FAIR: Fee Arbitrated Incentive Architecture in Wireless Ad Hoc Networks

Alvin Mok, Bina Mistry, Eric Chung, Baochun Li

***Abstract*— In ad hoc wireless networks, nodes communicate through a cooperative network in which other nodes function as relays. Since resources are frequently constrained, incentives must be provided to entice nodes to relay. The correct amount of incentive is essential to the efficient and optimal operation of the network. Excessive incentive results in widespread altruism, leading to diminished system lifetime. In contrast, insufficient incentive leads to selfish nodes that dramatically decrease network throughput. Nodes are assumed to be rational, and seek to maximize their own utilities. In this paper, we use virtual credits as the incentive to stimulate cooperative behavior between nodes. The result of this research is Fee Arbitrated Incentive Architecture (FAIR), an application layer protocol that enhances fairness and collaboration in ad hoc wireless networks. FAIR utilizes autonomous feedback mechanisms to configure itself to the changing demands of the nodes, users, and network conditions. *PriceSim*, a network simulator implementing the FAIR protocol, was created to evaluate FAIR's performance.**

I. INTRODUCTION

Wireless ad hoc networks provide ubiquitous communication and collaboration without the burden of wires or pre-established infrastructure. Since each node has limited transmission power, packets from one node may be unable to arrive at their intended destination without assistance of intermediate nodes. The network fabric is established by nodes wirelessly forwarding packets on behalf of each other. We assume nodes are rational and belong to different organizations that do not have an intrinsic reason to cooperate. Without a central authority, there is no enforcement of cooperation between nodes, leading to potentially selfish behavior. In fact, given that node's resources (energy, processor and bandwidth) are limited, nodes have a disincentive to cooperate because they can maximize their own utility by only sending packets and refusing to forward other's packets.

In such contexts, the interest of an individual node is in conflict with the system-wide throughput. A node may extend its operating lifetime by refusing to forward all packets. However, if all nodes behave similarly, the ad hoc network will revert back to a single hop network and system throughput will degrade dramatically. *Incentives* are therefore required to entice nodes to cooperate for the common good of all network participants.

In this paper, we design and propose a Fee Arbitrated Incentive Architecture (FAIR), an middleware-layer framework and algorithm to promote cooperation in wireless ad hoc networks. FAIR uses exchangeable virtual credits to provide incentive to the nodes in service, and to charge nodes that require service. Such exchanges of credits between the consumers and providers of services promote collaboration among inherently non-cooperative nodes. One of the innovative features of our algorithm is an active feedback mechanism that dynamically tunes pricing parameters to adapt to different topologies and respond to various network changes (e.g., node mobility and changes in traffic patterns). To minimize overhead, FAIR makes all its decisions based on local information alone and does not need any specialized protocols to exchange state information aside from charging for packet forwarding.

In FAIR, when a node wishes to send a packet, it first checks if its internal resources will support this packet. It then determines the next hop node by examining the previous interactions to predict the packet forwarding cost. Once an intermediate node has been selected, the packet is forwarded and the sender is charged for the forwarding effort. These charges, which are calculated by the intermediate node, are then incorporated into the estimation table within the sender to refine its estimates. This process repeats hop by hop until the packet reaches the final destination. The feedback loop continuously monitors changes in each node's state and changes its pricing variables accordingly.

FAIR seeks to address two main issues — fairness and collaboration. Fairness implies the amount of benefit that a node can derive from the system is proportional to its contribution to the system; collaboration involves delivering a packet from source to destination if any route exists. These two goals are non-trivial to achieve for the following reasons. First, the nodes form a distributed system, with global states inaccessible to the individual nodes. Hence, the nodes can only use local and a limited amount of regional information to tune their performance. However, the end goals of the system — fairness and collaboration — are defined globally. Second, inherent tradeoffs between fairness and collaboration make it difficult to achieve both simultaneously. To enforce fairness, the benefit to contribution ratio needs to be upheld much more rigidly, potentially leading to the stagnation of some packets. Likewise, to enhance collaboration, various aspects of fairness need to be relaxed. Third, the mobility and transient links of ad hoc networks make the formation of alliances and permanent partnerships between nodes impossible. All decisions need to be made locally with concern only for immediate gain instead of long term relationships. In our proposed algorithms, we seek to achieve both goals with the autonomous feedback mechanism, by allowing a node to respond to the local,

