

[4] D. S. Lun, M. Médard, and R. Koetter, "Efficient operation of wireless packet networks using network coding," in *Proc. IWCT*, 2005, pp. 1–5.

[5] A. Ramamoorthy, J. Shi, and R. Wesel, "On the capacity of network coding in wireless networks," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2878–2885, Aug. 2005.

[6] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *Proc. ACM SIGCOMM*, 2006, pp. 243–254.

[7] Z. Li and B. Li, "Network coding: The case for multiple unicast sessions," in *Proc. Allerton Conf.*, 2004, pp. 11–19.

[8] T. Ho and R. Koetter, "Online incremental network coding for multiple unicast," in *Proc. DIMACS*, 2005, pp. 1–14.

[9] T. Ho, Y.-H. Chang, and K. J. Han, "On constructive network coding for multiple unicasts," in *Proc. Allerton Conf.*, 2006, pp. 779–788.

[10] J. Le, J. C. S. Lui, and D. M. Chiu, "How many packets can we encode?—An analysis of practical wireless network coding ," in *Proc. IEEE INFOCOM*, 2008, pp. 1040–1048.

[11] S. Rayanchu, S. Sen, J. Wu, S. Banerjee, and S. Sengupta, "Loss-aware network coding for unicast wireless sessions: Design, implementation, and performance evaluation," in *Proc. ACM SIGMETRICS*, 2008, pp. 85–96.

[12] J. Le, J. C. S. Lui, and D. M. Chiu, "DCAR: Distributed coding-aware routing in wireless networks," in *Proc. IEEE ICDCS*, 2008, pp. 462–469.

[13] Y. Yan, B. Zhang, H. T. Mouftah, and J. Ma, "Practical coding-aware mechanism for opportunistic routing in wireless mesh networks ," in *Proc. IEEE ICC*, 2008, pp. 2871–2876.

[14] B. Guo, H. Li, C. Zhou, and Y. Cheng, "General network coding conditions in multihop wireless networks," in *Proc. IEEE ICC*, 2010, pp. 1–5.

[15] B. Ni, N. Santhapuri, Z. Zhong, and S. Nelakuditi, "Routing with opportunistically coded exchanges in wireless mesh networks," in *Proc. IEEE WiMesh*, 2006, pp. 157–159.

[16] S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding for unicast sessions: The case for coding-aware routing ," in *Proc. IEEE INFOCOM*, 2007, pp. 1028–1036.

[17] Q. Dong, J. Wu, W. Hu, and J. Crowcroft, "Practical network coding in wireless networks," in *Proc. Poster Session MobiCom*, 2007, pp. 306–309.

[18] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multihop wireless routing," in *Proc. ACM MOBICOM*, San Diego, CA, 2003, pp. 134–146.

[19] R. Draves, J. Padhye, and B. Zill, "Routing in multiradio, multihop wireless mesh networks," in *Proc. ACM MOBICOM*, 2004, pp. 114–128.

[20] B. Ni, N. Santhapuri, C. Gray, and S. Nelakuditi, "Selection of bit rate for wireless network coding," in *Proc. IEEE SECON*, 2008, pp. 1–6.

[21] Y. Yan, Z. Zhao, B. Zhang, H. T. Mouftah, and J. Ma, "Rate-adaptive coding-aware multiple-path routing for wireless mesh networks," in *Proc. IEEE GLOBECOM*, 2008, pp. 1–5.

[22] J. Zhang and Q. Zhang, "Cooperative network coding-aware routing for multirate wireless networks," in *Proc. IEEE INFOCOM*, 2009, pp. 181–189.

[23] I. F. Akyildiz and X. Dong, "A survey on wireless mesh networks," *IEEE Commun. Mag.*, vol. 43, no. 9, pp. S23–S30, Sep. 2005.

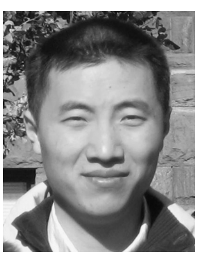
[24] C. Fragouli, D. Katabi, A. Markopoulou, M. Medard, and H. Rahul, "Wireless network coding: Opportunities and challenges," in *Proc. MILCOM*, 2007, pp. 1–8.

[25] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proc. ACM SIGCOMM*, 2007, pp. 168–180.

[26] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," in *Proc. ACM SIGCOMM*, 2007, pp. 397–408.
[27] S. Zhang, S. C. Liew, and P. P. Lam, "Physical-layer network coding," in *Proc. ACM MobiCom*, 2006, pp. 358–365.
[28] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," in *Proc. ACM SIGCOMM*, 2008, pp. 401–412.
[29] D. Johnson, Y. Hu, and D. Maltz, "The dynamic source routing protocol (DSR) for mobile ad hoc networks," IETF RFC 4728 [Online]. Available: http://www.ietf.org/rfc/rfc4728.txt 2009
[30] Y. Yang and J. Wang, "Design guidelines for routing metrics in multihop wireless networks," in *Proc. IEEE INFOCOM*, 2008, pp. 1615–1623.

Bin Guo (S'09) received the B.S. degree from Zhejiang University, Hangzhou, China, in 2004 and the M.S. degree in digital communications from Chalmers University of Technology, Gothenburg, Sweden, in 2008, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago.

His research interests include network coding applications in wireless networks and cooperative diversity techniques in wireless communication.



Hongkun Li (S'09) received the B.S. degree in information engineering from Zhejiang University, Hangzhou, China, in 2004 and the M.S. degree in electrical engineering from the Beijing University of Post and Telecommunications, Beijing, China, in 2007. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago.

His research interests include capacity analysis and resource allocation for multiradio multichannel wireless networks and routing protocol design and analysis for wireless networks.

Chi Zhou (SM'09) received the B.S. degrees in automation and in business administration from Tsinghua University, Beijing, China, in 1997 and the M.S. and Ph.D. degrees in electrical and computer engineering from Northwestern University, Evanston, IL, in 2000 and 2002, respectively.

From 2002 to 2006, she was an Assistant Professor with the Department of Electrical and Computer Engineering, Florida International University, Miami. Since August 2006, she has been with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, as an Assistant Professor. Her research interests include wireless communications and mobile networks in general and, in particular, power control/resource allocation for various wireless networks, the integration of heterogeneous networks, and reliable communications over orthogonal frequency division multiplexing or multiple-input–multiple-output systems.

Yu Cheng (S'01–M'04–SM'09) received the B.E. and M.E. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1995 and 1998, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2003.

From September 2004 to July 2006, he was a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON. Since August 2006, he has been with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, as an Assistant Professor. His research interests include next-generation Internet architectures and management, wireless network performance analysis, network security, and wireless/wireline interworking.

Dr. Cheng served as a Cochair for the 2009 IEEE International Conference on Commuincations Wireless Networking Symposium, a Cochair for the 2011 IEEE Global Communications Quality-of-Service, Reliability, and Modeling Symposium, and a Technical Program Committee Cochair for the Sixth International Conference on Wireless Algorithms, Systems, and Applications in 2011. He is an Associated Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He received a Postdoctoral Fellowship Award from the Natural Sciences and Engineering Research Council of Canada in 2004 and a Best Paper Award at the International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness in 2007. He received the National Science Foundation Faculty Early Development Career Award in 2011.