Journal of Network and Computer Applications 46 (2014) 113–123

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Efficient broadcasting in slow fading wireless ad-hoc networks

Vesal Hakami, Mehdi Dehghan *

Department of Computer Engineering and Information Technology, Amirkabir University of Technology, 424 Hafez Avenue, Tehran, Iran

ARTICLE INFO

Article history:
Received 17 March 2013
Received in revised form
9 April 2014
Accepted 13 August 2014
Available online 24 August 2014

ABSTRACT

Broadcasting in wireless ad-hoc networks is the dissemination of messages from a source node to all nodes in the network. Since the nodes may have many common neighbors which can receive the same message from multiple forwarders, many schemes have been proposed to achieve high throughput without much redundancy. The majority of the schemes, however, assume either perfect link reliability or a static unreliability regime. In this paper, we consider the case where broadcasting is performed under slow fading, and thus the link qualities can vary over the course of broadcast periods. In this case, a static forwarding scheme based on a fixed probabilistic model of the link qualities cannot track the instantaneous channel conditions, and is prone to blind re-transmissions. Instead, the nodes have to coordinate in real-time so that high coverage is achieved by involving only a subset of nodes experiencing good channel states. To avoid the need for an explicit costly coordination, we model the broadcast problem as a game in which each node is equipped with a regret-based learning strategy. By repeatedly playing this game, the nodes can learn to reach a consensus (equilibrium) in their forwarding strategies for each global channel state. Also, the nodes proactively adapt their strategies so that their collective forwarding behavior actively tracks the broadcast game's equilibrium as it varies with changes in channel states due to slow fading. Simulation results reveal that our solution excels in terms of both the number of transmissions and load distribution, while also maintaining near perfect throughput, especially in dense crowded environments.

1. Introduction

1.1. Research background

Broadcasting plays a key role for disseminating data and topology information from source nodes to all other nodes in wireless ad-hoc networks (WANETs). It is also an underlying operation in many applications such as addressing, paging, publishing services, data gathering, task distribution, alarming, etc. (Basagni et al., 2004). Furthermore, many standard WANET routing protocols use broadcasting for route discovery such as dynamic source routing (DSR) (Broch et al., 1998), ad-hoc on-demand distance vector routing (AODV) (Perkins and Royer, 1999), location aided routing (LAR) (Ko and Vaidya, 2000), optimized link state routing (OLSR) (Clausen et al., 2006), etc. The most naive form of broadcasting is known as flooding, in which each node rebroadcasts a message when receiving that message for the first time. Clearly, even moderate-scale, well-connected networks can be easily crippled by the amount of redundant traffic generated by simple flooding. This is commonly referred to as the *broadcast storm* problem (Ni et al., 1999). To mitigate this problem, several broadcast schemes have been proposed whose origins almost date back to the advent of the ad hoc networks themselves (Jetcheva et al., 2001; Peng and Lu, 1999; Peng and Lu, 2000).

Traditionally, the broadcasting problem has been tackled under the link reliability assumption; i.e., when a node rebroadcasts a message, all nodes within its transmission range can definitely receive the message without error. Within this mindset, various methods have been proposed which can be categorized into two broad classes: structure-based and structure-free. To reduce the number of message forwarders, structure-based methods construct efficient communication substrates such as: spanning tree (Chen and Kao, 2013; Ruiz et al., 2012), connected dominating set (CDS) (Peng and Lu, 1999; Wan et al., 2002; Funke et al., 2006), or clusters (Sivaraman, 2010; Foroozan and Tepe, 2005). In tree-based schemes, the idea is to maintain a maximum leaf spanning tree (MLST) in the network in which internal nodes need to broadcast the message once but leaf nodes do not. In CDS-based methods,

* Correspondence to: Mobile Ad hoc and Wireless Sensor Networks Lab, Department of Computer Engineering and Information Technology, Amirkabir University of Technology, PO Box 15875-4413, 424 Hafez Avenue, Tehran, Iran.
Tel.: +98 2164542749; fax: +98 2166495521.
E-mail addresses: vhakami@aut.ac.ir (V. Hakami), dehghan@aut.ac.ir (M. Dehghan).
URL: http://ceit.aut.ac.ir/~dehghan (M. Dehghan).

http://dx.doi.org/10.1016/j.jnca.2014.08.003
1084-8045/

some nodes are selected to form a virtual backbone. Messages are then disseminated throughout the network by only using the links within this backbone and direct links from other nodes to the backbone. To do so efficiently, the backbone has to form a minimum connected dominating set (MCDS). In cluster-based methods, the broadcasting traffic is divided into internal (flow inside a cluster) and external traffic (flow among the clusters). For internal flooding traffic, cluster-heads are responsible for re-broadcasting but for external ones, border nodes may perform the forwarding function as well. To be efficient, the number of nodes in a cluster should be minimized, which gives rise to an underlying graph-theoretic problem of finding a maximum independent set (MIS). However, computations of MLST, MCDS, and MIS are NP-hard problems, and existing methods only form suboptimal structures based on approximations and relaxations of these notions. Also, to form and maintain these structures the nodes should frequently exchange information with each other which may increase the control overhead packets. In structure-free methods, on the other hand, no explicit communication substrate is constructed. These methods can be loosely classified as: neighbor-designated methods (Liang et al., 2006; Peng and Lu, 2001), and probabilistic broadcasting (Sasson et al., 2003; Naserian and Tepe, 2009; Hu et al., 2012). In neighbor-designated schemes, the node that transmits a message specifies which one of its one-hop neighbors should forward the packet. The most prominent example is the multi-point relaying (MPR) scheme (Liang et al., 2006), in which the nodes use two-hop topological information to select the minimal subset of forwarders from 1-hop neighbors to cover all 2-hop neighbors. In probabilistic approaches, the nodes that receive a broadcast packet retransmit the packet with some probability p or discard (drop) the packet with probability $(1-p)$. Various methods have been suggested for determining this probability. For example, in Naserian and Tepe (2009) and Hu et al. (2012), the broadcast coordination problem has been modeled as a one-shot normal-form game. Each node faces a decision between forwarding a message and generating a benefit; or not forwarding the message to enjoy the benefit to be generated. The forwarding probability of a node is calculated using the number of candidate nodes to forward the message, i.e., the number of nodes that are listening to the transmission, as well as the cost/benefit relation to forward the message by the node. The mixed Nash equilibrium of the game is used to set the nodes' forwarding probabilities. However, in Naserian and Tepe (2009), every node needs to know the number of players of the game, which requires an auxiliary neighbor discovery protocol.

All these aforementioned schemes rely on the assumption of links being reliable; this is while, given the effects of noise, interference, and fading on wireless links, the transmissions succeed probabilistically, and a random number of transmission attempts may be required to correctly deliver messages across a single hop. When dealing with network-wide broadcast, such randomness should be accounted for; otherwise, the network will suffer from blind re-transmissions. In fact, link unreliability introduces a whole new aspect to the broadcast coordination problem which deserves a special treatment. Broadcasting over erroneous links has been mainly investigated by assuming a static unreliability regime; i.e., it is assumed that the link failure probabilities remain fixed over the course of the network operation. Within this perspective, one can distinguish between two major trends in providing for reliable broadcasting: the schemes which only mitigate the implications of link unreliability and those which prescribe a new design to explicitly take into account link failure probabilities. Among the mitigating schemes, the work in Lou and Wu (2007), for instance, provides a double dominating set construct by introducing redundancy into the set of forwarders to ensure better coverage of the non-forwarding nodes. Also, the tree-based method in Banerjee et al. (2003) mainly deals with the acknowledgment (Ack) implosion problem (Foroozan and Tepe, 2005) which arises when a large number Ack messages are sent to ensure reliable delivery, and as a result, the overhead becomes overwhelming. On the other hand, the schemes in Chen and Kao (2013), Moulahi et al. (2012), directly incorporate the expected costs associated with fixed error probabilities of the outgoing links into the broadcast substrate construction. For instance, the study in Moulahi et al. (2012) proposes a connected dominating set construct in which the reception probability of the nodes is considered as a criterion for joining the dominating set. However, the reception probability is assumed to be a priori known lognormal shadowing model. Also, the work in Chen and Kao (2013) presents a distributed scheme for constructing a delivery tree over unreliable links using the notion of potential games. Again, the costs of the links in Chen and Kao (2013) are derived assuming fixed bit error rates based on a known probabilistic model of the wireless connections.