---

Alvin Mok, Eric Chung and Baochun Li are affiliated with the Department of Electrical and Computer Engineering, University of Toronto. Their email addresses are $\{$*alvin,eric,bli*$\}$*@iqua.ece.toronto.edu*. Bina Mistry is affiliated with Department of Computer Science, University of Toronto. Her email address is *bina.mistry@utoronto.ca*.

regional and global system changes to enhance both fairness and collaboration and maintain a balance between them.

The original contributions of this paper are as follows. First, we present FAIR, an algorithm that provides incentive to selfish, autonomous mobile nodes to collaborate, with a focus on promoting both fairness and collaborations, with a balanced tradeoff. Second, we introduce the feedback mechanism to dynamically tune the pricing variables so the forwarding price reflects the state of the wireless network (as perceived by each node). Third, results from *PriceSim*, a simulator that implements various pricing algorithms, are presented to show the effectiveness and superiority of FAIR and the feedback mechanism. To the best of our knowledge, this is the first paper applying a feedback mechanism within the pricing strategies, by adjusting variable weights in real-time to promote cooperative behavior between nodes.

The remainder of the paper is organized as follows. Sec. II presents the preliminaries and the system model that serve as a basis for the FAIR algorithm and the subsequent PriceSim simulation. Sec. III details the basic and feedback-enhanced FAIR algorithm design. In Sec. IV, we conduct extensive performance evaluations of FAIR using our *PriceSim* simulator. Finally, we discuss related work in Sec. V and conclude the paper in Sec. VI.

II. PRELIMINARIES

In this work, we consider an ad hoc wireless network with a finite number of nodes, N, distributed in various topological configurations. All nodes are associated with multiple resource constraints, the usual candidates include the energy constraint (E), the bandwidth constraint (B) and the processor constraint (P). All the constraints in the system are of two classes — renewable and non-renewable. Bandwidth and processor are renewable resources, since they become replenished after each discrete interval; energy is non-renewable since it is a non-increasing function over time. Without loss of generality, we consider three resource constraints of $\{E_i, B_i, P_i\}$, on node n_i $(i < N)$ in the system.

Each node n_i resides at a location and is associated with a range of transmission. For n_i to send packets to n_j, n_i must be in n_j's range. Note that the links can be asymmetrical due to differences in the range of transmission. For a node's attempt at transmission to be successful, there must be other nodes in range.

Each node has an associated mobility model. The random waypoint mobility model is chosen for performance evaluation because of its simplicity and applicability in ad hoc wireless networks, where nodes frequently have a set destination and migrate towards that location. For the sake of simplicity, in our study, we assume the degree of node mobility is sufficiently low to permit the routes to be stable for several discrete periods. This assumption is realistic in most cases since many packets would have been successfully transmitted during the time that the node is in range with a certain set of nodes. This assumption simplifies routing since nodes do not need to rediscover the paths every period.

A node expends its resources in transmitting, receiving and processing packets. A node must have sufficient resources to successfully send its queued packets. To simplify the system model, we associate all the costs with the transmitting action. In the *PriceSim* simulations, the resources $\{E_i, B_i, P_i\}$ required to send a packet from node n_i increase linearly with the size of the packet. While this may not be a very realistic assumption, it allows us to concentrate on the salient parts of the problem.

