A Metric for DISH Networks:
Analysis, Implications, and Applications

Tie Luo, *Member, IEEE,* Vikram Srinivasan, *Member, IEEE,* and Mehul Motani, *Member, IEEE*

Abstract—In wireless networks, node cooperation has been exploited as a data relaying mechanism for decades. However, the wireless channel allows for much richer interaction among nodes. In particular, Distributed Information SHaring (DISH) represents a new improvement to multi-channel MAC protocol design by using a cooperative element at the control plane. In this approach, nodes exchange control information to make up for other nodes' insufficient knowledge about the environment, and thereby aid in their decision making. To date, what is lacking is a theoretical understanding of DISH. In this paper, we view cooperation as a network resource and evaluate the availability of cooperation, p_{co}. We first analyze p_{co} in the context of a multi-channel multi-hop wireless network, and then perform simulations which show that the analysis accurately characterizes p_{co} as a function of underlying network parameters. Next, we investigate the correlation between p_{co} and network metrics such as collision rate, packet delay, and throughput. We find a near-linear relationship between p_{co} and the metrics, which suggests that p_{co} can be used as an appropriate performance indicator itself. Finally, we apply our analysis to solving a channel bandwidth allocation problem, where we derive optimal schemes and provide general guidelines on bandwidth allocation for DISH networks.

Index Terms—Distributed information sharing, control-plane cooperation, performance analysis, channel bandwidth allocation, multi-channel MAC protocols.

❖

1 INTRODUCTION

COOPERATIVE diversity is not a new concept in wireless communications. Key ideas and results in cooperative communications can be traced back to the 1970s to van der Meulen [1] and Cover & El Gamal [2], whose works have spurred numerous studies on this topic from an information-theoretic perspective (e.g., [3], [4], [5], [6]) or a protocol-design perspective (e.g., [7], [8], [9], [10]). To date, cooperation has been intensively studied in various contexts, as a *data relaying* mechanism where intermediate nodes help relay packets from a transmitter to a receiver. However, the wireless channel allows for much richer interaction among nodes. Consider another possibility, where nodes exchange control information to make up for other nodes' insufficient knowledge about the environment, and thereby aid in their decision making. This represents a notion of *control-plane cooperation*, which augments the conventional understanding of cooperation at the data plane.

This paper specifically studies the distributed flavor of control-plane cooperation, which we call Distributed Information SHaring (DISH). It is a general approach and can be customized into specific mechanisms in different environments. For example in a multi-channel network, due to insufficient knowledge about channel or node status, a node may select a conflicting channel to exchange data, or a transmitter may initiate communication to a receiver who is however on a different channel. In both cases (which are the variants of a *multi-channel coordination problem* define later), DISH can be used such that neighboring nodes who possess relevant information share the information

to help the particular node avoid the problem. Recently, a DISH-based multi-channel MAC protocol was proposed in [11], [12], and simulations therein demonstrated significant performance gain over other non-cooperative protocols.

However, to date, what is still lacking is a theoretical understanding of DISH. The benefit of DISH is that it can remove the need for multiple transceivers [13], [14], [15], [16], [17] and time synchronization [18], [19], [20], [21], [22], [23] in designing multi-channel MAC protocols. This motivates us to understand DISH from a theoretical perspective. In this paper, we provide an analysis of DISH in terms of the *availability of cooperation* by viewing cooperation as a network resource. We define a metric p_{co} in the context of a general multi-channel network. It characterizes the availability of cooperation as the probability that a multi-channel coordination problem is identified by any neighboring node (who thus becomes cooperative). We analyze this metric in multi-hop networks with randomly distributed nodes, and verify the analysis via simulations. It is shown that our analysis accurately characterizes the availability of cooperation as a function of underlying network parameters, and reveals *what* factors and *how* these factors affect cooperation. This serves as a guide in provisioning a DISH network toward its optimal performance.

We then carry out a detailed investigation of p_{co} with three different contexts of DISH: non-cooperative case, ideal DISH, and real DISH. The investigation shows that p_{co} is a very meaningful performance indicator for DISH networks. Moreover, we apply our analysis to solving a practical channel bandwidth allocation problem, which yields insights different from non-cooperative networks.

Throughput analysis for multi-hop networks is difficult (and still an open problem in general), and it gets even more complicated in a multi-channel context with DISH.

- *Tie Luo and Mehul Motani are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117582.*
- *Vikram Srinivasan is with Bell Labs Research, Bangalore 560095, India.*

Our approach in this paper is to first look at p_{co} and then correlate it with network performance. We found a simple relationship between p_{co} and typical performance metrics.

The specific findings of this study are:

1) The availability of cooperation is high ($p_{co} > 0.7$) in typical cases, which suggests that DISH is feasible to use in multi-channel MAC protocols.
2) The performance degradation due to an increase in node density can be alleviated due to the simultaneously increased availability of cooperation.
3) The metric p_{co} increases with packet size for a given *bit* arrival rate, but decreases with packet size for a given *packet* arrival rate.
4) Node density and traffic load have opposite effects on p_{co} but node density is the dominating factor. This implies an improved scalability for DISH networks as p_{co} increases with node density.
5) p_{co} is strongly correlated to network performance and has a near-linear relationship with metrics such as throughput and delay. This may significantly simplify performance analysis for cooperative networks, and suggests that p_{co} be used as an appropriate performance indicator itself.
6) p_{co} is concave and not monotonic with respect to the ratio between control channel bandwidth and data channel bandwidth; there exists one and only one maximum p_{co} for each given set of parameters.
7) The optimal bandwidth ratio between the control and a data channel increases with node density but decreases with traffic load. This tells us that, for example, in a sparser network with heavier traffic, the control channel should be allocated less bandwidth for larger p_{co}.
8) In most cases (the number of channels is not too small), to boost the availability of cooperation, the control channel should be allocated more bandwidth than each data channel, rather than using *smaller* frequency band for a control channel or the usual *equal* bandwidth partition as suggested by many other studies.

Part of this work was presented in [24]. We review related work in Section 2, give the system model in Section 3 and then the analysis in Section 4. The analysis is verified in Section 5 where we also provide an detailed investigation of p_{co} in different contexts of DISH. Then we present an application of our analysis in Section 6. Finally, Section 7 gives concluding remarks.

2 RELATED WORK

Multi-channel MAC protocols for ad hoc networks can be categorized into multi-radio schemes and single-radio schemes. In the first category [13], [14], [15], [16], [17], each node dedicates a radio to monitoring channel usage when the other radio(s) are engaged in communication. In the second category, the mainstream approach is using time synchronization to (i) require nodes to negotiate channels in common time slots [18], [19], [20] or (ii) hop among channels according to well-known or predictable sequences [21], [22], [23]. The first category sacrifices cost-efficiency due to the extra hardware requirement, and the second category underperforms in scalability due to the complexity of time synchronization.

There are three other studies most related to this work. One is CAM-MAC [11], [12] which is the first DISH-based protocol. This multi-channel MAC protocol uses a single transceiver but does not require time synchronization. In the protocol, a transmitter and a receiver perform a handshake on the control channel to reserve a free data channel. If they choose a busy channel or the receiver is not on the control channel (and hence deaf to the transmitter), neighboring nodes who recognize this (via overhearing) may send cooperative messages to inform the transmitter and/or the receiver. Our later evaluation of real DISH will be based on an adapted version of CAM-MAC.

The second work, CoopMAC [10], is also a cooperative MAC protocol but the cooperation refers to data relaying as in many other protocols such as [7], [8], [9]. This protocol replaces a single low-rate transmission with two high-rate transmissions by using a relay node, in order to achieve higher throughput. The paper provides a protocol analysis which involves computing the probability that a relay node is available. This probability seems to be similar to p_{co} but they are actually very different. The probability defined in [10] is a *spatial* metric because it is completely determined by the (static) node placement, i.e., a relay is deemed available if there is a node in the intersection of the radio footprints of the source and the destination. This can be easily calculated via simple geometric analysis (as the paper assumes uniformly distributed nodes). On the other hand, p_{co} is a metric involving spatial, temporal, and frequency factors: besides node locations, nodes' communication activities (which vary with time), and the channels that nodes are tuned to (while [10] assumes a single channel), all will affect p_{co}. Furthermore, the problem context of the analysis in [10] is a *wireless LAN* whereas our context is a multi-hop network. Finally, [10] deals with data-plane cooperation whereas our work deals with control-plane cooperation.

The last work is by Han et al. [25] who considered a multi-channel MAC protocol that adopts ALOHA on the control channel to reserve data channels. Queueing theory was applied to calculate the throughput of the protocol. However, there are some noteworthy limitations. First, the scenario under consideration is single-hop. Second, it is assumed that each node can communicate on the control channel and a data channel simultaneously. This implicitly requires *two transceivers* per node and consequently leads to collision-free data channels. Third, a unique *virtual queue* was assumed to store the packets arriving at *all* nodes for the purpose of *centralized transmission scheduling*, and the precise status of this queue was assumed to be known to the entire network. This assumption is impractical and makes the calculated throughput actually an upper bound. Fourth, the access to the control channel uses the ALOHA mechanism while CSMA/CA is widely adopted in practice.

3 SYSTEM MODEL

We consider a static and connected ad hoc network in which each node is equipped with a single half-duplex transceiver that can dynamically switch between a set of orthogonal frequency channels but can only use one at a time. One channel is designated as the control channel and the others as data channels. Nodes are placed in a plane area according to a two-dimensional Poisson point process.

We consider a class of multi-channel MAC protocols with their common framework described below. A transmitter-receiver pair uses an McRTS/McCTS handshake on the control channel to set up communication (like 802.11 RTS/CTS) for their subsequent DATA/ACK handshake on a data channel. To elaborate, a transmitter sends an McRTS on the control channel using CSMA/CA, i.e., it sends McRTS after sensing the control channel to be idle for a random period (addressed below) of time. The intended receiver, after successfully receiving McRTS, will send an McCTS and then switch to a data channel (the McRTS informs the receiver of the chosen data channel). After successfully receiving the McCTS, the transmitter will also switch to its chosen data channel, and otherwise it will back off on the control channel for a random period (addressed below) of time. Hence it is possible that only the receiver switches to the data channel. After switching to a data channel, the transmitter will send a DATA and the receiver will respond with a ACK upon successful reception. Then both of them switch back to the control channel.

In the above we have mentioned two random periods. They are assumed to be designed such that idle intervals on the control channel are well randomized. Specifically, when a node is on the control channel, it sends control messages (an aggregated stream of McRTS and McCTS) according to a Poisson process with an unknown average rate (will be determined in Section 4). The validity of this assumption will be verified via simulations.

Note that we use McRTS, McCTS, DATA and ACK to refer to different packets (frames) without assuming specific frame formats. Since, logically, they must make a protocol functional, we assume that McRTS carries channel usage information (e.g., "who will use which channel for how long") and, for simplicity, McCTS is the same as McRTS.