1.2. Motivation and contributions

The static unreliability assumption in prior art is still far from being realistic. In a real-life WANET, link qualities (e.g., measured in terms of signal-to-noise ratio (SNR)) are high and low randomly across the network and evolve with time under the influence of wireless channel fading (e.g., see (Goldsmith, 2005)). Channel fading may occur on a slow or a fast timescale. In this paper, we specifically consider slow fading, under which, channel conditions are subject to sudden and infrequent changes over time. Accordingly, the link failure probabilities and thus the forwarding costs vary with time. Hence, a static rebroadcasting scheme based on a fixed probabilistic model of the transmission costs cannot track the instantaneous channel conditions, and is still prone to blind re-transmissions. On the other hand, the overall cost would be much lower if the nodes dynamically coordinate their forwarding decisions to opportunistically exploit the spatial and temporal diversity of the wireless channels across the network. This is because, a large portion of the network may be coverable by recruiting nodes experiencing good channel conditions, and avoiding those with poor link qualities. However, an explicit coordination mechanism would require that the nodes frequently exchange information (e.g., channel states, neighbor sets, etc.). Moreover, such information needs to be shared in a large neighborhood, since potential forwarders may not necessarily be neighbors themselves, yet their decisions are coupled together due to having common neighbors. Therefore, explicit negotiation can impose considerable communication overhead on the network, which negates its benefit. In this paper, we seek a different alternative by making the following contributions:

- To come up with an implicit highly distributed coordination scheme, we model the broadcast dissemination problem as a dynamic game. In this game, all nodes in the network are players and the game is played every time a new broadcast message is received. Each node's action is simply assumed to be a choice between *forwarding* or *dropping* the current message. The utility of each node is defined as the weighted sum of its local coverage ratio and its forwarding cost, which are both locally measurable without information exchange by other nodes. Given the overlap between the transmission ranges, the achieved coverage ratio for a node is affected not only by its own decision but also by the decision of other potential forwarders. A node's forwarding cost, on the other hand, is taken to be the total number of transmission attempts until successful delivery to its one-hop neighbors. The cost incurred by a node depends on its channel states which evolve randomly with time according to the slow fading process.

- The game's goal is for the nodes to reach a consensus (equilibrium) so that in each period, high coverage is achieved by involving only a subset of nodes experiencing good channel states. Also, this equilibrium has to be maintained dynamically since the channels states may vary over the broadcast periods. To achieve this, we define the set of equilibria of our game in a state-dependent manner. However, instead of Nash equilibrium (NE), we seek to obtain the correlated equilibrium (CE) (Aumann, 1987), as it gives rise to a higher degree of cooperation and better performance compared to the non-cooperative NE (Krishnamurthy et al., 2008; Krishnamurthy, 2011; Wu et al., 2013; Huang and Krishnamuthy, 2011). At CE, the action of each node is a best response to the environmental state and to the estimated actions of other potential forwarders. Therefore, reaching CE can be viewed as formation of a suboptimal consensus amongst the nodes' rebroadcasting decisions.
- In order for the nodes to learn and track the state-dependent CE behavior in the broadcast coordination game, we present a distributed algorithm, namely regret-tracking broadcast (RTB), which is based on the regret-based procedure of Krishnamurthy et al. (2008), Hart and Mas-Colell (2001). In RTB, each node dynamically updates its forwarding strategy to reinforce the actions it regrets not having played enough in the past. Based on the result in Krishnamurthy et al. (2008), Gharehshiran et al. (2013), we argue that if all nodes follow RTB, their collective forwarding behavior tracks the desirable CE-based consensus as it evolves under slow fading. This trackability is obtained without requiring to know the statistics of the channel fading process. The main advantage of our algorithm is that each node only consults its own private history of observations (e.g., achieved coverage ratios, costs, actions) to adjust its forwarding strategy, without having to explicitly negotiate with other potential forwarders. Hence, our proposed scheme satisfies the requirements of broadcast coordination for WANETS: decentralization and low overhead.
- We simulate RTB' operation to evaluate its efficacy. Experimental and comparative results demonstrate RTB's superior performance in terms of both the number of transmissions and load distribution, while also maintaining near perfect throughput in the presence of time-varying link qualities. The contrast between RTB and the literature becomes even more apparent with increasing node density.

The rest of the paper is organized as follows: In Section 2, we present the network model along with the general characteristics of the slow fading process we assume in this paper. In Section 3, we characterize the broadcast dissemination problem under slow fading, and motivate the need for an adaptive low cost broadcast coordination scheme. In Section 4, we present our proposed broadcast coordination game model, and define the channel state-dependent correlated equilibria as the system-wide solution concept. In Section 5, we introduce the RTB algorithm which provides for channel-adaptive forwarding by tracking the broadcast game's equilibria under slow fading. Section 6 is dedicated to the comparative numerical evaluation of the RTB algorithm. We conclude the article in Section 7.

2. System model

In this section, we describe the system model we assume for our broadcast dissemination scheme. After discussing the network model in Section 2.1, we characterize the slow fading channel model in Section 2.2, and subsequently express the dependence of link reception probabilities on instantaneous channel states.

2.1. Network model

We consider a wireless ad-hoc network (WANET) consisting of a set $\mathcal{I}=\{1,2,\ldots,I\}$ of nodes, where I is the cardinality of $\mathcal{I}$. The nodes are equipped with omni-directional antennas, and are distributed arbitrarily. It is assumed that each node knows its 1-hop neighbors. This can be achieved, for instance, by exchanging periodic hello messages. Node j is a neighbor of node i if j is within the transmission range of i. For a given node, say node i, $\mathcal{N}_i$ is defined as the set of all its neighbors. Node i always fails to deliver packets to a node outside its transmission range; on the other hand, packet delivery within transmission range succeeds with a probability. For any two nodes i and j, the reception probability that j can successfully receive a packet sent from i depends on the instantaneous channel state on link ij. The better the current channel state, the greater is the chance for a packet to make it to j correctly. In this paper, we assume that the nodes' channel states vary with time according to a slow fading process. Hence, before expressing the link reception probabilities, we need to characterize the channel model.

2.2. Channel model

In this paper, we assume that the channel states on wireless links are subject to a slow fading effect. Slow fading is a random process that captures long timescale variations in the received signal caused by changes in the environment (e.g., irregular terrains, buildings, foliage and motion in the surroundings) (Goldsmith, 2005). In general, the evolution of fading channels can be modeled as a discrete time finite state Markov chain (FSMC) with time discretized to a given interval (typically the packet transmission time) (Goldsmith, 2005; Wang and Moayeri, 1995). In this model, the signal-to-noise ratio (SNR) range for a link ij, $j\in\mathcal{N}_i$ is discretized into K distinct regions and then mapped into a finite-state space: $S_{ij}=\{1,2,\ldots,K\}$, $\forall j\in\mathcal{N}_i$. More precisely, suppose a set Γ of $(K+1)$ SNR thresholds: $\Gamma=\{\Gamma_1=0,\Gamma_2,\ldots,\Gamma_{K+1}=\infty\}$. Assume the instantaneous SNR associated with the link ij is equal to γ. If γ satisfies $\Gamma_k\le\gamma<\Gamma_{k+1}$, the ij channel is said to be in state k. We use $s_{ij}^n\in S_{ij}$ to denote the channel state of the link ij at time n. Also, denote by $\boldsymbol{s}_i^n=\{s_{ij}^n\}_{j\in\mathcal{N}_i}$ the instantaneous channel state vector of node i over all links with its neighbors. Similarly, we use $\boldsymbol{s}^n\in\mathbf{S}=\times_{i\in\mathcal{I}}\boldsymbol{S}_i$ to represent the global instantaneous channel state in the network, where $\mathbf{S}$ is the channel state composition over all nodes (the symbol '$\times$' denotes Cartesian product). Now, let $\mathbf{P}$ be the stochastic transition matrix representing the evolution law of the slow fading process. In this paper, we assume that the global channel state evolves slowly with time; i.e., if at time n the global state is in $\boldsymbol{s}\in\mathbf{S}$, then at $(n+1)$, it remains with a high probability at $\boldsymbol{s}$, and transitions to another state $\acute{\boldsymbol{s}}\in\mathbf{S},\boldsymbol{s}'\neq\boldsymbol{s}$ with a much lower probability.

Remark 1. we do not assume that the dynamics $\mathbf{P}$ is known to the nodes $i\in\mathcal{I}$.

Next, we express the dependence of link reception probabilities on instantaneous channel states. The bit error rate $BER_{ij}(\gamma)$ on link ij, $j\in\mathcal{N}_i$ is a function of the random SNR γ. Assuming M-ary QAM modulation, $BER_{ij}(\gamma)$ can be calculated as Eq. (1) below (Chung and Goldsmith, 2001)

$$BER_{ij}(\gamma)=0.2\exp\left[\frac{-1.6\gamma}{(M-1)}\right]. \tag{1}$$

Hence, if the channel state on link ij at time n is $s_{ij}^n = k$, then $BER_{ij}^k(s_{ij}^n)$ on link ij is as follows:

$$BER_{ij}^k(s_{ij}^n) = \frac{\int_{\Gamma_k}^{\Gamma_{k+1}} BER_{ij}(\gamma) g(\gamma) d\gamma}{\int_{\Gamma_k}^{\Gamma_{k+1}} g(\gamma) d\gamma}, \tag{2}$$

where, $g(\gamma)$ is the probability density function (PDF) of γ. Assuming that the broadcast message is L bits long, i's transmission to j at time n would succeed with probability $p_{ij}(s_{ij}^n)$, calculated as

$$p_{ij}(s_{ij}^n) = (1 - BER_{ij}^k(s_{ij}^n))^L. \tag{3}$$

Remark 2. To compute $p_{ij}(.)$, a node must know both its channel state and the distribution $g(\gamma)$. This can be a demanding assumption in practical wireless systems. In fact, while it is a reasonable approximation to assume the perfect channel state information at the receiver, usually, the channel state at the transmitter cannot be assumed perfect due to factors such as erroneous or outdated feedback, and frequency offsets between the reciprocal channels. In our proposed scheme (Sections 4 and 5), a node requires neither a priori knowledge of the distribution $g(\gamma)$ (which depends on the type of channel) nor its instantaneous channel state.