In addition to its available resources, a node maintains a "wallet" of virtual credits at any given time. Nodes exchange virtual credits to request and obtain packet forwarding services. A node requires sufficient virtual credits to send a packet. Charging and pricing are performed in a purely distributed manner, with each node trying to maximize its own utility U_i, which can be simply interpreted as the number of packets it may send. Forwarding efforts are charged per hop and each intermediate node uses only local information to derive the pricing and the next hop node.

III. FAIR: ALGORITHMS AND ANALYSIS

A. On Fair and Collaborative Systems

The three defining properties of a *fair* system are proportionality, accountability and autonomy. Proportionality suggests that a node's ability to send packets should be proportional to its contribution to the overall system. This definition is reasonable because it discourages a node from becoming a "parasite" in the network by spending resources in other nodes without providing anything in return. As in the "Prisoners' Dilemma", a node will be able to gain some short term benefit by cheating. However, since this forwarding "game" is repeated over many periods, a well designed algorithm will ensure that it is not in the node's interest to "cheat" or be selfish. Accountability implies that the costs for packet transmissions should be related to the resources required to forward the packet. Given that the root cause of the conflicts between nodes arises from the scarcity of resources, this second property guarantees that the system is resource utilization motivated. This resource utilization sensitivity also allows different nodes to leverage their strengths to compensate for their weakness (*i.e.*, a slow node but with a powerful energy can still contribute to the system because it can afford to more frequently route other node's packets, albeit slower). Finally, autonomy implies that a node is free to choose its forwarding behavior based on its own needs (*e.g.*, if a node does not need to send any packets, it should not be forced to forward other node's packets). Extending this concept, a node's propensity to contribute should be propositional to its own demand to send packets.

The two properties of a *collaborative* system are necessity and sufficiency. Necessity means that a node will only accumulate as much virtual credit as it needs to send its packets. Since there is only a limited supply of virtual credits in the system, it is important that nodes do not become credit "sinks". Otherwise, other nodes will not have an adequate amount of credits to send their packets. Sufficiency entails a node to have enough credits for its future packet sending requirements. This ensures that credits become a vehicle of communication and cooperation between nodes instead of a constraint that limits the sending abilities of a node even when the nearby region has sufficient capacity.

B. FAIR: an Overview

The Fee Arbitrated Incentive Architecture (FAIR) is a middleware architecture and a distributed algorithm designed to enhance the fairness and collaboration of an ad hoc wireless network by virtual credits pricing and dynamic feedback. Its two main functions are: (1) to determine the optimal next hop for packet forwarding with the help of lower layer routing protocols; (2) to determine the "price" to charge the packet originator (hop by hop) for the forwarding service.

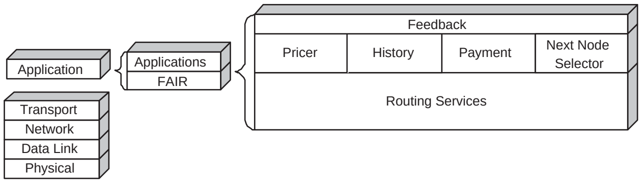


Fig. 1. FAIR: Fee Arbitrated Incentive Architecture

Fig. 1 shows the architecture of FAIR, as a middleware layer beyond transport protocols. The components of FAIR are explained as follows.

First, the *Routing Services* component masks the differences between various lower layer routing protocols and provide a list of next hop candidates to the upper layers for assessment. FAIR relies on the next hop reachability and statistics from the lower layer routing protocols. As a middleware architecture, FAIR does not implement its own routing protocol but relies on whatever routing protocol that already exists in the network protocol stack. The specific routing protocol in use must be able to present a list of possible next hop candidates to *Routing Services*.

Second, the *Pricer* component determines the price for forwarding a particular packet. One of the most difficult problems in the creation of an incentive mechanism is the correct pricing model for packet forwarding. If the model is too greedy (*i.e.*, nodes charge too much for their services), traffic in the network will grind to a halt because nodes are unwilling or unable to pay other nodes for forwarding; if the model is too conservative (*i.e.*, nodes charge too little for their services), traffic volumes will likewise decrease because the intermediate nodes will not have sufficient incentives to forward packets. It is imperative that the right balance of incentive (for intermediate nodes) and affordability (for sender) be maintained at all times.