We assume that, after switching to a data channel, a node will stay on that channel for a period of T_d, where T_d is the duration of a successful data channel handshake.[1] We ignore channel switching delay as it will not fundamentally change our results if it is negligible compared to T_d (the delay is $80\mu s$ [23] while T_d is more than $6ms$ for a 1.5KB data packet on a 2Mb/s channel). We also ignore SIFS and propagation delay for the same reason, provided that they are smaller than the transmission time of a control message.

We assume a uniform traffic pattern — all nodes have the equal data packet arrival rate, and for each data packet to send, a node chooses a receiver equally likely among

---

1. This is also valid for a failed data channel handshake. See appendix for the elaboration.

its neighbors. We also assume a stable network — all data packets can be delivered to destinations within finite delay. In addition, packet reception fails if and only if packets collide with each other (i.e., no capture effect), transmission and interference ranges are equal, and neighboring nodes do not start sending control messages simultaneously (there is no time synchronization).

We do not assume a specific channel selection strategy; how a node selects data channels will affect how often conflicting channels are selected, but will not affect p_{co}. This is because, intuitively, we only care about the availability of cooperation (p_{co}) when a multi-channel coordination problem (a precise definition is given in Def. 1), which includes channel conflicting problem, *has been* created.

We do not assume a concrete DISH mechanism, i.e., nodes do not physically react upon a multi-channel coordination problem, because analyzing the availability of cooperation does not require the use of this resource. In fact, assuming one of the (numerous possible) DISH mechanisms will lose generality. Nevertheless, we will show in Section 5 that, when an ideal or a real DISH mechanism is used, the results do not fundamentally change. This could be an overall effect from contradicting factors which will be explained therein.

We assume that the following parameters are known:

- n: node density. In a multi-hop network, it is the average number of nodes per R^2 where R is the transmission range. In a single-hop network, it is the total number of nodes.
- λ: the average data packet arrival rate at each node, including retransmissions.
- T_d: the duration of a data channel handshake.
- b: the transmission time of a control message. $b \ll T_d$.

4 ANALYSIS

4.1 Problem Formulation and Analysis Outline

We first formally define p_{co}, which depends on two concepts called the *MCC problem* and the *cooperative node*.

Definition 1 (MCC Problem): A multi-channel coordination (MCC) problem is either a *channel conflict* problem or a *deaf terminal* problem. A channel conflict problem is created when a node, say y, selects a channel to use (transmit or receive packets) but the channel is already in use by a neighboring node, say x. A deaf terminal problem is created when a node, say y, initiates communication to another node, say x, that is however on a different channel. In either case, we say that an MCC problem is created by x and y.

In a protocol that transmits DATA without requiring ACK, a channel conflict problem does not necessarily indicate an impending data collision. We do not consider such a protocol.

Definition 2 (Cooperative Node): A node that identifies an MCC problem created by two other nodes, say x and y, is called a cooperative node with respect to x and y.

Fig. 1 gives a visualization of the above two concepts in our system model.

TABLE 1: Notation

|		
Probabilities	p_{co}^{xy}	the probability that at least one cooperative node with respect to x and y exists
	$p_{co}^{xy}(v)$	the probability that node v is a cooperative node with respect to x and y
	p_{ctrl}	the probability that a node is on the control channel at an arbitrary point in time
	p_{succ}	the probability that a control channel handshake (initiated by an McRTS) is successful
	p_{oh}	the probability that an arbitrary node successfully overhears a control message
Events	$\mathcal{C}_v(t)$	node v is on the control channel at time t
	$\mathcal{O}(v \leftarrow i)$	node v successfully overhears node i's control message, given that i sends the message
	$\mathcal{S}_v(t_1, t_2)$	node v is silent (not transmitting) on the control channel during interval $[t_1, t_2]$
	$\mathcal{I}_v(t_1, t_2)$	node v does not introduce interference to the control channel during interval $[t_1, t_2]$, i.e., it is on a data channel or is silent on the control channel
	$\Omega_u(t_1, t_2)$	node u, which is on a data channel at t_1, switches to the control channel in $[t_1, t_2]$
Others	$\mathcal{N}_i, \mathcal{N}_{ij}, \mathcal{N}_{v \setminus i}$	$\mathcal{N}_i$ is the set of node i's neighbors, $\mathcal{N}_{ij} = \mathcal{N}_i \cap \mathcal{N}_j$, $\mathcal{N}_{v \setminus i} = \mathcal{N}_v \setminus \mathcal{N}_i \setminus \{i\}$ (v's but not i's neighbors)
	$K_{ij}, K_{v \setminus i}$	$K_{ij} = |\mathcal{N}_{ij}|$, $K_{v \setminus i} = |\mathcal{N}_{v \setminus i}|$
	s_i	the time when node i starts to send a control message
	$\lambda_c, \lambda_{rts}, \lambda_{cts}$	the average rates of a node sending control messages, McRTS, and McCTS, respectively, *when it is on the control channel*. Clearly, $\lambda_c = \lambda_{rts} + \lambda_{cts}$
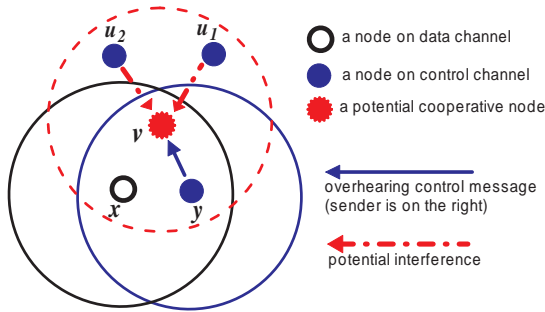
Fig. 1: Illustration of an MCC problem and a cooperative node. Node x is performing a data channel handshake on CH_x, and y has just sent a control message during a control channel handshake. If this control message is an McRTS addressed to x, then a *deaf terminal* problem is created. If this control message indicates that y selects CH_x (recall that a control message carries channel usage information), a *channel conflict* problem is created. In either case, if a third node v identifies this problem (by overhearing x's and y's control messages successively), it is a cooperative node.

Definition 3 (p_{co}): p_{co} is the probability for two arbitrary nodes that create an MCC problem to obtain cooperation, i.e., there is at least one cooperative node with respect to these two nodes.

Note that if there are multiple cooperative nodes and they are allowed by a DISH mechanism to send cooperative messages concurrently, a collision can result at node y. However, this collision *still indicates an MCC problem* and thus cooperation is still obtained. CAM-MAC [11], [12] also implements this.

We distinguish the receiving of control messages. A transmitter receiving McCTS from its intended receiver is referred to as *intentional receiving*, and the other cases of receiving are referred to as *overhearing*, i.e., any node receiving McRTS (hence an intended receiver may also be a cooperative node) or any node other than the intended transmitter receiving McCTS.

Our notation is listed in Table 1. Overall, we will determine p_{co} by following the order of $p_{co}^{xy}(v) \rightarrow p_{co}^{xy} \rightarrow p_{co}$.

Consider $p_{co}^{xy}(v)$ first. Fig. 1 illustrates that node v is cooperative if and only if it successfully overhears x's and

y's control messages successively. Hence $\forall v \in \mathcal{N}_{xy}$,

$$p_{co}^{xy}(v) = \Pr[\mathcal{O}(v \leftarrow x), \mathcal{O}(v \leftarrow y)]$$
$$= \Pr[\mathcal{O}(v \leftarrow x)] \cdot \Pr[\mathcal{O}(v \leftarrow y)|\mathcal{O}(v \leftarrow x)]. \quad (1)$$

Consider $\mathcal{O}(v \leftarrow i)$. For v to successfully overhear i's control message which is being sent during interval $[s_i, s_i + b]$, v must be silent on the control channel and not be interfered, i.e.,

$$\Pr[\mathcal{O}(v \leftarrow i)] = \Pr[\mathcal{S}_v(s_i, s_i + b), \bigcap_{u \in \mathcal{N}_v \setminus \{i\}} \mathcal{I}_u(s_i, s_i + b)],$$
$$\forall v \in \mathcal{N}_i. \quad (2)$$

Now we outline our analysis as below.
- Section 4.2: solves (2).
- Section 4.3: solves (1) and the target metric p_{co}.
- Section 4.4: case study in single-hop networks.

Proofs are in the Appendix unless otherwise specified.

4.2 Solving (2)

Proposition 1: If node v is overhearing a control message from node i during $[s_i, s_i + b]$, then the probability that a node $u \in \mathcal{N}_v$ does not interfere with v is given by

$$\Pr[\mathcal{I}_u(s_i, s_i + b)] = \begin{cases} 1, & u \in \mathcal{N}_{vi}; \\ p_{ni\text{-}oh}, & u \in \mathcal{N}_{v \setminus i}. \end{cases}$$

where

$$p_{ni\text{-}oh} = p_{ctrl} \cdot e^{-2\lambda_c b} + (1 - p_{ctrl}) \cdot (1 - \frac{2b}{T_d} + \frac{1 - e^{-2\lambda_c b}}{\lambda_c T_d}).$$

Thus (2) can be reduced to (see Appendix)

$$\Pr[\mathcal{O}(v \leftarrow i)] \approx p_{ctrl} \, p_{ni\text{-}oh}^{K_{v \setminus i}}. \quad (3)$$

The above contains two unknown variables, p_{ctrl} and λ_c, and the following solves for them.

For p_{ctrl}, consider a sufficiently long period T_0. On the one hand, the number of arrival data packets at each node is λT_0. On the other hand, each node spends a total time of $(1 - p_{ctrl})T_0$ on data channels, a factor η of which is used for sending arrival data packets. Since the network is stable (incoming traffic is equal to outgoing traffic), we establish a balanced equation:

$$\lambda T_0 T_d = \eta (1 - p_{ctrl})T_0.$$

To determine η, noticing that a node switches to data channels either as a transmitter (with an average rate of λ) or as a receiver (with an average rate of λ_{cts}), we have $\eta = \lambda/(\lambda + \lambda_{cts})$. Substituting this into the above yields

$$p_{ctrl} = 1 - (\lambda + \lambda_{cts})T_d. \qquad (4)$$

For λ_c (and λ_{cts}), we need a proposition and two lemmas.

Proposition 2: If node i (transmitter) is intentionally receiving McCTS from node j (receiver) during $[s_j, s_j + b]$, then the probability that a node $u \in \mathcal{N}_i$ does not interfere with i is given by

$$\Pr[\mathcal{I}_u(s_j, s_j + b)] = \begin{cases} 1, & u \in \mathcal{N}_{ij}; \\ p_{ni\text{-}cts}, & u \in \mathcal{N}_{i \setminus j}. \end{cases}$$

where $p_{ni\text{-}cts} =$

$$(1 - p_{ctrl})[1 - \frac{b}{T_d}(1 + \frac{b}{T_d} - \frac{1 - e^{-\lambda_c b}}{\lambda_c T_d} - e^{-\lambda_c b})] + p_{ctrl}.$$

Lemma 1: For a Poisson random variable K with mean value $\overline{K}$, and $0 < p < 1$,

$$\mathbb{E}[p^K] = e^{-(1-p)\overline{K}}.$$

Proof:

$$\mathbb{E}[p^K] = \sum_0^\infty p^k \Pr(K = k) = e^{-\overline{K}} \sum_0^\infty \frac{(p\overline{K})^k}{k!} = e^{-(1-p)\overline{K}}.$$

$\square$

Lemma 2: For three random distributed nodes v, i and j,
(a) $\mathbb{E}[K_{v \setminus i} | v \in \mathcal{N}_i] \approx 1.30n$.
(b) $\mathbb{E}[K_{v \setminus i} | v \in \mathcal{N}_{ij}] \approx 1.19n$.
(c) $\mathbb{E}[K_{ij}] \approx 1.84n$.