3. Problem description

The central theme of this paper is to efficiently manage broadcast message dissemination in a WANET with slow fading channels. We assume that the broadcast flow emanates from a single source node s which periodically sends out broadcast messages to be diffused across the network. Time progresses discretely with index n representing the current broadcast period. Accordingly, $\mathscr{M}_n$ denotes the current message in transit. The global channel state $\boldsymbol{s}^n$ evolves over the broadcast periods according to the slow fading model discussed in Section 2.2. We consider the case where nodes make their forwarding decisions on the fly. For example, Fig. 1 depicts a portion of a WANET where the neighbors of s have each received a copy of $\mathscr{M}_n$ and may decide whether or not to rebroadcast. The dissemination process should be carried out in a reliable fashion. Therefore, if a node chooses to rebroadcast $\mathscr{M}_n$, it keeps re-transmitting until either $\mathscr{M}_n$ is successfully delivered to all its neighbors or the next message $\mathscr{M}_{n+1}$ is received by i, marking the extinction of $\mathscr{M}_n$. Reliable delivery can be achieved, for example, by utilizing ACK-based re-transmission mechanisms (Banerjee et al., 2003; Ros et al., 2012; Lou and Wu, 2003).

In each period, a naive forwarding decision can be made by a node in isolation with respect to considerations such as: the number of its neighbors, its channel states, etc. However, a greater social benefit can be obtained if the potential forwarders negotiate with each other. This is because the forwarding decisions are coupled together due to nodes having common neighbors. These neighbors can receive the same message from multiple forwarders. An uncoordinated broadcast strategy leads to redundant messages and to the waste of network resources. Also, the same nodes may be coverable with much less effort (e.g., by intervention of forwarders experiencing better channel states). Consider period n in Fig. 1 for instance. The quality of the link ij is higher than that of $i'j$, in the sense that on average, it takes fewer transmissions for i to deliver $\mathscr{M}_n$ to j. As long as the channel states remain unchanged, it makes sense for i' to reduce its effort, effectively letting i take over much of the forwarding task. However, as shown in the figure with i and i', potential forwarders may not necessarily be neighbors. Therefore, despite the benefit that can be obtained by negotiation, it also creates control overhead for sharing information such as channel states, neighbor sets, etc. Also, because of slow fading, channel states eventually change with time. As depicted in Fig. 1, at $(n+\tau)$, the signal quality at ij may degrade, while it improves at $i'j$. Such sudden changes necessitate re-negotiations so that the nodes revise their decisions before causing much inefficiency. Since slow fading is a random process, without the luxury of a global change detection mechanism, the nodes have to re-negotiate basically at every period to make effective decisions. This gives rise to a significant overhead which defeats the very purpose of negotiation.

In the next section, we present a game-theoretic approach as one way of circumventing these complications. In our approach, the nodes are interpreted as players in a game, equipped with a multi-agent learning rule as their strategy of play. In each period, rather than engaging in a costly negotiation to determine their joint action, they autonomously make their forwarding decisions according to their learning strategy. The nodes implicitly acquire a coordination signal by inferring the impact of the environmental state and the others' decisions on their realized cost and gain, and accordingly adapt their forwarding strategy in the next period. By repeatedly playing this game over the broadcast intervals, they gradually learn to reach a consensus in their forwarding strategies, which corresponds to a correlated equilibrium of the game they are actually playing. Also, the nodes proactively adapt their forwarding strategies to track the game's equilibrium as the global channel state changes due to slow fading. In the long run, this corresponds to recruiting a team consisting of a subset of forwarders with high coverage level who are also endowed with good channel qualities, and maintaining this team as the channel states change due to slow fading.

In what follows, we first formalize the broadcast coordination problem using a game-based specification. We then define the

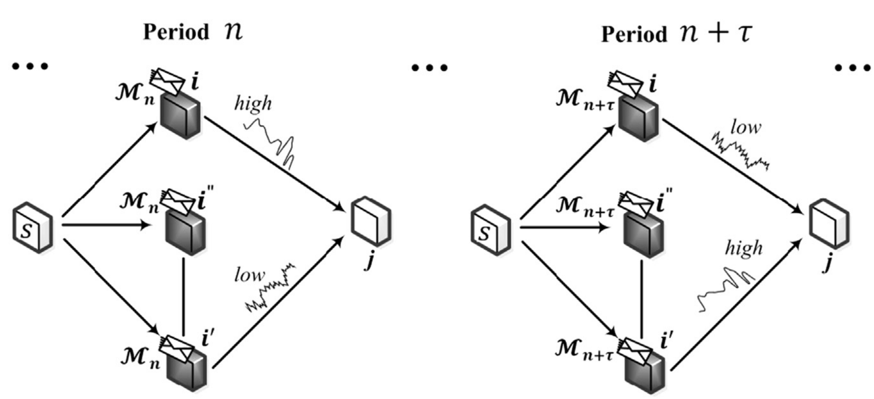

Fig. 1. Nodes i and i' (non-neighbors) both hold similar broadcast messages. They may choose to re-broadcast or not. Node i has j as common neighbor with i'. At time n, the quality of the link ij is higher than that of $i'j$. Conversely, at $(n+\tau)$, $\tau \gg 1$, the signal quality at ij degrades, while it improves at $i'j$.

state-dependent correlated equilibria of the game as the system-wide solution concept we aim to achieve. Finally, in Section 5, we present the regret-tracking broadcast (RTB) algorithm, which provides an adaptive forwarding scheme by tracking the game's equilibria in the presence of slow fading.

4. Broadcast coordination game

We formulate the broadcast coordination problem as a game $\mathcal{G}=(\mathcal{I},\boldsymbol{A},\mathbf{S},(u_i(.))_{i\in\mathcal{I}},(\boldsymbol{\sigma}_i)_{i\in\mathcal{I}})$, which is played in every broadcast period n. We define the components of the game as follows:

(1) ***Set of players***: In $\mathcal{G}$, all nodes in the network are players as denoted by set $\mathcal{I}$. A node $i\in\mathcal{I}$ only knows its immediate neighbor set $\mathcal{N}_i$. It is not explicitly aware of other players.

(2) ***Action set***: We use $a_i^n\in A_i\{0,\ 1\}$ to denote the action of node i at broadcast period n, where 0 represents '*drop*' and 1 represents '*forward*'. Also, let $\boldsymbol{a}^n=\{a_1^n,\ldots,a_I^n\}\in\boldsymbol{A}$ be the composition of the actions from all nodes in period n. $\boldsymbol{A}=\times_{i=1}^{I}A_i$ is the joint action space. Following the common notation in game theory, we use $\boldsymbol{a}_{-i}^n\in\boldsymbol{A}_{-i}$ to denote the joint action taken by all nodes except i.

Remark 3. Node i does not observe the actions $\boldsymbol{a}_{-i}^n$ of other potential forwarders.

(3) ***State***: The symbol $\mathbf{S}$ is the joint channel state space as introduced in Section 2.2. Also as before, $\boldsymbol{s}^n=\{\boldsymbol{s}_1^n,\ldots,\boldsymbol{s}_I^n\}\in\mathbf{S}$ is the global channel state in period n. $\boldsymbol{s}^n$ evolves over the course of broadcast periods according to the slow fading process $\mathbf{P}$.

Remark 4. The nodes cannot observe the global channel state $\boldsymbol{s}^n$. Also, we do not assume that that the evolution law $\mathbf{P}$ of the channel states is known to the nodes.

(4) ***Utility function***: $u_i(a_i^n,\boldsymbol{a}_{-i}^n;\boldsymbol{s}^n)$ is the utility function of node i which depends on both the global channel state as well as the actions of other players. We define the utility function of each node $i\in\mathcal{I}$ as a weighted sum of its forwarding cost and gain: $u_i(a_i^n,\boldsymbol{a}_{-i}^n;\boldsymbol{s}^n)=R_i(a_i^n,\boldsymbol{a}_{-i}^n;\boldsymbol{s}^n)-\alpha.a_i^n.\overline{C}_i(\boldsymbol{s}_i^n)$. Below we describe the components $\overline{C}_i(.)$ and $R_i(.)$:

(4.1) ***Forwarding cost function***: In case $a_i^n=0$, node i incurs zero cost. Therefore, we discuss the case $a_i^n=1$; i.e., when a node i with current channel state $\boldsymbol{s}_i^n$ *intends* to rebroadcast $\mathcal{M}_n$. To guarantee delivery, node i will need to make a total number $C_i(\boldsymbol{s}_i^n)$ of retransmissions until $\mathcal{M}_n$ successfully reaches all of $j\in\mathcal{N}_i$. $C_i(\boldsymbol{s}_i^n)$ is a random variable whose realization depends on i's channel state in period n. Using the link reception probabilities $p_{ij},\forall j\in\mathcal{N}_i$ from (3) in Section 2.2, and the derivation in Chen and Kao (2013), we may express the expected value of $C_i(\boldsymbol{s}_i^n)$ as

$$\overline{C}_i(\boldsymbol{s}_i^n)=1+\sum_{j\in\mathcal{N}_i}\frac{1-p_{ij}(s_{ij}^n)}{p_{ij}(s_{ij}^n)}. \tag{4}$$

$\overline{C}_i(\boldsymbol{s}_i^n)$ can be interpreted as the *average* cost to be incurred by node i for rebroadcasting messages when its channel state is $\boldsymbol{s}_i^n$. Note that the costs $\overline{C}_i(\boldsymbol{s}_i^n),\forall i$ may vary over the course of broadcast periods according to the slow fading model discussed in Section 2.2.