Third, the *Feedback* component dynamically tunes various pricing parameters to maximize both fairness and collaborativeness in the global system. With the assistance of the *Feedback* component, the algorithms deployed in the *Pricer* become adaptive algorithms that depend on observations of the environment.

Finally, the *Pricer* and *Feedback* components are assisted by several additional helper components. The *Next Node Selector* component selects the next node to forward the packet to, based on homogenized information returned by different routing protocols from *Routing Services*; the *Payment* component pays (charges) other nodes for forwarding packets, while the *History* component maintains a record of all the node's transactions to aid in pricing and next node selection. Note that we do not include a "quotation" mechanism in FAIR because of the high overhead in the exchange of such messages. However, this implies that the originator node is unaware of the forwarding cost for a packet until after the packet has been sent and the bill received. In the absence of a "quotation" mechanism, histories are essential for making the decision of the next hop node.

The above components cooperate to achieve the two principle functions of FAIR: Optimal Next Hop Determination and Pricing Decision. Optimal Next Hop Determination first uses *Routing Services*, which offers several viable next hop nodes possibilities. *Next Node Selector* examines the possibilities and consults the *History* component to arrive at estimates for the cost of sending the packet through the various nodes. The next hop node with the lowest virtual credit cost is usually selected as the Optimal Next Hop. However, to allow exploration of alternate lower cost paths, previously untried nodes will be favored over tried nodes. Moreover, *History* uses a soft state mechanism to "retire" old entries for keeping the estimates fresh. Next hop nodes with expired histories will be interpreted as previously untried, which allows for rediscovery.

Pricing Decision is the other function of FAIR which predominantly relies on the *Pricer* and *Feedback* components. The *Pricer* examines the various resource requirements of the packet, the external environment, and the next node's charges to price the forwarding effort. The estimate of the next node's charges are derived from the *History* component. *Feedback* is used extensively to adapt the weightings of the factors to the current network and node conditions. Unlike other approaches that allow virtual credits to be purchased externally [1], the amount of credits in the FAIR system is conserved — *i.e.*, a node has no other way to earn credits other than providing service to its neighbors. We feel that this constraint not only makes the problem more interesting, but also results in a fairer system, one that is impossible to "buy" one's way out of service. Without the option of externally acquiring the credits, proper pricing becomes even more important because there is a low tolerance for error. If pricing strategies are too generous or stringent, many nodes will be unable to send packets, thus greatly impairing the operations of the network.

C. FAIR: baseline algorithms

The intuition behind FAIR is as follows. At the first glance, it would appear that nodes have the incentive to charge outrageously high prices to maximize their profits; however, the free market forces keep this behavior in check. Under the "free market economy", excessive pricing will cause the node's "customers" to choose other nodes as the forwarder. To ensure a steady revenue stream of credits (for use in the node's own sending), it is in the interest of the node to price accordingly to stimulate the appropriate level of demand and earn a normal profit. This is exactly the goal of the FAIR pricing model, which reflects both the internal costs and the external costs for forwarding a packet. The internal forwarding costs refer to the resources expended in the effort and should be taken into account to give nodes an incentive to cooperate. The external costs for forwarding include the charges from the next hop, the opportunity cost for forwarding other's packets and the desire to make a normal profit. A node should never be subjected to a situation where it needs to forward packets at a loss because this negative pricing model will cause a node to lose all incentives in forwarding other nodes' packets.

The internal forwarding costs refer to the costs that node n_i incurs for forwarding the packet. FAIR takes the energy e^k, bandwidth b^k, and processor p^k required for the forwarding of packet k into consideration. Intuitively, the more resources the packet forwarding requires, the more the node should charge for the forwarding effort. For node n_i, the price contribution from the internal costs is $\alpha_i b^k + \beta_i e^k + \gamma_i p^k$, where $\alpha_i, \beta_i, \gamma_i$ are pricing weighting variables for n_i.