Now we can prove (see Appendix)

$$p_{oh} \approx p_{ctrl} \exp[-1.30n(1 - p_{ni\text{-}oh})],$$
$$p_{succ} \approx p_{oh} \exp[-1.30n(1 - p_{ni\text{-}cts})]. \qquad (5)$$

Now we solve λ_c (together with λ_{cts}). From the perspective of a transmitter, the average number of successful control channel handshakes that it initiates per second is $p_{ctrl}\lambda_{rts}p_{succ}$. Since each successful control channel handshake leads to transmitting one data packet, we have

$$p_{ctrl}\lambda_{rts}p_{succ} = \lambda.$$

From the perspective of a receiver, it sends an McCTS when it successfully receives (overhears) an McRTS addressed to it, [2] and hence $\lambda_{cts} = \lambda_{rts} p_{oh}$. Then combining these with $\lambda_c = \lambda_{rts} + \lambda_{cts}$ yields

$$\lambda_c = \frac{\lambda(1 + p_{oh})}{p_{ctrl} \, p_{succ}} \text{ and } \lambda_{cts} = \frac{\lambda \, p_{oh}}{p_{ctrl} \, p_{succ}} \qquad (6)$$

where p_{oh} and p_{succ} are given in (5).

---

2. In a more sophisticated protocol model, a receiver may not respond even if it receives McRTS due to, e.g., disagreeing with the transmitter's channel selection. This behavior is modeled by ideal DISH and real DISH, which will be shown in Section 5 that it does not fundamentally change the results.

4.3 Solving (1) and Target Metric p_{co}

Based on the proof of (3), it can be derived that

$$\Pr[\mathcal{O}(v \leftarrow y) | \mathcal{O}(v \leftarrow x)] \approx p_{ctrl}^\star \, p_{ni\text{-}oh}^{K_{v \setminus y}}, \qquad (7)$$

$$\text{where} \quad p_{ctrl}^\star \triangleq \Pr[\mathcal{C}_v(s_y) | \mathcal{O}(v \leftarrow x)].$$

Note that $p_{ctrl}^\star \neq p_{ctrl}$, because s_y is not an *arbitrary* time for v due to the effect form $\mathcal{O}(v \leftarrow x)$. The reason is that $\mathcal{O}(v \leftarrow x)$ implies $\mathcal{C}_v(s_x)$, and thus for $\mathcal{C}_v(s_y)$ to happen, v must stay *continuously* on the control channel during $[s_x, s_y]$ (otherwise, a switching will lead to v staying on the data channel for T_d, but $s_x + T_d > s_y$ since x's data communication is still ongoing at s_y, and hence $\mathcal{C}_v(s_y)$ can never happen).

It can be proven that (see Appendix)

$$p_{ctrl}^\star = \frac{(w\lambda_c - \frac{1-w}{T_d}) \, g(\lambda_c + \lambda_w) + \frac{1-w}{T_d} \, g(\lambda_w)}{1 - w + (w\lambda_c - \frac{1-w}{T_d}) \, g(\lambda_c)} \qquad (8)$$

where

$$g(x) = \frac{1 - e^{-xT_d}}{x}, \quad w = \frac{p_{ctrl} - p_{oh}}{1 - p_{oh}}, \text{ and}$$

$$\lambda_w = \lambda_{rts}p_{succ} + \lambda_{cts}.$$

Combining (3) and (7) reduces (1) to

$$p_{co}^{xy}(v) \approx p_{ctrl} \, p_{ctrl}^\star \, p_{ni\text{-}oh}^{K_{v \setminus x} + K_{v \setminus y}}, \; \forall v \in \mathcal{N}_{xy}. \qquad (9)$$

Let $p_{co}^{xy}(\star)$ be the average of $p_{co}^{xy}(v)$ over all $v \in \mathcal{N}_{xy}$, i.e., $p_{co}^{xy}(\star)$ is the probability that an arbitrary node in $\mathcal{N}_{xy}$ is cooperative with respect to x and y, Using Lemma 2-(b),

$$p_{co}^{xy}(\star) \approx p_{ctrl} \, p_{ctrl}^\star \, \exp[-2.38n(1 - p_{ni\text{-}oh})]. \qquad (10)$$

By the definition of p_{co}^{xy} in Table 1,

$$p_{co}^{xy} \approx 1 - \prod_{v \in \mathcal{N}_{xy}} [1 - p_{co}^{xy}(v)] \approx 1 - [1 - p_{co}^{xy}(\star)]^{K_{xy}}, \qquad (11)$$

where the events corresponding to $1 - p_{co}^{xy}(v)$, i.e., nodes not being cooperative with respect to x and y, are regarded as independent of each other, as an approximation.

Thus p_{co} is determined by averaging p_{co}^{xy} over all (x, y) pairs that are possible to create MCC problems. It can be proved [26] that these pairs are neighboring pairs (x, y) satisfying (d_i denoting the degree of a node i)

(a) $d_x \geq 2$, $d_y \geq 2$, but not $d_x = d_y = 2$, or
(b) $d_x = d_y = 2$, but x and y are not on the same three-cycle (triangle).

This condition is satisfied by all neighboring pairs in a *connected* random network, because the connectivity requires a sufficiently high node degree ($5.18 \log N$ where N is the total number of nodes [27]) which is much larger than 2. Therefore, taking expectation of (11) over all neighboring pairs using Lemma 1 and Lemma 2-(c),

$$p_{co} = 1 - \exp[-p_{co}^{xy}(\star)\overline{K_{xy}}]$$
$$\approx 1 - \exp[-1.84n \, p_{co}^{xy}(\star)]. \qquad (12)$$

This completes the analysis.

4.4 Special Case: Single-Hop Networks

Now that all nodes are in the communication range of each other, we have $p_{ni\text{-}oh} = p_{ni\text{-}cts} = 1$ according to Prop. 1 and 2, which leads to $p_{succ} = p_{oh} = p_{ctrl}$ according to (5), and $p_{co}^{xy}(v) = p_{ctrl}\, p_{ctrl}^{\star}$ according to (9). Hence (11) reduces to

$$p_{co}^{xy} = 1 - (1 - p_{ctrl}\, p_{ctrl}^{\star})^{K_{xy}},$$

where K_{xy} is the number of all possible cooperative nodes with respect to x and y, leading to $K_{xy} = n - 4$. [3] So, as the average of p_{co}^{xy},

$$p_{co} = 1 - (1 - p_{ctrl}\, p_{ctrl}^{\star})^{n-4}, \tag{13}$$

where p_{ctrl} is given below, by solving equations (4), (5) and (6),

$$p_{ctrl} = \frac{1}{2}(1 - \lambda T_d + \sqrt{1 + \lambda T_d(\lambda T_d - 6)}\,),$$

and $p_{ctrl}^{\star}$ is given below, by reducing (8) with $w = 0$,

$$p_{ctrl}^{\star} = \frac{g(\lambda_w) - g(\lambda_c + \lambda_w)}{T_d - g(\lambda_c)}$$

where $\quad \lambda_c = \frac{1}{2}(\frac{1 - \sqrt{1 + \lambda T_d(\lambda T_d - 6)}}{\lambda T_d^2} - \frac{3}{T_d})$,

$$\lambda_w = \frac{1 - \sqrt{1 + \lambda T_d(\lambda T_d - 6)}}{T_d} - \lambda.$$

5 INVESTIGATING p_{co} WITH DISH

We verify the analysis in both single-hop and multi-hop networks and identify key findings therein. We also investigate the correlation between p_{co} and network performance.

5.1 Protocol Design and Simulation Setup

5.1.1 Non-Cooperative Case

This is a multi-channel MAC protocol based on the protocol framework described in Section 3. Key part of its pseudo-code is listed below, where S_{ctrl} is the control channel status (FREE/BUSY) detected by the node running the protocol, S_{node} is the node's state (IDLE/TX/RX, etc.), L_{queue} is the node's current queue length, and they are initialized as FREE, IDLE and 0, respectively. The frame format of McRTS and McCTS is shown in Fig. 2, where we can see that they carry channel usage information. A node that overhears McRTS or McCTS will cache the information in a *channel usage table* shown in Fig. 3, where Until is converted from Duration by adding the node's own clock.

As is based on the system model, this protocol does not use a concrete DISH mechanism, i.e., cooperation is treated as a resource while not actually utilized.

---

3. This means all nodes excluding x, x', y and y', where x' is the node that x is currently communicating with (on a data channel), and y' is the node that y *was* communicating with (on a data channel) when x was setting up communication with x' (hence y and y' missed x's control message). None of these four nodes can be a cooperative node.

FC	Tx	Rx	Ch	Duration	CRC
2	6	6	1	2	2

FC: frame control   Tx: address of transmitter
Ch: channel index   Rx: address of receiver

Fig. 2: Frame format of McRTS and McCTS.

Tx	Rx	Ch	Until
T1	R1	1	00:15:36
T2	R2	3	00:16:01

Fig. 3: Channel usage table.

5.1.2 Ideal DISH

This protocol is by adding an ideal cooperating mechanism to the non-cooperative case. Each time when an MCC problem is created by nodes x and y and if at least one cooperative node is available, the node that is on the control channel, i.e., node y, will be informed without any message actually sent, and then back off to avoid the MCC problem.

5.1.3 Real DISH

In this protocol, cooperative nodes actually send cooperative messages to inform the transmitter or receiver of an MCC problem so that it will back off. We design a real DISH protocol by adapting CAM-MAC [11] and show the control channel handshake in Fig. 4. A transmitter and a receiver first perform a PRA/PRB exchange like the 802.11 RTS/CTS to negotiate a data channel, and then perform a CFA/CFB exchange to confirm (to their respective neighbors) the channel selection. A cooperative node sends an INV packet when identifying an MCC problem through PRA or PRB. We change CAM-MAC such that $\|PRA\| + \|CFA\| = \|McRTS\| = \|PRB\| + \|CFB\| = \|McCTS\|$, where $\|\cdot\|$ denotes packet size.

5.1.4 Simulation Setup

There are six channels of data rate 1Mb/s each (the number of channels does not affect results as long as the network is kept stable). Data packets arrive at each node as a Poisson process. The uniform traffic pattern as in the model is used. Traffic load λ (pkt/s), node density n $(1/R^2)$, and

---

Procedure 1 PKT-ARRIVAL

[Called when a data packet arrives]

1: enqueue the packet, $L_{queue}{++}$
2: **if** $S_{ctrl} = FREE \wedge S_{node} = IDLE \wedge L_{queue} = 1$ **then**
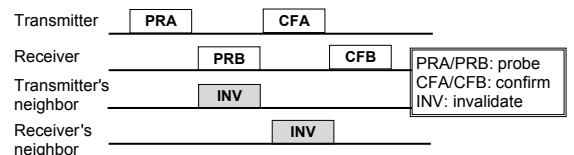3:   call ATTEMPT-RTS
4: **end if**

Fig. 4: Control channel handshake of Real DISH.