Remark 5. The cost $\overline{C}_i(\boldsymbol{s}_i^n)$ cannot be computed by node i beforehand, since it does not know its reception probabilities $p_{ij},\forall j\in\mathcal{N}_i$. Instead, it actually broadcasts $\mathcal{M}_n$ and obtains the sample value c_i^n by counting the number of (re)transmissions.

(4.2) ***Forwarding gain function***: We use $R_i(a_i^n,\boldsymbol{a}_{-i}^n;\boldsymbol{s}^n)$ to denote the gain function of node i. $R_i(.)$ is defined to be the achieved local coverage ratio for node i by handling message $\mathcal{M}_n$ when global channel state is $\boldsymbol{s}^n$. More specifically, let $\acute{\mathcal{N}}_i^n$ denote the subset of node i's neighbors that are covered by the joint effort of i and other forwarders before the current broadcast period ends. Then, i's gain is defined as: $R_i(a_i^n,\boldsymbol{a}_{-i}^n;\boldsymbol{s}^n)=\frac{|\mathcal{N}'^n_i|}{|\mathcal{N}_i|}$, where $|.|$ denotes the cardinality of a set.

Remark 6. The gain function $R_i(.)$ cannot be computed using a mathematical expression. Instead, it has to be perceived numerically by actual decision making. In practice, node i can count the members of $\mathcal{N}'^n_i$ by listening to the Acks its next-hop neighbors send out for receiving $\mathcal{M}_n$ either from i or from other forwarders. Hence, following our notational convention, we use r_i^n to represent a realized sample of node i's gain in period n. Based on Remarks 5 and 6, the forwarding utility of node i in period n reduces to the realized sample value $u_i^n=r_i^n-\alpha.a_i^n.c_i^n$. Both c_i^n and r_i^n can be measured locally without explicit information of the other potential forwarders, which significantly helps the decentralization of the broadcast coordination. Also, defining the local utilities in this fashion mimics the global objective of establishing the best throughput-cost trade-off.

(5) ***Forwarding strategy***: In each broadcast period n of the game $\mathcal{G}$, a node $i\in\mathcal{I}$ uses a randomized strategy $\boldsymbol{\sigma}_i^n=[\sigma_i^n(a)]_{a\in A_i}$ to choose a forwarding action a_i^n. It then receives a numerical noisy value u_i^n of its utility in that period, an then adjusts $\boldsymbol{\sigma}_i^{n+1}$ according to its history of observations. The game's history through the prism of each node i is a private history h_i^n of length n; i.e., $h_i^n=(a_i^0,\ u_i^0,\ a_i^1,\ u_i^1,\ldots,\ a_i^{n-1},\ u_i^{n-1})\in H_i^n:=(A_i\times\mathbb{R})^n$; hence, each node i selects its actions autonomously according to a map $\boldsymbol{\sigma}_i^{n+1}:\cup_n H_i^n\rightarrow\Delta(A_i)$, where $\Delta(A_i)$ denotes the set of all probability distributions over node i's action set A_i. In fact, a node's history is only a noisy cumulative indicator of the environment and the actions of other forwarders, rather than explicit observation of global events.

With the game now fully specified, we may proceed to introduce the system-wide solution concept we aim to realize. To achieve global consensus in the nodes' forwarding decisions, we focus on an important generalization of Nash equilibrium (NE), known as the correlated equilibrium (CE). Unlike NE, in which each node only considers its own strategy, CE achieves better performance by allowing each node to consider the joint distribution of the nodes' actions. In other words, each node needs to consider others' behavior to see if there are mutual benefits to explore (Wu et al., 2013; Huang and Han, 2010). However, in our forwarding game $\mathcal{G}$, the channel states and accordingly, the nodes' forwarding utilities, evolve according to the slow fading process. Hence, the set of CE should be defined in a state-dependent manner:

Definition 1. (*System-wide solution concept*) Let $\boldsymbol{s}\in\mathbf{S}$ be a global channel state, and define the simplex $\Delta(\boldsymbol{A})=^{\text{def}}\{\boldsymbol{p}\in\mathbb{R}^{|\boldsymbol{A}|};p(\boldsymbol{a})\geq 0,\sum_{\boldsymbol{a}\in\boldsymbol{A}}p(\boldsymbol{a})=1\}$. We denote by $\boldsymbol{\pi_s}\in\Delta(\boldsymbol{A})$ a probability distribution over the joint action space $\boldsymbol{A}$ for state $\boldsymbol{s}$. The state-dependent set of CE of $\mathcal{G}$, denoted by $\mathbb{C}(\boldsymbol{s})$, is defined as (5) below (Krishnamurthy et al., 2008; Gharehshiran et al., 2013):

$$\mathbb{C}(\boldsymbol{s})=\{\boldsymbol{\pi_s}:\sum_{\boldsymbol{a}_{-i}\in\boldsymbol{A}_{-i}}\boldsymbol{\pi_s}(a,\boldsymbol{a}_{-i}).[u_i(b,\boldsymbol{a}_{-i};\boldsymbol{s})-u_i(a,\boldsymbol{a}_{-i};\boldsymbol{s})]\leq 0,\quad \forall a,b\in A_i, i\in\mathcal{I}\}, \tag{5}$$

In words, a CE $\boldsymbol{\pi_s}\in\mathbb{C}(\boldsymbol{s})$ is a probability distribution over the nodes' joint action space which possesses an equilibrium (quiescence) property: If a joint forwarding action $(a, \boldsymbol{a}_{-i})$ is drawn from this distribution (presumably by a trusted third party), and each node $i \in \mathcal{I}$ is told separately its own component a, then it has no incentive to choose a different forwarding action b, because, assuming that all other nodes $i' \in \mathcal{I} \setminus \{i\}$ also obey, the suggested action a is the best in expectation (Papadimitriou and Roughgarden, 2008). Therefore, reaching a CE can be viewed as all nodes being coordinated in their choice of forwarding actions.

In the next section, we present a distributed algorithm to be deployed by each node i to gradually learn its forwarding strategy $\boldsymbol{\sigma}_i$. As will be discussed later, when every node adapts its forwarding strategy using this algorithm, their joint play over time tracks the channel-state dependent set $\mathbb{C}(\boldsymbol{s})$ as it varies due to slow fading.

5. Regret-tracking broadcast (RTB)

In this section, we present RTB, a distributed broadcast coordination algorithm based on the regret-tracking procedure of Krishnamurthy et al. (2008), Gharehshiran et al. (2013)), to shape the nodes' forwarding strategies $\boldsymbol{\sigma}_i, \forall i \in \mathcal{I}$ in real time. By iteratively executing RTB, the nodes' collective forwarding behavior asymptotically tracks the system-wide solution concept $\mathbb{C}(\boldsymbol{s})$ as it evolves under slow fading. In the long run, RTB coordinates the forwarding task by having each node learn when it should be actively forwarding messages and when it should remain silent to avoid inefficiency. In Section 5.2, we discuss RTB's trackability and computational complexity. In Section 5.3, we give some remarks on complications of broadcast coordination in the presence of fast fading together with directions for future research.

5.1. Regret-based forwarding strategy

In RTB, the nodes reinforce the forwarding action they regret not having played enough in the past. In particular, the average regret of node i from $a \in A_i$ to $b \in A_i$ in broadcast period n is defined as follows:

$$R_i^n(a,b) = \left[\sum_{\eta \le n: a_i^\eta = a} \varepsilon (1-\varepsilon)^{n-\eta} [u_i^\eta(b, \boldsymbol{a}_{-i}^n; \boldsymbol{s}^n) - u_i^\eta] \right]^+, \tag{6}$$

where $[.]^+$ denotes max $\{0, .\}$. Eq. (6) has a clear interpretation as a measure of the average *regret* of node i in period n for not having selected action b every time that action a was selected in the past. Note that this averaging is performed in a discounted manner to value more recent utilities higher than more distant utilities. Intuitively, the constant discount factor $0 < \varepsilon \ll 1$ introduces exponential forgetting of the past and permits tracking of a slowly time-varying environment. Another subtlety is that in our broadcast game, node i cannot evaluate its utility function $u_i^\eta(b, \boldsymbol{a}_{-i}^n; \boldsymbol{s}^n)$ for the periods $\{\eta \le n : a_i^\eta = a\}$. This is because over these periods, i has only perceived sample values u_i^η for its actually implemented action a. Instead, we can estimate the average regret using the method in Hart and Mas-Colell (2001) as follows:

$$\hat{R}_i^n(a,b) = \left[\sum_{\eta \le n: a_i^\eta = b} \varepsilon (1-\varepsilon)^{n-\eta} \left[\frac{\sigma_i^\eta(a)}{\sigma_i^\eta(b)} \right] u_i^\eta - \sum_{\eta \le n: a_i^\eta = a} \varepsilon (1-\varepsilon)^{n-\eta} u_i^\eta \right]^+, \tag{7}$$

where $\sigma_i^\eta(.)$ represents the play probability that node i chooses action $a \in A_i$ in period η, and $\hat{R}_i^n(a,b)$ represents the corresponding estimated average regret. In effect, the estimated regret measures the difference (strictly speaking, its positive part) of the average utility over the periods when b was used and the periods when a was used. In addition, the utilities of these periods are normalized in a manner that, intuitively speaking, makes the length of the respective periods comparable.