The external forwarding costs model the interaction between the node and its neighbors. The simplest one is $x^k_{(i,j)}$, representing n_i's estimate for the next hop n_j's forwarding charges for packet k. To prevent a node from losing virtual credits from the forwarding effort, the estimate provided by the *History* component must be based on recent transactions and the changes in the target node's prices must be gradual, allowing time for the other nodes' *History* component to adapt. The next two factors measure the demand for n_i's service, both externally, d_i, and internally, q_i. d_i is measured by the number of packets that originated from an external source and are currently in the node's queue. q_i is measured by the number of packets originated from the current node and are still in the queue. In fact, d_i and q_i are related by $d_i + q_i = \text{Queue size}_i$. Finally, another factor that is considered in the external forwarding cost is the amount of credits, c_i, that is available to the node. For clarity, a list of mathematical notations that we have defined is provided in Table I.

TABLE I

LIST OF MATHEMATICAL NOTATIONS

parameter	definition
e^k	Amount of battery required for forwarding of packet k
p^k	Amount of processor required for forwarding of packet k
b^k	Amount of bandwidth required for forwarding of packet k
$x^k_{(i,j)}$	The estimated amount that the next hop, n_j, will charge for forwarding of packet k, based on the previous interactions with n_j
d_i	The demand from other nodes for this node's service, measured by the number of packets in the queue that belongs to other nodes
q_i	The demand from own node for sending, measured by the number of packets in the queue that originates from the same node
c_i	The amount of credits that is available to the node

FAIR uses a two step process to arrive at n_i's *Packet Forwarding Cost* for packet k, Γ_i^k. The calculation of the Γ_i^k is broken into two steps. The first step calculates the Packet Extrinsic Cost, Ω_i, for any packet that are forwarded by n_i. Since the parameters in Ω_i have potentially large fluctuations, we use an exponential weighted average of these parameters to perform a low pass filter to smooth out the fluctuations. The general formula is $\bar{Z}_n = \sum_{i=1}^{n}(a_i \bar{Z}_{i-1}) + \chi Z_n$, where $\sum_{i=1}^{n} a_i + \chi = 1$. The calculations of Ω_i is presented as follows.

$$\Delta\Omega_i = \delta \bar{d}_i + \omega \frac{\bar{c}_i}{\bar{q}_i} \tag{1}$$

$$\Omega_{i_l} = \frac{\Omega_{i_{l-1}} + \Delta\Omega_i}{2} \tag{2}$$

Eq. (1) uses the exponential weighted average of the parameters to calculate $\Delta\Omega_i$. The term $\delta \bar{d}_i$ raises the price as the external demand increases and aids in both throughput and collaboration. Throughput is enhanced by diverting traffic to less congested nodes; collaboration is improved by making other nodes more attractive for forwarding and giving them an opportunity to earn more virtual credits. The term $\omega \cdot \bar{c}_i / \bar{q}_i$ represents the amount of credits per node's packets. If this ratio is high, the node can raise its prices to divert traffic to conserve its own resources and distribute the forwarding effort.

Eq. (2) uses a recursive formula with $\Delta\Omega_i$ to converge to the final Ω_i value. Ω_{i_l} represents the packet extrinsic cost of n_i at period l. This method of calculation allows Ω_i to remember history (*i.e.*, $\Omega_{i_{l-1}}$ for $i = 1 \ldots l$) and yet continuously reflect the external condition changes. The second stage of the pricing algorithm takes the final Ω_i and further includes the cost for the internal resources and the charges from next hop to arrive at the Packet Forwarding Cost, Γ_i^k. Γ_i^k represents the amount of virtual credits charged by n_i for the forwarding effort of packet k and is defined in Eq. (3).