Procedure 2 ATTEMPT-RTS

[Called by PKT-ARRIVAL or CHECK-QUEUE]

1: construct a set $\mathcal{F}$ of free channel indexes using channel usage table
2: **if** $\mathcal{F} \neq \phi$ **then**
3:     send McRTS with CH:=RANDOM($\mathcal{F}$)
4: **else**
5:     Timer $\leftarrow \min(\texttt{until} - now)$
6:     **while** $S_{ctrl} = FREE \wedge$ Timer not expired **do**
7:         wait {carrier sensing remains on}
8:     **end while**
9:     **if** Timer expired **then**
10:        call CHECK-QUEUE
11:     **else**
12:        call PASSIVE {receive a control message}
13:     **end if**
14: **end if**

Procedure 3 CHECK-QUEUE

[Called when $S_{ctrl} = FREE \wedge S_{node} = IDLE$ changes from $FALSE$ to $TRUE$]

1: **if** $L_{queue} > 0$ **then**
2:     Timer $\leftarrow$ RANDOM$(0, 10b)$ {FAMA [28], [29]}
3:     **while** $S_{ctrl} = FREE \wedge$ Timer not expired **do**
4:         wait {carrier sensing remains on}
5:     **end while**
6:     **if** Timer expired **then**
7:        call ATTEMPT-RTS
8:     **else**
9:        call PASSIVE {receive a control message}
10:     **end if**
11: **end if**

---

packet size L (byte) will vary in simulations. In multi-hop networks, the network area is 1500m×1500m and the transmission range is 250m. Each simulation is terminated when a total of 100,000 data packets are sent over the network, and each set of results is averaged over 15 randomly generated networks.

5.2 Investigation with Non-Cooperative Case

5.2.1 Verification of Analysis

The p_{co} obtained via analysis and simulations are compared in Fig. 5. We see a close match between them, with a deviation of less than 5% in almost all single-hop scenarios, and less than 10% in almost all multi-hop scenarios. Particularly, the availability of cooperation is observed to be at a high level ($p_{co} > 0.7$ in most cases), which suggests that a large percentage of MCC problems would be avoided by exploiting DISH, and DISH is feasible to use in multi-channel MAC protocols. (**Finding 1**)

Specifically, Fig. 5a and Fig. 5c consistently show that, in both single-hop and multi-hop networks, p_{co} monotonically decreases as λ increases. The reasons are two folds. First, as traffic grows, each node spends more time on data channels for data transmission and reception,

(a) p_{co} versus λ ($n = 8$), single-hop. (b) p_{co} vs. n ($\lambda = 10$), single-hop.

(c) p_{co} vs. λ ($n = 10$), multi-hop. (d) p_{co} vs. n ($\lambda = 10$), multi-hop.
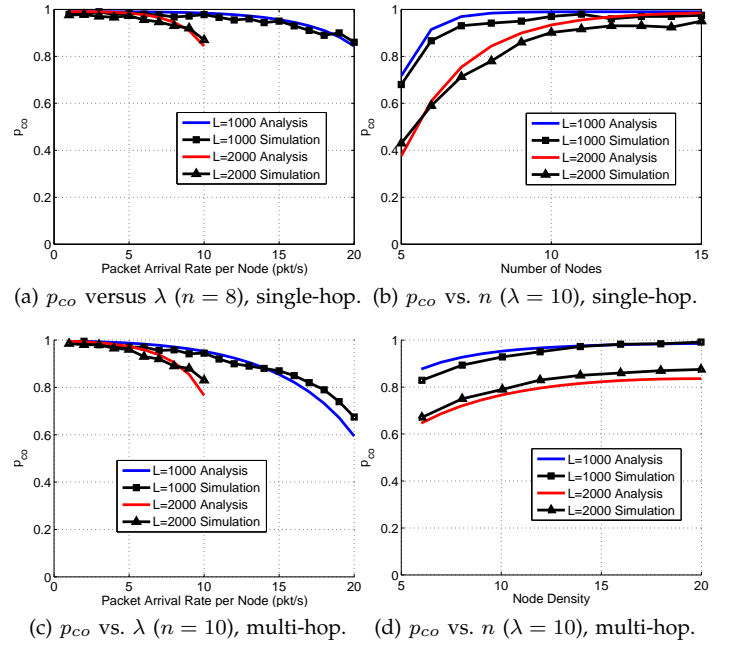
Fig. 5: Non-cooperative case: Impact of traffic load and node density, with different packet sizes. The value ranges of X axes are chosen such that the network is stable.

which reduces p_{ctrl} and hence the chance of overhearing control messages (p_{oh}), resulting in lower p_{co}. Second, as the control channel is the rendezvous to set up all communications, larger traffic intensifies the contention and introduces more interference to the control channel, which is hostile to messages overhearing and thus also reduces p_{co}.

Fig. 5b and Fig. 5d show that p_{co} monotonically increases as n increases, and is concave. The increase of p_{co} is because MCC problems are more likely to have cooperative nodes under a larger node population, while the deceleration of the increase is because more nodes also generate more interference to the control channel.

An important message conveyed by this observation is that, although a larger node density creates more MCC problems (e.g., more channel conflicts as data channels are more likely to be busy), it also boosts the availability of cooperation which avoids more MCC problems. This implies that the performance degradation can be mitigated. (**Finding 2**)

In both single-hop and multi-hop networks, a larger packet size L corresponds to a lower p_{co}. However, note that this is observed under the same *packet* arrival rate (pkt/s), which means actually a larger *bit* arrival rate for a larger L, and can be explained by the previous scenarios of p_{co} versus λ. Now if we consider the same *bit* arrival rate, by examining the two analysis curves in Fig. 5c where we compare p_{co} with respect to the same $\lambda \cdot L$ product, e.g., ($\lambda = 5, L = 2000$) versus ($\lambda = 10, L = 1000$), and ($\lambda = 10, L = 2000$) versus ($\lambda = 20, L = 1000$), then we will see that a larger L corresponds to a *higher* p_{co}, which is contrary to the observation under the same *packet* arrival rate. The explanation is that, for a given bit arrival
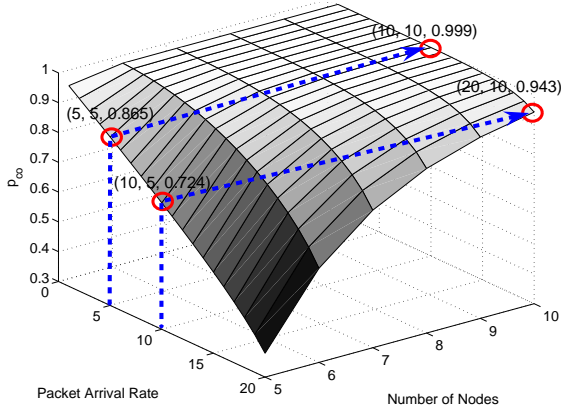
Fig. 6: p_{co} versus λ and n. Each of the two arrows indicates a multiplicative increase of λ and n with the same factor (two).

rate, increasing L reduces the number of packets and hence *fewer control channel handshakes* are required, thereby alleviating control channel interference. (**Finding 3**)

5.2.2 *Dominating Impact Factor*

The above results indicate that node density and traffic load affect the availability of cooperation in *opposite* ways. This section aims to find which one dominates over the other. In Fig. 6, we plot the relationship of p_{co} versus λ and n, given $L = 1000$ and based on the *analytical* result for single-hop networks. We multiplicatively increase λ and n with the same factor (two), and find that, when increasing (λ, n) from (5,5) to (10,10), p_{co} keeps *increasing* from 0.865 to 0.999, and when increasing (λ, n) from (10,5) to (20,10), p_{co} keeps *increasing* from 0.724 to 0.943. Consistent results were also observed in other scenarios, i.e., $L = 2000$ in single-hop, and $L = 1000, 2000$ in multi-hop networks.

This investigation shows that n is the dominating factor over λ that determines the variation of p_{co}. This implies that DISH networks should have better scalability than non-DISH networks, since p_{co} increases when both traffic load and node density scale up. (**Finding 4**)

5.3 Investigation with Ideal DISH

For ideal DISH, we only present results in multi-hop networks, as the single-hop simulation results were similar and the discussion in this section applies to both sets of results.

5.3.1 *Verification of Analysis*

The results of comparison are shown in Fig. 7, where p_{co} with ideal DISH well matches p_{co} of analysis. This confirms **Findings 1-3**, and we speculate the reasons to be as follows. With ideal DISH, a transmitter will be informed of a deaf terminal problem at times and thus back off for a fairly long time, which leads to *fewer* McRTS being sent. On the other hand, a node will also be informed of a channel conflict problem at times and thus re-select channel and retry shortly, which leads to *more* control messages being sent. Empirically, channel conflict problems occur more

often, and hence there will be an overall *increase* of control messages being sent. This escalates interference and thus would lower down p_{co}. However, nodes will switch to data channels less frequently because cooperation suppresses a number of conflicting data channel usages. This makes nodes stay *longer* on the control channel and would elevate p_{co}. Consequently, p_{co} does not significantly change.

As **Finding 4** is obtained via analysis which, as shown in Fig. 7, matches the simulations with ideal DISH, it is automatically confirmed.

5.3.2 *Correlation between p_{co} and Performance*

We investigate how p_{co} correlates to network performance — specifically, data channel collision rate ξ, packet delay δ, and aggregate throughput S. We consider both stable networks and saturated networks under multi-hop scenarios.

In stable networks, we measure (ξ, δ) and (ξ_{co}, δ_{co}) when without and with cooperation (ideal DISH), respectively. Then we compute $\eta_\xi = \xi_{co}/\xi$ and $\eta_\delta = \delta_{co}/\delta$ to compare to p_{co} with ideal DISH. The first set of results, by varying traffic load λ, is shown in Fig. 7a. We observe that the two *ascending* and *convex* curves of η_ξ and η_δ approximately *reflect* the *descending* and *concave* curve of p_{co}, which hints at a *linear* or *near-linear* relationship between p_{co} and these two performance ratios. That is, $\eta_\xi + p_{co} \approx c_1$, $\eta_\delta + p_{co} \approx c_2$, where c_1 and c_2 are two constants. The second set of results, by varying node density n, is shown in Fig. 7b. On the one hand, η_ξ and η_δ decreases as n increases, which is contrary to Fig. 7a. This confirms our earlier observations: n is amicable whereas λ is hostile to p_{co} (the smaller η_ξ and η_δ, the better performance cooperation offers). On the other hand, the correlation between p_{co} and the performance ratios is found again: as p_{co} increases on a concave curve, it is reflected by η_ξ and η_δ which decrease on two convex curves.