Based on $\hat{R}_i^n(a,b)$, each node i updates its forwarding strategy as follows: if it chooses action a in period n, then the probability of switching to a different action b in period $(n+1)$ is approximately proportional to the average regret from a to b; with the remaining probability, the same action a is selected again. Therefore, the forwarding action with larger regret in current broadcast period will be selected with higher probabilities in the next period. This way, the average regret of node i for any pair of its choices will diminish over broadcast periods. More specifically, in case action a was used in period n, the play probabilities of node i in period $(n+1)$ are determined as follows, for $b \in A_i, b \ne a$:

$$\sigma_i^{n+1}(b) = \left(1 - \frac{\delta}{n^\rho}\right) \min\left\{ \frac{\hat{R}_i^n(a,b)}{\mu}, 1 \right\} + \frac{\delta}{2n^\rho}, \tag{8}$$

$$\sigma_i^{n+1}(a) = 1 - \sigma_i^{n+1}(b), \tag{9}$$

where $\mu > \hat{R}_i^n(a,b)$ is a normalization constant which can be viewed as update inertia. It is used to keep the sum of play probabilities from exceeding one. $0 < \delta < 1$ is an exploration factor which is essential as nodes continuously learn their utilities. Such exploration forces all actions to be chosen with a minimum frequency, and it rules out the possibility that some action being rarely chosen. Finally, ρ should be less than 0.25 (Hart and Mas-Colell, 2001).

Now that a complete picture of each node's learning strategy is described, we may walk through the complete pseudo-code of RTB in an event-driven style (see Algorithm 1 for pseudo-code and Table 1 for symbols and definitions):

BROADCAST_MESSAGE_RECEIVED: Upon reception of a fresh copy of a broadcast message $\mathcal{M}_n$ in the n-th broadcast period $(n \ge 1)$, each node $i \in \mathcal{I}$ first fires an event indicating the expiration of its previously handled message $\mathcal{M}_{n-1}$ (line 1). In line 2, according to its regret-driven strategy $\boldsymbol{\sigma}_i^n$, node i chooses probabilistically whether or not to re-broadcast $\mathcal{M}_n$. It then calls the Handle_Broadcast() routine in line 3 to proceed with the handling of $\mathcal{M}_n$. Lines 11–15 in Handle_Broadcast() correspond to the case when the node has chosen to forward $\mathcal{M}_n$. It basically keeps retransmitting $\mathcal{M}_n$ until either all $j \in \mathcal{N}_i$ are covered (by i or other forwarders) or the next message is received by i. The number of total transmissions made by i is tracked by the counter c, which is a random variable whose realization depends on i's channel states in period n. Line 17 in Handle_Broadcast() corresponds to when a node has chosen not to forward $\mathcal{M}_n$, it just idly listens to the medium to overhear the Acks from $j \in \mathcal{N}_i$.

$\mathcal{M}_{n-1}$_EXPIRED: Processing this event provides the opportunity to update the parameters of the learning engine. The numerical value of i's payoff u_i^n for $\mathcal{M}_{n-1}$ is computed in line 4. Lines 5–7 are

Table 1
Notations used in regret tracking broadcast (RTB) algorithm.

Symbol	Definition
n	Time index
$\mathcal{M}_n$	Fresh copy of the n-th broadcast message delivered to node i
$\mathcal{N}_i$	Node i's neighbor set
$\mathcal{N}'_i$	Node i's covered set at the end of each broadcast period, $(\mathcal{N}'_i \subseteq \mathcal{N}_i)$
$\boldsymbol{\sigma}_i^n$	Node i's forwarding strategy in period n
a_i^n	Node i's selected action in period n
c	Number of (re)transmissions made by node i in each broadcast period
α	A constant weight factor to balance node i's forwarding gain and cost
u_i^n	Instantaneous utility node i receives in period n
ε	Discount factor, $0 < \varepsilon \ll 1$
μ	Normalization constant (update inertia), $\mu > \hat{R}_i^n(a,b)$
$\hat{R}_i^n(a,b)$	Node i's *regret* for not having played b instead of a

essentially the regret-based routine for updating i's forwarding strategy $\boldsymbol{\sigma}_i^{n+1}$.

ACK_MESSAGE_RECEIVED_OR_OVERHEARD: The firing of this event notifies i that $\mathscr{M}_n$ has been received by a $j \in \mathcal{N}_i$ either through i or other forwarders. It allows i to calculate its local coverage ratio $\frac{|\mathcal{N}'_i|}{|\mathcal{N}_i|}$ which is also a random variable whose realization depends on i's forwarding decision as well as the decisions of its fellow players'.

Algorithm 1. Regret tracking broadcast (RTB) algorithm.

Initialization*:*
$\mathcal{N}'_i := \varnothing$; $c := 0$; $\sigma_i^0(0) = \sigma_i^0(1) := \frac{1}{2}$; $n := 0$; receive $\mathscr{M}_0$;
choose a_i^0 according to $\boldsymbol{\sigma}_i^0$; call Handle_Broadcast($\mathscr{M}_0, a_i^0$);
begin
case (*event*) **do**
BROADCAST_MESSAGE_RECEIVED:
/* Node i has received a fresh copy of $\mathscr{M}_n$, $n \geq 1$. */
1. Fire $\mathscr{M}_{n-1}$_*EXPIRED*;
/* Node i fires an event indicating the expiration of previous message $\mathscr{M}_{n-1}$. */
2. Choose $a_i^n = a$ with probability $\sigma_i^n(a)$;
3. Call Handle_Broadcast($\mathscr{M}_n, a$);
$\mathscr{M}_{n-1}$_*EXPIRED*:
/* $\mathscr{M}_{n-1}$ expires when $\mathscr{M}_n$ is received.*/
4. $u_i^n := \frac{|\mathcal{N}'_i|}{|\mathcal{N}_i|} - \alpha.c$;
5. Use (7) to calculate $\hat{R}_i^n(a, 1-a)$;
6. Use (8) to calculate $\sigma_i^{n+1}(1-a)$;
7. Use (9) to calculate $\sigma_i^{n+1}(a)$;
8. $\mathcal{N}'_i \neq \varnothing$; $c := 0$;
/* Reset covered set and (re)transmission counter for the next period.*/
9. $n := n+1$. /* Update the time index. */
ACK_MESSAGE_RECEIVED_OR_OVERHEARD:
/* Node i is Acked or overhears an Ack for $\mathscr{M}_n$ from a $j \in \mathcal{N}_i$.*/
10. $\mathcal{N}'_i := \mathcal{N}'_i \cup \{j\}$;
end
Function Handle_Broadcast (broadcast_Msg $\mathscr{M}$, action a);
begin
11. **if** $(a=1)$ **then**
12. **while** $(\mathcal{N}'_i \neq \mathcal{N}_i)$ **do**
13. Broadcast $\mathscr{M}$;
14. $c = c+1$;
15. **end while**
16. **else**
17. /* idle listening… */
18. **end if**
end

5.2. Trackability and computational complexity

In this section, we discuss RTB's trackability and computational complexity. As in Krishnamurthy et al., (2008), Krishnamurthy (2011)), a formal definition of the nodes' collective forwarding behavior under RTB is as follows:

Definition 2. (*The nodes' average collective forwarding behavior*) The average collective forwarding behavior of the nodes up to time n is denoted by $\boldsymbol{z}_\varepsilon^n \in \Delta(\boldsymbol{A})$. It is the joint empirical frequency of play up to time n when all nodes $i \in \mathscr{I}$ simultaneously play the broadcast coordination game using RTB (Algorithm 1); i.e., for a joint action profile $\boldsymbol{a} = (a_1, \ldots, a_I) \in \boldsymbol{A}$, $z_\varepsilon^n(\boldsymbol{a}) = \sum_{\eta \leq n} \varepsilon (1-\varepsilon)^{n-\eta}.\mathbb{I}_{\{\boldsymbol{a}^\eta = \boldsymbol{a}\}}$, where $\mathbb{I}_{\{.\}}$ is the indicator function.