In saturated networks, we vary node density n and measure aggregate throughput without and with cooperation (ideal DISH), as S and S_{co}, respectively. Then we compute $\eta_S = S/S_{co}$ (note that this definition is inverse to η_ξ and η_δ, in order for $\eta_S \in [0,1]$) to compare to p_{co} with ideal DISH. The results are summarized in Fig. 8. We see that (i) p_{co} grows with n, which conforms to Finding 2, and particularly, (ii) the declining and convex curve of η_S reflects the rising and concave curve of p_{co}, which is consistent with the observation in stable networks. In addition, the p_{co} here is lower than the p_{co} in stable networks. This is explained by our earlier result that higher traffic load suppresses p_{co}.

In summary, the experiments in stable networks and saturated networks both demonstrate a strong correlation (*linear* or *near-linear* mapping) between p_{co} and network performance ratio in terms of typical performance metrics. This may significantly simplify performance analysis for cooperative networks via bridging the *nonlinear* gap between network parameters and p_{co}, and also suggests that p_{co} be used as an appropriate performance indicator itself. (**Finding 5**)

Note that this does not imply that the delay, the channel collision rate, and the throughput are linear with respect to
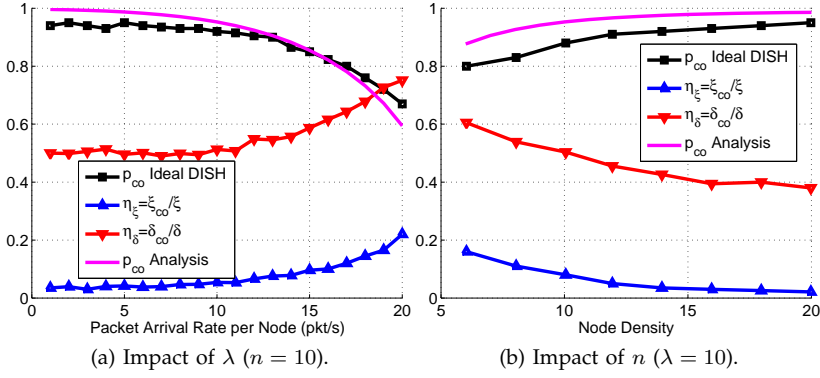
(a) Impact of λ ($n = 10$).

(b) Impact of n ($\lambda = 10$).

Fig. 7: Investigating p_{co} with ideal DISH in stable networks. This includes (i) verification of analysis, and (ii) correlation between p_{co} and (η_ξ, η_δ) (ratio of data collision, ratio of packet delay). $L = 1000$ bytes. The Y axis represents multiple metrics.
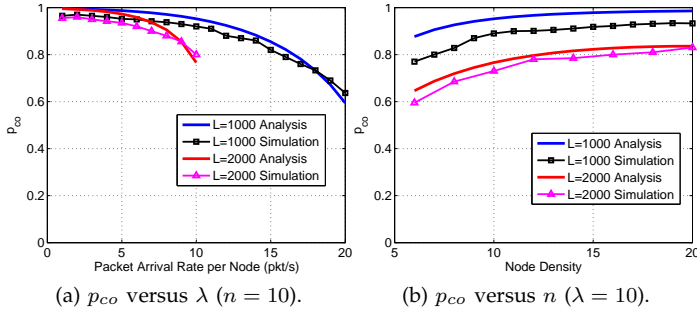

Fig. 8: Investigating p_{co} with ideal DISH in saturated networks: correlation between p_{co} and η_S (throughput ratio). $L = 1000$ bytes. The Y axis represents multiple metrics.

(a) p_{co} versus λ ($n = 10$).

(b) p_{co} versus n ($\lambda = 10$).

Fig. 9: Verification of p_{co} with real DISH in multi-hop networks. $L = 1000$ bytes.

each other, because the above stated relationship is for the performance *ratios* between DISH and non-DISH networks.

The explanation to this linear or near-linear relationship should involve intricate network dynamics. We speculate that the rationale might be that (i) MCC problems are an essential performance *bottleneck* to multi-channel MAC performance, and (ii) p_{co} is equivalent to the *ratio* of MCC problems that can be avoided by DISH. In any case, we reckon that this observation may spur further studies and lead to more thought-provoking results.

5.4 Investigation with Real DISH

For the same reason as mentioned for ideal DISH, we present the results for multi-hop networks only.

5.4.1 Verification of Analysis

As shown in Fig. 9, the simulation and analytical results still match, with deviation less than 15%. This is explained by two underlying factors. On the one hand, since real DISH actually sends cooperative messages, the control channel will have more interference which tends to diminish p_{co}. On the other hand, these cooperative messages inform transmitters or receivers of conflicting channel selections, which leads to a reduced number of channel switchings. Hence nodes will stay longer on the control
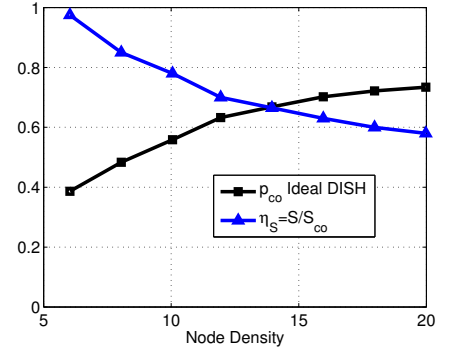
channel and hence p_{co} tends to be elevated. Consequently, p_{co} is kept close to the analytical result. **Findings 1-4** are thus confirmed.

5.4.2 Correlation between p_{co} and Performance

We examine this issue using scatter plots which provide another point of view besides the direct representation in Section 5.3. These plots are given in Fig. 10, where it is more apparent to see the near-linear relationship between p_{co} and $\eta_\xi, \eta_\delta, \eta_S$ (respectively). This confirms **Finding 5**.

An observation is that, while η_δ and η_S (the ratio of delay and throughput, respectively) in real DISH are slightly larger than those in ideal DISH, η_ξ (the ratio of data channel collision) is slightly smaller. Note that, by definition, the smaller these ratios are, the better the corresponding performance is (i.e., shorter delay, fewer collisions, and higher throughput, respectively). To explain this difference, first notice that the larger η_δ and η_S is simply because real DISH has to afford overhead for cooperation (physically send cooperative messages) while ideal DISH does not need to. Second, the smaller η_ξ, which counter-intuitively conveys fewer data channel collisions than ideal DISH, relates to two factors: (i) the transmissions of cooperative messages in real DISH lowers down the efficiency of cooperation than in ideal DISH, as explained before, and hence each use of a data channel in real DISH is slightly more likely to encounter collision, (ii) the cooperative messages suppress nearby nodes from *initiating* handshakes (via CSMA) and also interfere *ongoing* control channel handshakes, leading to fewer accomplished control channel handshakes per second, and hence fewer data channel usages than ideal DISH. The latter factor, according to the simulation results, outweigh the former factor, thereby explains the smaller η_ξ. In fact, these two factors also contribute to the longer delay and lower throughput in real DISH than in ideal DISH.

6 CHANNEL BANDWIDTH ALLOCATION

Our investigation shows that p_{co} is a meaningful performance indicator for DISH networks and captures several

(a) η_δ and η_ξ versus p_{co}, varying λ ($n = 10$). (b) η_δ and η_ξ versus p_{co}, varying n ($\lambda = 10$). (c) η_S versus p_{co}.
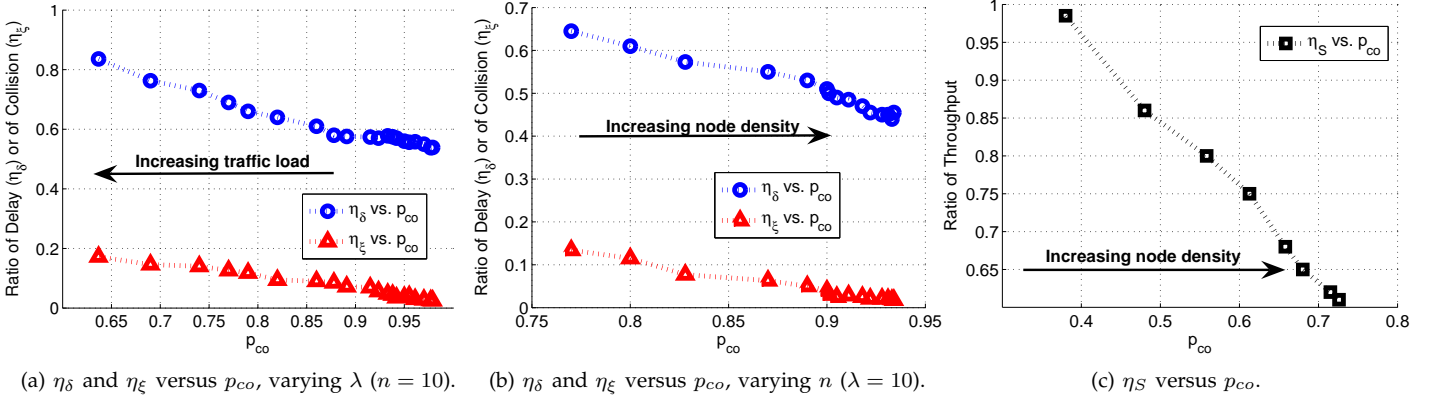
Fig. 10: Correlation between p_{co} and different performance metrics with real DISH in multi-hop networks.

other critical performance metrics. In this section, we leverage p_{co} as an instrument and apply our analysis to solving an important issue in multi-channel operation, channel bandwidth allocation.

6.1 Problem Formulation

For a given amount of total channel bandwidth W, define a bandwidth allocation scheme $\mathcal{A}_m = (m, \sigma)$, where m is the number of data channels and $\sigma = w_c/w_d$, in which w_c is the bandwidth of the control channel and w_d is the bandwidth of a data channel. The objective is to obtain the optimal scheme $\mathcal{A}_m^* = (m, \sigma^*)$ which achieves the maximum p_{co} for a given m. We remark that:

- An implicit assumption of the problem formulation is that bandwidth is equally partitioned among all *data* channels.
- In the formulation, m is designated as an input parameter instead of a variable subject to optimization. This is because of the following: (i) in practice, the main consideration on choosing m is *system capacity* (the number of users a system can accommodate), (ii) if, otherwise, m is a variable subject to optimization, the formulation will be equivalent to $\mathcal{A} = (w_c, w_d)$ and its solution will be a *single* "universally optimal" m which generally does not fit into practical situations.
- There exists another bandwidth allocation problem, where data channel bandwidth w_d is fixed and only control channel bandwidth w_c can be adjusted (or vice versa). We do not investigate this problem because (i) from a practical perspective, radio frequency band is a regulated or highly limited resource and cannot be arbitrarily claimed, and (ii) even if the band can be arbitrarily claimed, then the solution becomes obvious — p_{co} will monotonically increase as w_c or w_d grows, as a consequence of using more resource.

In the following, we solve the above optimization problem using the analytical results derived in Section 4. The feasibility of applying our analysis is based on the consistency between p_{co} of analysis and that of simulations across non-cooperative case, ideal DISH and real DISH, as shown in Section 5.
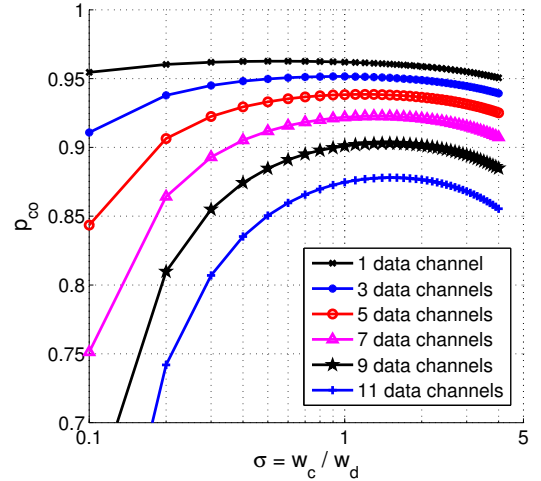

Fig. 11: p_{co} versus σ under different m.

6.2 Solutions and Discussion

Denoting the size of a control packet and a data packet by l_c and L, respectively, we have[4]

$$l_c = w_c b, \quad L = w_d T_d.$$

Combined with $W = w_c + m \cdot w_d$, we have

$$\frac{l_c}{b} + \frac{mL}{T_d} = W. \tag{14}$$

Then we rewrite σ as

$$\sigma = \frac{T_d l_c}{bL}. \tag{15}$$

Next, for a given W and protocol-specified l_c and L, compute b and T_d using (14) and (15) for different combinations of m and σ. Then substitute each pair of b and T_d into the equations derived in Section 4 and calculate p_{co}. By this means, we will obtain a matrix of p_{co} which corresponds to different combinations of m and σ. Finally, comparing p_{co} for each m in order to find the maximum will obtain the optimal solution $\mathcal{A}_m^* = (m, \sigma^*)$ for a given m.

---

4. We assume that ACK packet size is negligible compared to data packet size. Or alternatively, one can simply define L as the sum size of data and ACK packets.
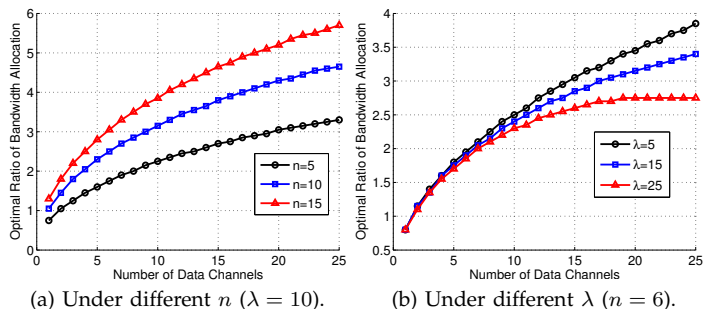
(a) Under different n ($\lambda = 10$).   (b) Under different λ ($n = 6$).

Fig. 12: σ^* versus m under different combinations of n and λ. $L = 1000$ bytes.

Using this method, we can demonstrate results given a set of parameters. Fig. 11 is such a plot given the following parameters: $L = 2000$ bytes, $l_c = 34$ bytes (cf. Fig. 2, plus PHY preamble and header), $W = 40$Mb/s, $n = 6$, $\lambda = 20$. We can see that p_{co} is concave and *not* monotonic with respect to σ, and it reaches the maximum at a certain σ on each curve corresponding to a specific m. There are two counteractive factors attributing to this. First, as σ increases, the control channel is allocated more bandwidth, so the time needed for transmitting a control packet, and hence the vulnerable period of receiving a control packet, is being reduced. As such, the probability of successful overhearing will increase and hence p_{co} tends to be higher. Second, as W is fixed, increasing control channel bandwidth has to squeeze data channels simultaneously, which prolongs data packet transmission time and hence is to the effect of enlarging data packet size L. According to Section 5, a larger L leads to a lower p_{co}. (**Finding 6**)

In particular, we obtain $\mathcal{A}_1^* = (1, 0.55)$, $\mathcal{A}_3^* = (3, 0.95)$, $\mathcal{A}_5^* = (5, 1.15)$, $\mathcal{A}_7^* = (7, 1.35)$, $\mathcal{A}_9^* = (9, 1.45)$, $\mathcal{A}_{11}^* = (11, 1.5)$. This conveys the following message: when the number of channels is small, the control channel should be allocated less bandwidth than a data channel (i.e., $\sigma < 1$), whereas when there are more channels, the control channel should be allocated more bandwidth than a data channel (i.e., $\sigma > 1$). The rationale is that, as the number of channels increases, it becomes easier for a node to secure a data channel for data transmission, and thus fewer nodes will be waiting for free data channels on the control channel. This reduces the chances of having cooperative nodes. In order to counteract this effect, the control channel should be allocated more bandwidth (than equal partition) to increase the probability of successful overhearing (by shortening the vulnerable periods of receiving control packets).

Then we investigate the relationship between σ^* (the optimal bandwidth allocation ratio) and m. In each set of computation, we use (14), (15) and the equation array derived in Section 4, compute the optimal ratio σ^* by search in each series of (σ, p_{co}) for each m. Also, a larger range of m (1..25) and a smaller step size (one) are used. We perform multiple sets of computation with different n and λ corresponding to different network scenarios.

Fig. 12 presents these results. The first observation is

that σ^* monotonically increases with m. This is consistent with the previous series of ($\mathcal{A}_1^*$, $\mathcal{A}_3^*$, ... $\mathcal{A}_{11}^*$). The second observation is that σ^* increases with n (Fig. 12a) but decreases with λ (Fig. 12b). This tells us that, to achieve a high availability of cooperation in a sparse network with heavy traffic, the control channel should be allocated much smaller bandwidth than in a dense network with light traffic. (**Finding 7**)

Fig. 12 also shows that $\sigma^* > 1$ in most cases. This means that, for a DISH network to achieve larger p_{co}, it generally prefers larger bandwidth for the control channel than for *each* data channel. This is contrary to the prior approach of using a smaller frequency band for control ([30], [31]) or dividing total bandwidth equally among all channels (numerous studies) in non-cooperative multi-channel networks. (**Finding 8**)

7 DISCUSSION AND CONCLUSION

Distributed Information SHaring (DISH) represents another dimension of exploiting cooperative diversity, in addition to data relaying, as a distributed flavor of control-plane cooperation. This paper gives the first theoretical treatment of this notion by addressing the availability of cooperation via a metric p_{co}. Through analysis and investigation of the metric under three contexts, i.e., non-cooperative case, ideal DISH, and real DISH, our study demonstrates that p_{co} is a useful performance indicator for DISH networks and bears significant implications. We also apply our results to solving a practical bandwidth allocation problem.

Our study yields eight findings as listed in Section 1, which serve for different purposes. Findings 1 and 2 back up the feasibility and benefit of DISH. Findings 3 and 4 give hints on improving system performance by adjusting packet size, node density and traffic load. Finding 5 demonstrates the significance of the metric p_{co}. Findings 7 and 8 suggest ways of performance improvement from a system design perspective.

In the case of *mobile* ad hoc networks, a node v cannot become a cooperative node with respect to nodes x and y who create an MCC problem, if v fails to decode at least one control message from x and y due to its mobility. However, in most cases when the mobility level is not high, node v will still remain in the intersection region of x and y during the period x and y are sending their control messages, and thus is still able to cooperate. Also, another effect can compensate for the missing of cooperative nodes: a node, although not in the intersection range of x and y, first hears x's control message and then moves into y's range and hears y's control message. In this case, it can also identify the MCC problem and become a cooperative node.

This work attempts to encourage an insightful understanding of DISH, and based on our findings, it demonstrates that p_{co} is a useful metric capable of characterizing the performance of DISH networks, and also bears significant implications. We contend that DISH, as a new cooperative approach, is practical enough to be a part of future cooperative communication networks.
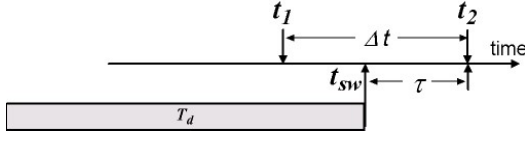
Fig. 13: A node switches to the control channel after data channel handshaking.

Fig. 14: The vulnerable period of v is $[s_i - b, s_i + b]$, in which node $u \in \mathcal{N}_{v\backslash i}$ should not start transmission on the control channel.

APPENDIX A
REMARK ON DATA CHANNEL SOJOURN TIME

In our system model specified in Section 3, we assume that, after switching to a data channel, a node will stay on that channel for T_d, the duration of a successful data channel handshake. This is also valid for a failed data channel handshake. To elaborate, let J_t and J_r be the period of staying on a data channel for a transmitter and a receiver, respectively, and let T_{pkt}, T_{pkt}^{tmo} and T_{pkt}^{det} be the transmission time of a pkt (DATA or ACK), the timeout interval for receiving a pkt, and the interval for detecting pkt collision, respectively. Hence clearly, $T_d = T_{data} + T_{ack}$. For a failed data channel handshake, there are three possible cases: (i) DATA is collided, in which case $J_t = T_{data} + T_{ack}^{tmo}$ and $J_r = T_{data}^{det}$, (ii) ACK is collided, in which case $J_t = T_{data} + T_{ack}^{det}$ and $J_r = T_d$, or (iii) only the receiver switches to the data channel (McCTS fails to reach the transmitter), in which case $J_r = T_{data}^{tmo}$. In all above cases, $J_t \approx J_r \approx T_d = T_{data} + T_{ack}$, because $T_{data}^{tmo} \approx T_{data} \gg T_{ack}^{tmo}$, and $T_{pkt} \approx T_{pkt}^{det}$ (collision detection is typically by checking CRC at the footer of a packet).

APPENDIX B

B.1 Proof of Proposition 1

Lemma 3: If node u is on a data channel at t_1, then the probability that u does not introduce interference to the control channel during $[t_1, t_2]$, where $t_2 - t_1 = \Delta t < T_d$, is given by

$$\Pr[\mathcal{I}_u(t_1, t_2) | \overline{\mathcal{C}_u(t_1)}] = 1 - \frac{\Delta t}{T_d} + \frac{1 - e^{-\lambda_c \Delta t}}{\lambda_c T_d}.$$

Proof: By the total probability theorem,

$$l.h.s. = \Pr[\Omega_u(t_1, t_2)] \times \Pr[\mathcal{I}_u(t_1, t_2) | \Omega_u(t_1, t_2)]$$
$$+ \Pr[\overline{\Omega_u(t_1, t_2)}] \times 1.$$

Let t_{sw} be the time when node u switches to the control channel (see Fig. 13). It is uniformly distributed in $[t_1, t_1 + T_d]$ because the time when u started its data channel handshake is unknown, and hence

$$\Pr[\Omega_u(t_1, t_2)] = \frac{\Delta t}{T_d}. \tag{16}$$

Since control channel traffic is Poisson with rate λ_c,

$$\Pr[\mathcal{I}_u(t_1, t_2) | \Omega_u(t_1, t_2)] = \Pr[\mathcal{S}_u(t_{sw}, t_2) | \Omega_u(t_1, t_2)] = \mathbb{E}[e^{-\lambda_c \tau}]$$

where $\tau = t_2 - t_{sw}$ is uniformly distributed in $[0, \Delta t]$ by the same argument leading to (16). Hence

$$\mathbb{E}[e^{-\lambda_c \tau}] = \int_0^{\Delta t} e^{-\lambda_c \tau_0} \frac{1}{\Delta t} d\tau_0 = \frac{1 - e^{-\lambda_c \Delta t}}{\lambda_c \Delta t},$$
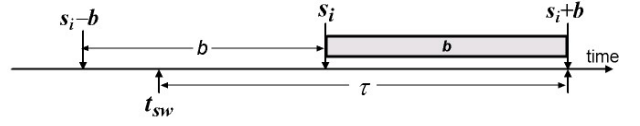
and then by substitution the lemma is proven. □

The proof of Prop. 1 is as below.