It has been shown in Krishnamurthy et al. (2008), Gharehshiran et al. (2013)) that $\boldsymbol{z}_\varepsilon^n$ asymptotically tracks the time-varying set of CE $\mathbb{C}(\boldsymbol{s})$ (see Definition 1). In a Markovian environment, the technical condition that guarantees this trackability is that the underlying Markov chain transitions at infrequent intervals (e.g., if the mean time between transitions is $O(1/\varepsilon)$, see (Krishnamurthy et al., 2008)). This condition is satisfied in our case, since we assumed fading evolves slower than the packet-level timescale. The interpretation of this trackability result is as follows: In the long run, in our broadcast coordination game, at each global channel state $\boldsymbol{s} \in \mathbf{S}$, all nodes do not regret their forwarding strategy $\boldsymbol{\sigma}_i$; i.e., each node responds optimally to the environment and to the actions of other potential forwarders, in the sense that it achieves the largest local coverage ratio and the least average forwarding cost. These locally optimal performances across all the nodes also lead to a considerable sub-optimal network performance. In fact, at the social level, the collective forwarding behavior $\boldsymbol{z}_\varepsilon^n$ at state $\boldsymbol{s}$ corresponds to the desirable consensus $\mathbb{C}(\boldsymbol{s})$ which means that only a subset of nodes with good channel states undertake the forwarding task while the benefit is enjoyed by the whole network. Also, $\boldsymbol{z}_\varepsilon^n$ is agile in tracking the consensus $\mathbb{C}(\boldsymbol{s})$ in anticipation of the slow fading process. This way, the nodes with expectations of poor channel conditions would proactively reduce their forwarding probability and instead rely more on those who enjoy higher quality links; likewise, those who anticipate good channel conditions would take on a more active role to make up for the lack of effort on the part of the nodes experiencing lower link qualities.

RTB's computational complexity: Note that RTB is a particularly lightweight broadcast coordination algorithm. The only step that deserves further discussion is the calculation of $\hat{R}_i^n(a, 1-a)$ in step 5. First, similarly to Gharehshiran et al. (2013)), we re-write (7) as a more efficient recursive formula. This avoids having to compute $\hat{R}_i^n(a,b)$ from scratch in every period, which leads to lower complexity:

$$\hat{R}_i^n(a,b) = \hat{R}_i^{n-1}(a,b) + \varepsilon\left(\left[\frac{\sigma_i^n(a)}{\sigma_i^n(b)} u_i^n(a_i^n).\mathbb{I}_{\{a_i^n = b\}} - u_i^n(a_i^n).\mathbb{I}_{\{a_i^n = a\}}\right]^+ - \hat{R}_i^{n-1}(a,b)\right), \tag{10}$$

Now, at each iteration, RTB needs just a few standard arithmetic operations and comparisons, along with one random number generation to calculate the next action a_i^{n+1}.

5.3. Discussion and directions for future research

In this section, we give a few remarks about two underlying assumptions in this paper, which can also serve as a basis for future research. The first issue is regarding the complications that exist for channel-adaptive forwarding in the face of higher fading rapidity. In RTB, the regret-tracking procedure is an instance of an adaptive filtering algorithm (Kushner and Yin, 2003; Haykin, 2002). Therefore, if the underlying random process changes too fast, then it is not possible to keep track of the time-varying parameters. This is because the dynamics of the underlying Markov chain is not explicitly accounted for in the algorithm. As discussed in Huang and Krishnamuthy (2011), if the timescale of variations matches the iteration index of the algorithm, an alternative approach is to formulate the problem as a stochastic game, and present an algorithm which guarantees convergence to equilibria by explicitly accounting for state evolution from one iteration to the next. However, there are other complications such as the need for observability of the global state, and associated scalability problems due to high dimensionality of the system state. In this paper, we have restricted ourselves to the case of coarse timescale fading, and leave fast timescale fading regimes for future investigation.

The second issue concerns our implicit assumption regarding node cooperation. We assume that the nodes view forwarding as a *task* not a *contribution*, and that the real concern is consensus formation amongst their strategies. In this way, the node that has a better channel condition does not care about the incurred cost, and prefers to contribute to substantially reduce the expenditure of other nodes. Hence, unlike mechanism design (Nisan and Ronen, 1999), here we assume that the nodes are programmable components, and that there is no concern that they are not obedient in executing instructions. This perspective on applied game theory has been more expressively termed as the '*engineering*' agenda in Marden and Shamma (2014). Within this agenda, the local performance measure in each node is artificially defined with intimate relation to the global system objective. The goal is for the nodes to align their decisions so as to achieve decentralized optimization of the system performance at the strategic equilibria of the formulated game. Accordingly, in our case, each node is required to maximize the net utility from its local forwarding gains and costs. This is in line with the global objective of minimizing the number of transmissions while guaranteeing near perfect throughput. That being said, an important direction for future research is to provide for channel-adaptive forwarding, while also accounting for two equally important issues: enforcing cooperative behavior (e.g., see (Srivastava et al., 2005; Li and Shen, 2012; Seredynski and Bouvry, 2013)), and protecting against malicious nodes (e.g., gray hole attacks (Shila and Anjali, 2008)).

6. Performance evaluation

In this section, we simulate the performance of RTB and compare it with prior art. We assume constant packet sizes of length $L = 1000$ bits. Each forwarding node transmits at a constant power of 0.1 W. Although RTB does not depend on any distribution for the channel SNR γ, for the purposes of modeling, we assume a Rayleigh channel in which γ is exponentially distributed with PDF $g(\gamma) = (1/\overline{\Gamma})e^{-(\gamma/\overline{\Gamma})}$, where $\overline{\Gamma} = \mathbb{E}[\gamma]$ is the average SNR. We simulate slow fading channels for each node by discretizing the channel into eight equal probability bins using the quantization procedure described in Section 2.2. The nodes are assumed to have modulation and coding schemes that support a transmission rate of $T = 1$ Mbps for all the links in the network. Each receiving node perceives a random SNR with the received signal also undergoing a random additive white Gaussian noise (AWGN) that leads to a packet error ranging from 5% to 30%. We assume that the nodes operate in a collision-free environment and that they periodically exchange beacon messages to maintain their one-hop neighbor sets. In the simulation runs, 50 nodes are distributed uniformly over a square region with the node density varying from 16 to 160 nodes/km^2. We chose the scaling parameter α in the nodes' utility functions as 0.03 in our implementation because it gave reasonable throughput-cost trade-off results among the values we tried. Table 2 lists the simulation parameters used in our experiments.

Table 2
Simulation parameters.

Parameter	Value
Number of nodes	50
Node density	16–160 nodes/km^2
Packet length	1000 bits
Noise model	AWGN
Transmission power	0.1 W
Transmission rate T	1 Mbps
Modulation	BPSK
Throughput-cost scaling factor α	0.03
RTB's discount factor ε	0.1
Packet origination rate	10 pkts/s

Performance evaluation is done in terms of throughput (network-wide coverage ratio), number of transmissions and the balance in load distribution. The links are generally un-reliable given the influence of noise and the time varying SNRs due to the fading process. Therefore, before the current message gets expired, a node may need to rebroadcast it multiple times until it is delivered to all neighboring nodes. We compare the RTB's performance with three schemes: simple flooding with retransmissions (e.g., RBAV in Alagar et al. (1995) or Ack-flooding in Wong et al. (2013), multi-point relaying (MPR) (Liang et al., 2006) with retransmissions, and the game-based broadcast tree construction (GB-BTC) scheme recently proposed in Chen and Kao (2013).

MPR is an efficient broadcasting scheme based on two-hop topological information, which is used in OLSR routing protocol (Clausen et al., 2006). Each node in the network designates a subset of its one-hop neighbors, called multipoint relays (MPRs), as forwarders to retransmit broadcast packets. Other nodes that are not in MPR set read but do not re-broadcast packets. The MPR set guarantees that all two-hop neighbors of each node receive a copy of the broadcast packets. Therefore, MPR guarantees 100% coverage in a network with reliable links. In order to apply MPR to our unreliable network, we incorporate an explicit Ack mechanism into the protocol operation so that a node retransmits a packet when it does not receive an Ack from any intended receivers within a predefined time. Variants of MPR with retransmission have been considered for instance in Wu et al. (2011).

The GB-BTC scheme (Chen and Kao, 2013), discussed briefly in Section 1.1, uses the notion of potential games to construct a broadcast tree in a distributed fashion. Unlike RTB, the game in Chen and Kao (2013) is played by the successors of each forwarding node with the overall objective of minimizing the number of internal nodes in the constructed tree; in particular, a node's utility for joining a parent on a link increases with the number of nodes selecting the same parent and decreases in proportion to the cost of the link. However, the links' costs in GB-BTC (i.e., the expected number of (re)transmissions) are derived assuming a fixed known probabilistic model. Also, the construction procedure in Chen and Kao (2013) is a one-time task and there is no discussion on how to gracefully maintain the tree structure in response to changes. In fact, the best-response algorithm used in Chen and Kao (2013) would not converge in non-static environments (Rose et al., 2011). Therefore, for the sake of comparison, we have simulated GB-BTC by constructing its tree using link costs derived according to the initial channel states only.

We first investigate the throughput achieved by our proposed RTB method. We allow for unlimited number of (re)transmissions with the lag between subsequent broadcast messages large enough so that 100% throughput is achievable by perfect delivery schemes such as by flooding in a collision-free setting. This would also be the case with MPR and GB-BTC; i.e., they also ensure 100% throughput given their perfect coverage guarantee. Therefore, there is no need to run throughput-wise experiments on these three methods. On the other hand, as shown in Fig. 2, RTB's throughput, when kicking in from a random start, has experienced a short-term drop but has never fallen below 60%. It rises fairly rapidly and asymptotically approaches to the perfect delivery ratio.