Proof: In the case of $u \in \mathcal{N}_{vi}$, no matter u is on the control channel at s_i, or is on a data channel at s_i but switches to the control channel before $s_i + b$, it will sense a busy control channel (due to CSMA) and thus keep silent.

In the case of $u \in \mathcal{N}_{v\backslash i}$, see Fig. 14. Note that the vulnerable period of v is $[s_i - b, s_i + b]$ instead of $[s_i, s_i + b]$, because a transmission started within $[s_i - b, s_i]$ will end within $[s_i, s_i + b]$. Therefore, by the total probability theorem,

$$p_{ni\text{-}oh} = \Pr[\mathcal{C}_u(s_i - b)] \cdot \Pr[\mathcal{I}_u(s_i - b, s_i + b) | \mathcal{C}_u(s_i - b)]$$
$$+ \Pr[\overline{\mathcal{C}_u(s_i - b)}] \cdot \Pr[\mathcal{I}_u(s_i - b, s_i + b) | \overline{\mathcal{C}_u(s_i - b)}]$$
$$= p_{ctrl} \cdot e^{-2\lambda_c b} + (1 - p_{ctrl}) \cdot (1 - \frac{2b}{T_d} + \frac{1 - e^{-2\lambda_c b}}{\lambda_c T_d}),$$

where $\Pr[\mathcal{I}_u(s_i - b, s_i + b) | \overline{\mathcal{C}_u(s_i - b)}]$ is solved by Lemma 3. □

B.2 Derivation of (3)

Proof: Based on the proof for the case $u \in \mathcal{N}_{vi}$ in Prop. 1, it is easy to show that $\mathcal{S}_v(s_i, s_i + b) \Leftrightarrow \mathcal{C}_v(s_i)$. Hence

$$\Pr[\mathcal{S}_v(s_i, s_i + b)] = \Pr[\mathcal{C}_v(s_i)] = p_{ctrl}.$$

Treating events $\mathcal{S}_v(s_i, s_i + b)$ (node v is silent on the control channel) and $\mathcal{I}_u(s_i, s_i + b)$ (node u does not interfere the control channel) being independent of each other, as an approximation, we have

$$\Pr[\mathcal{O}(v \leftarrow i)] \approx p_{ctrl} \prod_{u \in \mathcal{N}_{v\backslash i}} p_{ni\text{-}oh} = p_{ctrl} \, p_{ni\text{-}oh}^{K_{v\backslash i}}.$$

□

B.3 Proof of Proposition 2

Proof: The case of $u \in \mathcal{N}_{ij}$ follows the same line as the proof for Prop. 1. In the case of $u \in \mathcal{N}_{i\backslash j}$, the only difference from Prop. 1 is that now we are implicitly given the fact that i was transmitting McRTS during $[s_j - b, s_j]$. This excludes i's any neighbor u interfering in $[s_j - b, s_j]$. Therefore i's vulnerable period is $[s_j, s_j + b]$ instead of $[s_j - b, s_j + b]$ as compared to Prop. 1. So

$$p_{ni\text{-}cts} = \Pr[\mathcal{C}_u(s_j - b)] \cdot \Pr[\mathcal{I}_u(s_j, s_j + b) | \mathcal{C}_u(s_j - b)]$$
$$+ \Pr[\overline{\mathcal{C}_u(s_j - b)}] \cdot \Pr[\mathcal{I}_u(s_j, s_j + b) | \overline{\mathcal{C}_u(s_j - b)}].$$

Note that we condition on $\mathcal{C}_u(s_j - b)$ instead of $\mathcal{C}_u(s_j)$, because s_j is not an *arbitrary* time due to i's McRTS transmission during $[s_j - b, s_j]$, which leads to $\Pr[\mathcal{C}_u(s_j)] \neq p_{ctrl}$.
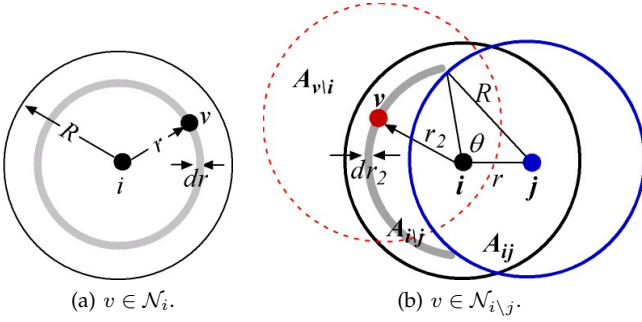
Fig. 15: Deriving the pdf of distance $||vi||$.

First, $\Pr[\mathcal{I}_u(s_j, s_j + b)|\mathcal{C}_u(s_j - b)] = 1$. This is because, as $\mathcal{C}_u(s_j - b) \Leftrightarrow \mathcal{S}_u(s_j - b, s_j)$ which is easy to show, u will successfully overhear i's McRTS, and hence will keep silent in the next period of b to avoid interfering with i receiving McCTS.

Next consider $\Pr[\mathcal{I}_u(s_j, s_j + b)|\overline{\mathcal{C}_u(s_j - b)}]$ where u is on a data channel at $s_j - b$. If u switches to the control channel (i) before s_j, it will be suppressed by i's McRTS transmission until s_j, and thus the vulnerable period of i receiving McCTS is $[s_j, s_j + b]$, (ii) within $[s_j, s_j + b]$, this has been solved by Lemma 3, or (iii) after $s_j + b$, the probability to solve is obviously 1. Therefore,

$$\Pr[\mathcal{I}_u(s_j, s_j + b)|\overline{\mathcal{C}_u(s_j - b)}] = \Pr[\Omega_u(s_j - b, s_j)]\, e^{-\lambda_c b}$$
$$+ \Pr[\Omega_u(s_j, s_j + b)]\, (1 - \frac{b}{T_d} + \frac{1 - e^{-\lambda_c b}}{\lambda_c T_d})$$
$$+ \{1 - \Pr[\Omega_u(s_j - b, s_j)] - \Pr[\Omega_u(s_j, s_j + b)]\} \times 1.$$

According to (16), $\Pr[\Omega_u(s_j - b, s_j)] = \Pr[\Omega_u(s_j, s_j + b)] = b/T_d$. Then by substitution the proposition is proven. $\square$

B.4 Proof of Lemma 2

Proof: Let $A_s(\gamma)$ be the intersection area of two circles with a distance of γ between their centers, and $\gamma < R$, where R is the circles' radius. It can be derived from [32] that

$$A_s(\gamma) = 2R^2 \arccos \frac{\gamma}{2R} - \gamma\sqrt{R^2 - \frac{\gamma^2}{4}}.$$

Let $A_c(\gamma)$ be the complementary area of $A_s(\gamma)$, i.e., $A_c(\gamma) = \pi R^2 - A_s(\gamma)$, and let A_{ij} and $A_{v\setminus i}$ be the areas where $\mathcal{N}_{ij}$ and $\mathcal{N}_{v\setminus i}$ are located, respectively.

(a) See Fig. 15a. Letting $\gamma = ||vi||$ where $v \in \mathcal{N}_i$, and $f(r)$ be its probability density function (pdf), we have $f(r)dr = 2\pi r dr/(\pi R^2)$, which gives $f(r) = 2r/R^2$. Thus

$$\mathbb{E}[A_{v\setminus i}|v \in \mathcal{N}_i] = \int_0^R A_c(r)f(r)dr \approx 1.30R^2,$$

and hence $\mathbb{E}[K_{v\setminus i}|v \in \mathcal{N}_i] \approx n \cdot 1.30R^2/R^2 = 1.30n$.

(b) Let $\gamma_1 = ||vi||$ where $v \in \mathcal{N}_{ij}$, and $f_1(r_1)$ be its pdf. To

solve $f_1(r_1)$, we consider $v \in \mathcal{N}_{i\setminus j}$ instead (see Fig. 15b):

$$\because \mathbb{E}[A_{v\setminus i}|v \in \mathcal{N}_i] = p_1 \cdot \mathbb{E}[A_{v\setminus i}|v \in \mathcal{N}_{ij}]$$
$$+ (1 - p_1) \cdot \mathbb{E}[A_{v\setminus i}|v \in \mathcal{N}_{i\setminus j}]$$
$$\text{where } p_1 \triangleq \Pr[v \in \mathcal{N}_{ij}|v \in \mathcal{N}_i] = \frac{A_s(r)}{\pi R^2},$$
$$\therefore \mathbb{E}[A_{v\setminus i}|v \in \mathcal{N}_{ij}] = p_1^{-1}\mathbb{E}[A_{v\setminus i}|v \in \mathcal{N}_i]$$
$$- (p_1^{-1} - 1) \cdot \mathbb{E}[A_{v\setminus i}|v \in \mathcal{N}_{i\setminus j}]. \quad (17)$$

To determine $\mathbb{E}[A_{v\setminus i}|v \in \mathcal{N}_{i\setminus j}]$, let $\gamma_2 = ||vi||$ where $v \in \mathcal{N}_{i\setminus j}$, and $f_2(r_2)$ be its pdf. It is determined by

$$f_2(r_2)dr_2 = \frac{2(\pi - \theta)r_2 dr_2}{A_{i\setminus j}} \text{ and } \cos\theta = \frac{r_2^2 + r^2 - R^2}{2r_2 r}.$$

Therefore

$$\mathbb{E}[A_{v\setminus i}|v \in \mathcal{N}_{i\setminus j}] = \int_{R-r}^R A_c(r_2)f_2(r_2)dr_2$$
$$= \int_{R-r}^R \frac{2r_2 A_c(r_2)}{A_c(r)}(\pi - \arccos\frac{r_2^2 + r^2 - R^2}{2r_2 r})dr_2.$$

Substituting this and $\mathbb{E}[A_{v\setminus i}|v \in \mathcal{N}_i] \approx 1.30R^2$ (by case (a)) into (17) solves $\mathbb{E}[A_{v\setminus i}|v \in \mathcal{N}_{ij}]$, which we denote by $M(r)$. Then we have

$$\mathbb{E}[K_{v\setminus i}|v \in \mathcal{N}_{ij}] = \frac{n}{R^2}\int_0^R M(r)f(r)dr \approx 1.19n.$$