Figure 3 shows RTB's performance in terms of the number of transmissions. Each data point in the figure represents the total number of (re)transmissions made by the forwarding nodes in a single network-wide dissemination of the broadcast message. An issue concerns the possibility of a remarkably large number of

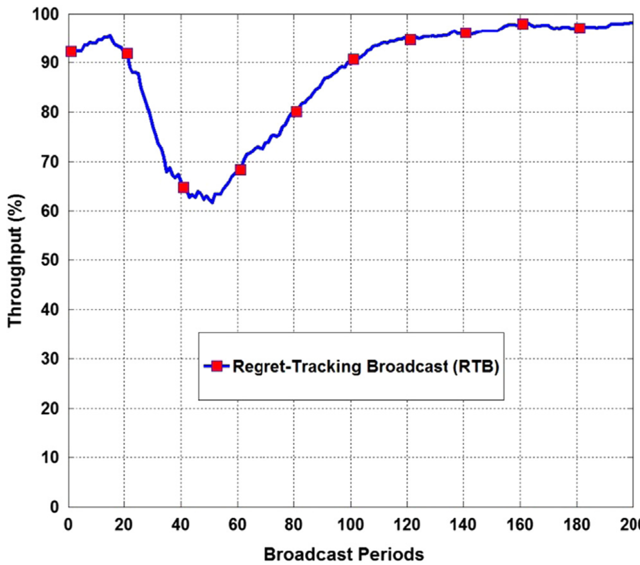

Fig. 2. RTB's throughput; density: 136 nodes/km^2.

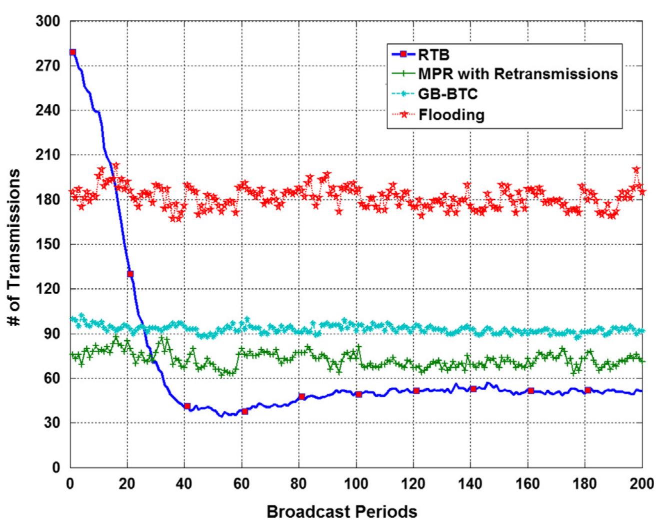

Fig. 3. The number of transmissions in RTB, MPR, GB-BTC, and basic flooding; density: 136 nodes/km^2.

transmissions by RTB at the early stages of the simulation (more than 35% higher than flooding in Fig. 3). Although this may not be the case in general, but the transient forwarding probabilities in RTB might exclude several promising relays from the flow dissemination process, and instead have some unfavorable nodes crudely take over the forwarding task. This is unlikely to happen in flooding given that the unfavorable nodes will soon back off by the intervention of nodes with higher link quality. However, as can be seen, such immature behavior quickly subsides and once converges, RTB outperforms by a margin of over 35% below MPR, the best among the three. It also exhibits more robustness against temporal variations in link qualities. The relatively poor performance of GB-BTC in this experiment can be attributed to the fact that the quality of the links forming its broadcast tree may degrade as the channel conditions vary with time; while in the meantime, it also fails to adaptively utilize higher quality links with smaller BERs.

We also study the impact of the node density on the average number of transmissions made by each of the four schemes. The node density is controlled by adjusting the simulation area while keeping the number of nodes fixed. Each point in Fig. 4 is the

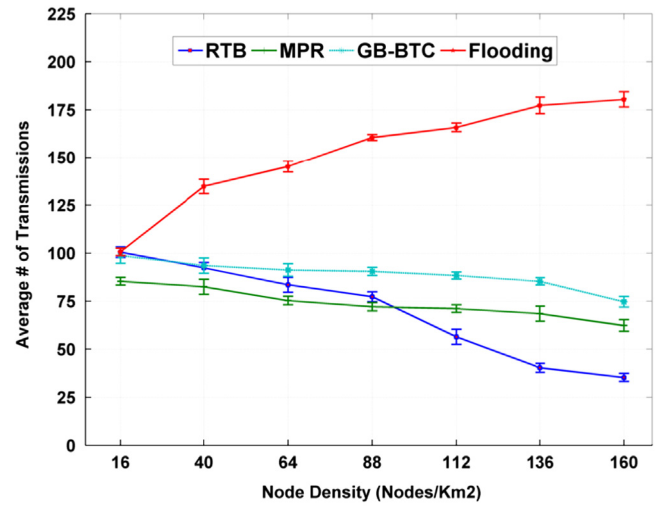

Fig. 4. Average number of transmissions vs. various node densities.

average of 250 simulation runs. We include error bars which indicate 95% confidence that the actual average is within the range of depicted interval. It is seen that the contrast between RTB and the other schemes becomes more apparent with increasing node density. RTB's behavior with respect to node density can be intuitively understood once we draw an analogy with other probabilistic broadcasting schemes in the literature; in effect, in dense networks, many nodes share similar coverage spans. Thus, randomly having some nodes not rebroadcast saves node and network resources without much harming the throughput. In sparse settings, however, there is much less shared coverage; hence, a probabilistic scheme would not guarantee near-perfect throughput unless by setting the forwarding probability higher. When the probability approaches 1, RTB's behavior would resemble that of flooding.

Figure 5 shows the average number of transmissions made by each node in RTB and MPR over 150 simulation runs with 200 time steps per each run. As can be seen, the distribution of load in RTB is significantly more balanced compared to that of MPR. One may argue that MPR is not particularly the most interesting benchmark for this scenario, given that it has no inherent support for periodic (re)appointment of nodes to the relaying role; yet again, this comparison also highlights the fact that these reappointments cannot be effectively done without accounting for channel variations, and to introduce such adaptation into an originally static scheme like MPR takes a total redesign with a methodical approach such as the one we have taken on in this article. We also report in Table 3 the mean absolute deviation (MAD) of the nodes' contributions in RTB, MPR and GB-BTC for various node densities. Compared to GB-BTC, the load in RTB will even up more effectively as the environment becomes more crowded.

7. Conclusion

Broadcasting over erroneous links in WANETS has been mainly investigated by assuming a static unreliability regime; i.e., it is assumed that the link failure probabilities remain fixed over the course of the network operation. In this paper, we considered the broadcasting problem in the presence of slow fading. Under this effect, the channel conditions are good and bad randomly across the network and can vary over the course of broadcast periods. As a result, a real-time coordination scheme is required to engage forwarders with good channel conditions and suppress those with high forwarding costs. To avoid the need for a costly explicit

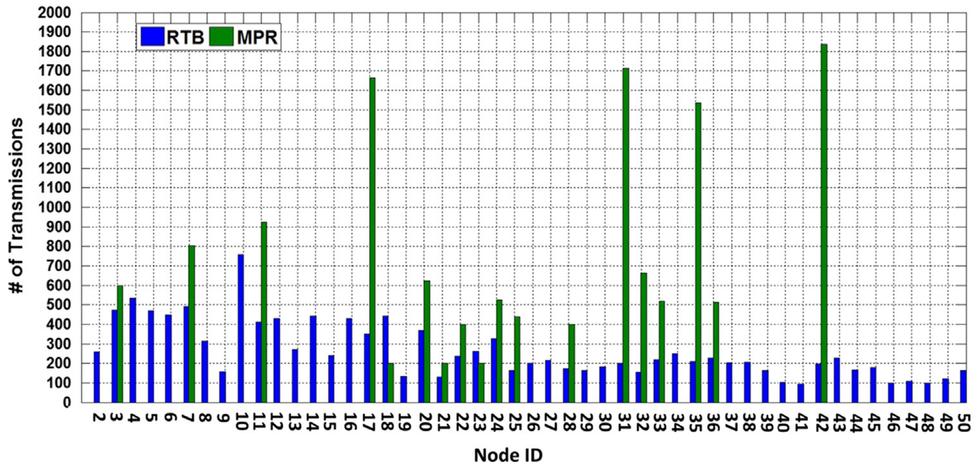

Fig. 5. Broadcast flow distribution in RTB and MPR; density: 136 nodes/km^2.

Table 3
MAD of the nodes' contributions.

	Node density (nodes/km^2)						
	16	40	64	88	112	136	160
RTB	203.28	210.23	156.74	126.99	113.77	98.26	88.06
MPR	207.15	287.92	383.87	329.26	361.36	469.51	282.44
GB-BTC	99.60	106.37	144.91	116.97	116.98	147.19	122.74

coordination, we modeled the broadcast problem as a game with all the network nodes as its players. We presented a distributive algorithm which each node deploys to learn its forwarding strategy. The algorithm determines in each broadcast period whether the node should forward the current message or remain silent. When all nodes play according to this algorithm, their collective forwarding behavior at each global channel state corresponds to a correlated equilibrium (CE); i.e., only a subset of nodes with good channel states forward messages while the benefit is enjoyed by the whole network. The algorithm proactively adapts the nodes' forwarding strategies to maintain this desired behavior as the channel states change due to slow fading. Also, our approach satisfies many informational limitations for deployment in a real-life WANET. In particular, the nodes need neither the knowledge of the statistics of the channel fading process nor the topological information beyond one-hop neighborhood. As evidenced from the numerical results, our channel-adaptive forwarding scheme outperforms its non-adaptive counterparts.