(c) Proven by noticing that A_{ij} is complementary to the area corresponding to case (a). $\square$

B.5 Derivation of (5)

Proof: Taking the expectation of $\Pr[\mathcal{O}(v \leftarrow i)]$ (given by (3)) over all neighboring (v, i) pairs using Lemma 1 and Lemma 2-(a):

$$p_{oh} \approx p_{ctrl}\, \mathbb{E}[p_{ni\text{-}oh}^{K_{v\setminus i}}] \approx p_{ctrl}\, \exp[-1.30n(1 - p_{ni\text{-}oh})].$$

To solve for p_{succ}, notice that for a control channel handshake to be successful, (i) the McRTS must be successfully received by the receiver, with probability p_{oh}, and (ii) the McCTS must be successfully received by the transmitter, with probability $\mathbb{E}[p_{ni\text{-}cts}^{K_{i\setminus j}}]$ based on Prop. 2 (assuming that $p_{ni\text{-}cts}$ holds for nodes in $\mathcal{N}_{i\setminus j}$ independently, as an approximation). Therefore,

$$p_{succ} \approx p_{oh}\mathbb{E}[p_{ni\text{-}cts}^{K_{i\setminus j}}] \approx p_{oh}\exp[-1.30n(1 - p_{ni\text{-}cts})].$$

$\square$

B.6 Derivation of (8)

Proof: Recall that node v must stay continuously on the control channel during $[s_x, s_y]$. Let $\tau_c = s_y - s_x$ and suppose v switches to a data channel at $s_x + \tau_w$, then we need $\tau_w > \tau_c$. Hence $p_{ctrl}^\star = \Pr(\tau_w > \tau_c)$, where $\tau_c \in [0, T_d]$.

Denote by $f_{\tau_c}(t)$ the pdf of an unbounded τ_c ($s_y \in (s_x, \infty)$). The fact that a MCC problem is created by x and y (at s_y) implies that y missed x's control message (at s_x). This is due to one of the following: (i) y is on the control channel at s_x but interfered, in which case $f_{\tau_c}(t)$ is $\lambda_c e^{-\lambda_c t}$ (ignoring the short interference period which is in
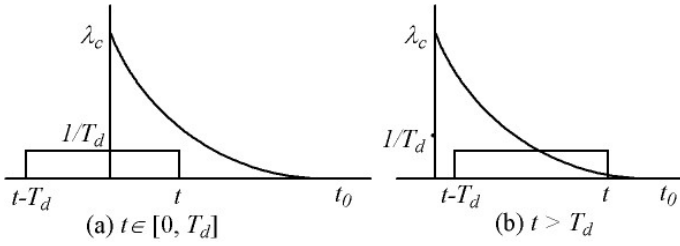
Fig. 16: The convolution of $\frac{1}{T_d}$ $(t \in [0, T_d])$ and $\lambda_c e^{-\lambda_c t}$ $(t > 0)$.

the magnitude of b, while τ_c is in the magnitude of T_d), (ii) y is on a data channel at s_x, in which case y must switch to the control channel before s_y. Again see Fig. 13, where t_1 and t_2 are now s_x and s_y, respectively. Let $\tau_1 = t_{sw} - s_x$ and $\tau_2 = s_y - t_{sw}$, then $\tau_c = \tau_1 + \tau_2$. Note that τ_1 is uniformly distributed in $[0, T_d]$, τ_2 is exponentially distributed with the mean of $1/\lambda_c$, and τ_1 and τ_2 can be regarded as independent. Therefore, $f_{\tau_c}(t)$ is the convolution of $\frac{1}{T_d}$ $(t \in [0, T_d])$ and $\lambda_c e^{-\lambda_c t}$ $(t > 0)$, which can be calculated by referring to Fig. 16, to be $f^d_{\tau_c}(t) =$

$$\frac{1 - e^{-\lambda_c t}}{T_d}[u(t) - u(t - T_d)] + \frac{e^{-\lambda_c t}}{T_d}(e^{\lambda_c T_d} - 1)u(t - T_d).$$

where $u(\cdot)$ is the unit step function.

A weighted sum of the above cases (i) and (ii) gives

$$f_{\tau_c}(t) = w\,\lambda_c e^{-\lambda_c t} + (1 - w)\,f^d_{\tau_c}(t)$$

where w is the weight for case (i). To determine w, note that the probability of case (ii) is $1 - p_{ctrl}$, and the probability of case (i) is $p_{ctrl}(1 - p_{ni-oh}^{K_{y \setminus x}})$ (using (3)) whose mean is $p_{ctrl}(1 - \exp[-1.30n(1 - p_{ni-oh})])$. Therefore

$$w = \frac{p_{ctrl}[1 - e^{-1.30n(1 - p_{ni-oh})}]}{p_{ctrl}[1 - e^{-1.30n(1 - p_{ni-oh})}] + (1 - p_{ctrl})} = \frac{p_{ctrl} - p_{oh}}{1 - p_{oh}}.$$

Finally we compute $p^\star_{ctrl} = \Pr(\tau_w > \tau_c)$ using $f_{\tau_c}(t)$. Recall that $f_{\tau_c}(t)$ is the pdf of an unbounded τ_c but τ_c is in fact bounded within $[0, T_d]$, therefore its actual pdf is $f_{\tau_c}(t)/\int_0^{T_d} f_{\tau_c}(t)dt$. Assuming that τ_w is exponentially distributed with mean $1/\lambda_w$, we have

$$p^\star_{ctrl} = \mathbb{E}_{\tau_c \in [0, T_d]}\,\Pr(\tau_w > \tau_c) = \int_0^{T_d} e^{-\lambda_w t}\frac{f_{\tau_c}(t)}{\int_0^{T_d} f_{\tau_c}(t)dt}dt$$

which reduces to (8). For λ_w, noticing that it is the average rate of a node on the control channel switching to data channels, which happens when a node successfully initiates a control channel handshake via McRTS or sends a McCTS, we have $\lambda_w = \lambda_{rts}p_{succ} + \lambda_{cts}$. □

REFERENCES

[1] E. C. van der Meulen, "Three-terminal communication channels," *Advances in Applied Probability*, vol. 3, no. 1, pp. 120–154, 1971.

[2] T. M. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Infomation Theory*, vol. 25, pp. 572–84, Sept. 1979.

[3] A. Host-Madsen, "Capacity bounds for cooperative diversity," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1522–1544, Apr. 2006.

[4] J. N. Laneman and G. W. Wornell, "Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks," *IEEE Transactions on Information Theory*, vol. 49, no. 10, October 2003.

[5] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity–part I: System description," *IEEE Trans. Commun.*, vol. 51, no. 11, December 2003.

[6] ——, "User cooperation diversity–part II: Implementation aspects and performance analysis," *IEEE Trans. Commun.*, vol. 51, no. 11, Dec. 2003.

[7] H. Zhu and G. Cao, "rDCF: A relay-enabled medium access control protocol for wireless ad hoc networks," in *IEEE Infocom*, 2005.

[8] A. Azgin, Y. Altunbasak, and G. AlRegib, "Cooperative MAC and routing protocols for wireless ad hoc networks," in *IEEE GLOBECOM*, 2005.

[9] S. Moh, C. Yu, S.-M. Park, and H.-N. Kim, "CD-MAC: Cooperative diversity MAC for robust communication in wireless ad hoc networks," in *IEEE ICC*, June 2007, pp. 3636–3641.

[10] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. S. Panwar, "CoopMAC: A cooperative MAC for wireless LANs," *IEEE Journal On Selected Areas In Communications*, vol. 25, no. 2, pp. 340–354, 2007.

[11] T. Luo, M. Motani, and V. Srinivasan, "CAM-MAC: A cooperative asynchronous multi-channel MAC protocol for ad hoc networks," in *IEEE Broadnets*, San Jose, CA, USA, October 2006.

[12] ——, "Cooperative asynchronous multichannel MAC: Design, analysis, and implementation," *IEEE Transactions on Mobile Computing*, vol. 8, no. 3, pp. 338–52, March 2009.

[13] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks," in *I-SPAN*, 2000.

[14] A. Nasipuri, J. Zhuang, and S. R. Das, "A multichannel CSMA MAC protocol for multihop wireless networks," in *WCNC*, 1999.

[15] N. Jain, S. R. Das, and A. Nasipuri, "A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks," in *IEEE ICCCN*, 2001.

[16] A. Adya, P. Bahl, J. Padhye, and A. Wolman, "A multi-radio unification protocol for IEEE 802.11 wireless networks," in *IEEE Broadnets*, 2004.

[17] R. Maheshwari, H. Gupta, and S. R. Das, "Multichannel MAC protocols for wireless networks," in *IEEE SECON*, 2006.

[18] J. Chen, S. Sheu, and C. Yang, "A new multichannel access protocol for IEEE 802.11 ad hoc wireless LANs," in *PIMRC*, 2003.

[19] J. So and N. Vaidya, "Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *ACM MobiHoc*, 2004.

[20] J. Zhang, G. Zhou, C. Huang, S. H. Son, and J. A. Stankovic, "TMMAC: an energy efficient multi-channel MAC protocol for ad hoc networks," in *IEEE ICC*, 2007.

[21] A. Tzamaloukas and J. Garcia-Luna-Aceves, "Channel-hopping multiple access," in *IEEE ICC*, 2000.

[22] ——, "Channel-hopping multiple access with packet trains for ad hoc networks," in *IEEE Device Multimedia Communications*, 2000.

[23] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *ACM MobiCom*, 2004.

[24] T. Luo, M. Motani, and V. Srinivasan, "Analyzing DISH for multi-channel MAC protocols in wireless networks," in *ACM MobiHoc*, Hong Kong, China, 2008.

[25] Y. S. Han, J. Deng, and Z. J. Haas, "Analyzing multi-channel medium access control schemes with ALOHA reservation," *IEEE Trans. on Wireless Comm.*, 2005.

[26] T. Luo, M. Motani, and V. Srinivasan, "Altruistic cooperation for energy-efficient multi-channel MAC protocols," in *ACM MobiCom*, Montreal, QC, Canada, 2007.

[27] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, no. 2, pp. 169–181, 2004.

[28] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Floor acquisition multiple access (FAMA) for packet-radio networks," in *SIGCOMM*, New York, NY, USA, 1995.

[29] J. J. Garcia-Luna-Aceves and C. L. Fullmer, "Floor acquisition multiple access (FAMA) in single-channel wireless networks," *Mobile Networks and Applications*, vol. 4, no. 3, pp. 157–174, 1999.

[30] P. Kyasanur, J. Padhye, and P. Bahl, "On the efficacy of separating control and data into different frequency bands," in *IEEE Broadnets*, October 2005.

[31] X. Yang, N. H. Vaidya, and P. Ravichandran, "Split-channel pipelined packet scheduling for wireless networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 3, pp. 240–257, 2006.

[32] E. W. Weisstein, "Circular segment—from mathworld," http://mathworld.wolfram.com/CircularSegment.html.