References

Alagar S, Venkatesan S, Cleveland JR. Reliable broadcast in mobile wireless networks. In: Proceedings of the IEEE 1995 military communications conference (MILCOM); 1995. p. 236–40.

Aumann RJ. Correlated equilibrium as an expression of Bayesian rationality. Econometrica 1987;55(1):1–18.

Banerjee S, Misra A, Yeo J, Agrawala A. Energy-efficient broadcast and multicast trees for reliable wireless communication. In: Proceedings of 2003 IEEE wireless communications and networking conference (WCNC); 2003. p. 660–7.

Basagni S, Conti M, Giordano S, Stojmenovi I. Mobile ad hoc networking, New York. 1st ed. IEEE Press/Wiley; 2004.

Broch J, Johnson DB, Maltz DA., The dynamic source routing protocol for mobile ad hoc networks. In: IETF Internet-Draft, draft-ietf-manet-dsr-00.txt; March 1998.

Chen F-W, Kao Jung-Chun. Game-based broadcast over reliable and unreliable wireless links in wireless multihop networks. IEEE Trans Mob Comput 2013;2(8):1613–24.

Chung ST, Goldsmith AJ. Degrees of freedom in adaptive modulation: a unified view. IEEE Trans Commun 2001;49(9):1561–71.

Clausen T, Dearlove C, Jacquet P. The optimized link state routing protocol version 2. In: IETF Internet-Draft, draft-ietf-manet-olsrv2-00; 2006.

Foroozan F, Tepe K. A high performance cluster-based broadcasting algorithm for wireless ad hoc networks based on a novel gateway selection approach. In: Proceedings of the 2nd ACM international workshop on performance evaluation of wireless ad hoc, sensor, and ubiquitous networks; 2005. p. 65–70.

Funke S, Kesselman A, Meyer U, Segal M. A simple improved distributed algorithm for minimum CDS in unit disk graphs. ACM Trans Sens Netw 2006;2:444–53.

Gharehshiran ON, Krishnamurthy V, Yin G. Distributed tracking of correlated equilibria in regime switching noncooperative games. IEEE Trans Autom Control 2013;58(10):2435–50.

Goldsmith A. Wireless communications. Cambridge University Press; 2005 p. 74.

Hart S, Mas-Colell A. A reinforcement procedure leading to correlated equilibrium. In: Debreu G., Neuefeind. W., and Trockel W., (Eds.), Economic essays; 2001. p. 181–200.

Haykin S. Adaptive filter theory. 4th ed Prentice-Hall; 2002.

Hu X, Wang C, Song X, Wang J. Stability-based RREQ forwarding game for stability-oriented route discovery in MANETs. Wirel Personal Commun 2012:1–17.

Huang J, Han Z. Game theory for spectrum sharing. In: Zhang Y., Zheng J., and Chen H-H., (Eds.), Cognitive radio networks: architectures, protocols, and standards. CRC Press; 2010. p. 291–317

Huang JW, Krishnamuthy V. Cognitive base stations in LTE/3GPP femtocells: a correlated equilibrium game-theoretic approach. IEEE Trans Commun 2011;59(12):3485–93.

Jetcheva J, Hu Y, Maltz D, Johnson D. A simple protocol for multicast and broadcast in mobile ad hoc networks. Internet Draft: draft-ietf-manet-simple-mbcast-01.txt; July 2001.

Ko YB, Vaidya NH. Location-aided routing (LAR) in mobile ad hoc networks. Wirel Netw 2000;6(4):307–21.

Krishnamurthy V. Networks of biosensors: decentralized activation and social learning. Eur J Control 2011;5(6):526–46.

Krishnamurthy V, Maskery M, Yin G. Decentralized adaptive filtering algorithm for sensor activation in an unattended ground sensor network. IEEE Trans Signal Process 2008;56(12):6086–101.

Kushner HJ, Yin G. Stochastic approximation algorithms and recursive algorithms and applications. 2nd ed. New York: Springer-Verlag; 2003.

Li Z, Shen H. Game-theoretic analysis of cooperation incentive strategies in mobile ad-hoc networks. IEEE Trans Mob Comput 2012;11(8):1287–303.

Liang O, Sekercioglu YA, Mani N. A Survey of multipoint relay-based broadcast schemes in wireless ad hoc networks. IEEE Commun Surv Tutor 2006;8(4):30–46.

Lou W, Wu J. A reliable broadcast algorithm with selected acknowledgements in mobile ad hoc networks. In: Proceedings of 2003 IEEE Globecom; 2003. p. 3536–41.

Lou W, Wu J. Toward broadcast reliability in mobile ad hoc networks with double coverage. IEEE Trans Mob Comput 2007;6:148–63.

Marden JR, Shamma JS. Game theory and distributed control. In: Young P, Zamir S, editors. Handbook of game theory, vol. 4. Elsevier Science; 2014.

Moulahi T, Nasri S, Guyennet H. Broadcasting based on dominated connecting sets with MPR in a realistic environment for WSNs & ad hoc. J Netw Comput Appl 2012;35(6):1720–7.

Naserian M, Tepe K. Game theoretic approach in routing protocol for wireless ad hoc networks. Ad-Hoc Netw 2009;7(3):569–78.

Ni S-Y, Tseng Y-C, Chen Y-S, Sheu J-P.. The broadcast storm problem in a mobile ad hoc network. in: Proceedings of the ACM MobiCom; 1999. p. 151–62.

Nisan N, Ronen A. Algorithmic mechanism design. In: Proceedings of the 31st annual ACM symposium on theory of computing; 1999. p. 129–40.

Papadimitriou CH, Roughgarden T. Computing correlated equilibria in multi-player games. J ACM 2008;55(3) 14:1, 14:29.

Peng W, Lu X., Efficient broadcast in mobile ad hoc networks using connected dominating sets J Softw, 1999.

Peng W, Lu, X. On the reduction of broadcast redundancy in mobile ad hoc network. In: Proceedings of MOBIHOC; 2000.

Peng W, Lu X. AHBP: An efficient broadcast protocol for mobile ad hoc networks. J Comput Sci Technol 2001;16(2):114–25.

Perkins C, Royer E. Ad hoc on-demand distance vector routing. In: Proceedings of the 2nd IEEE workshop on mobile computing systems and applications; 1999. p. 90–100.

Ros FJ, Ruiz PM, Stojmenovic I. Acknowledgment-based broadcast protocol for reliable and efficient data dissemination in vehicular ad hoc networks. IEEE Trans Mob Comput 2012;11:33–46.

Rose L, Lasaulce S, Perlaza SM, Debbah M. Learning equilibria with partial information in decentralized wireless networks. IEEE Commun Mag 2011;49 (8):136–42.

Ruiz P, Dorronsoro B, Bouvry P, Tardon L. Information dissemination in VANETs based upon a tree topology. Ah-Hoc Netw 2012;10(1):111–27.

Sasson Y, Cavin D, Schiper A. Probabilistic broadcast for flooding in wireless mobile ad hoc networks. In: Proceedings of 2003 IEEE wireless communications and networking conference (WCNC), vol. 2; 2003. p. 1124–30.

Seredynski M, Bouvry P. Analysing the development of cooperation in MANETs using evolutionary game theory. J Supercomput 2013;63(3):854–70.

Shila DM, Anjali T. A game theoretic approach to gray hole attacks in wireless mesh networks. In: Proceedings of 2008 IEEE military communications conference (MILCOM); 2008. p. 1–7.

Sivaraman E. Dynamic cluster broadcasting for mobile ad hoc networks. In: Proceedings of the 2010 international conference on communication and computational intelligence (INCOCCI); 2010. p. 123–7.

Srivastava V, Neel J, MacKenzie AB, Menon R, DaSilva LA, Hicks JE, et al. Using game theory to analyze wireless ad hoc networks. IEEE Commun Surv Tutor 2005;7 (4):46–56.

Wan P-J, Alzoubi KM, Frieder O. Distributed construction of connected dominating set in wireless ad hoc networks. In: Proceedings of the IEEE infocom; 2002. p. 1597–604.

Wang HS, Moayeri N. Finite-state markov channel – a useful model for radio communication channels. IEEE Trans Veh Technol 1995:473–9.

Wong GKW, Liu H, Chu X, Leung Y-W, Xie C. Efficient broadcasting in multi-hop wireless networks with a realistic physical layer. Ad Hoc Netw 2013;11:1305–18.

Wu C, Ohzahata S., Kato T. A broadcast path diversity mechanism for delay sensitive VANET safety applications. In: Proceedings of the IEEE vehicular networking conference (VNC), 2011. p. 171–6.

Wu D, Cai Y, Zhou L, Zheng Z, Zheng B. Cooperative strategies for energy-aware ad hoc networks: a correlated-equilibrium game-theoretical approach. IEEE Trans Veh Technol 2013;62(5):2303–14.