$$\Gamma_i^k = \theta_i[\alpha_i b^k + \beta_i e^k + \gamma_i p^k + \rho_i \Omega_i)] + \varphi_i \bar{x}^k_{(i,j)} \tag{3}$$

The $\varphi_i \bar{x}^k_{(i,j)}$ term represents the estimated cost for forwarding packet k to the next node n_j. The φ factor is introduced to the *Feedback* component to tune the importance of this parameter. Note that to prevent n_i from losing credits from its forwarding effort, FAIR enforces $\varphi \geq 1$. $\rho_i \Omega_i$ reflects the external system conditions. The $\alpha_i b^k + \beta_i e^k + \gamma_i p^k$ terms take energy, processor and bandwidth usage into account. θ is used to tune the overall pricing and will be used extensively in the *Feedback* component to help ensure collaborative pricing.

The final Γ_i^k will be passed to the *Payment* component and billed to the sender node of the packet. This payment and billing process occurs on a *per hop* basis for incentive, flexibility and efficiency reasons. Under this scheme, the sender node has the incentive to limit the amount of packet transmission since it will be charged on a per transmission basis. Contrast this with a destination payment scheme, where the sender can indiscriminately send packets and not be penalized. The connectionless aspect also allows each node to dynamically adjust its pricing strategy according to the present load conditions and resource constraints. Finally, network traffic overhead is minimized because the billing packets will only need to travel a single hop.

D. FAIR: feedback mechanism

The FAIR feedback mechanism, another major innovation of this paper, is discussed in this section along with its contribution to the stability, fairness and collaboration of the system. Each node, n_i, bases its feedback on local information and adjust the various weightings in Γ_i^k in Eq. (3) to dynamically adjust its pricing strategy. The key to successful feedback is being able to accurately estimate the local, regional and global states of the system from local information alone. This is very important because the global state is very hard to obtain in a distributed environment and the protocols that try to provide such a global state have very high overheads. FAIR uses feedback to optimize local, regional and global behavior.

The goals of the local feedback are to ensure the stability of the usage pattern and to equalize the usage of the various resources (bandwidth, energy and processor). The first goal, utilization stability, is to use roughly similar amounts of resource per time slot. This has the advantage of allowing a node to predict how much resource will be used in the future and tune its own packet generation activity accordingly. The second goal, usage equalization, is aimed at preventing scenarios where one resource is in deficit while the other resources are in abundance.

There is no direct method of communication between neighboring nodes to adjust the properties of the packets that are sent. To induce cooperation from the other rational nodes in the system, a node must rely on its Γ_i^k pricing strategy. The feedback for utilization stability is applied similarly to all three resources and will be illustrated with the energy utilization. To encourage a node to use roughly the same amount of energy each period, the feedback modifies the β_i by adding the difference in percentage of the energy used between two periods (l and $l-1$), as given by $\beta_{i_l} = \beta_{i_{l-1}} + (E_{i_l} - E_{i_{l-1}})$. This acts as a negative feedback mechanism by discouraging other nodes from sending lengthy packets that consume significant amounts of energy.

In addition to maintaining utilization stability, usage equalization is critical to renewable resources such as bandwidth and computational cycles, which replenishes after every discrete time interval. Bandwidth and processing power are frequently a more important constraint in the short term. For instance, Queue Size$_i$ in n_i is more influenced by the bandwidth, B_i, and processing power, P_i, than the energy E_i. Hence, we need to ensure that the utilization rate of renewable resources are roughly equal and none of them become a limiting factor. The two weightings α_i, χ_i are adjusted proportionally to the amount of their respective usage. The α_i adjustment is shown in Eq. (4).

$$\alpha_{i_l} = (\alpha_{i_{l-1}} + (B_{i_l} - B_{i_{l-1}})) \times (\frac{B_{i_l}}{B_{i_l} + P_{i_l}} + \frac{1}{2}) \quad (4)$$

The regional feedback tunes ρ_i, the weighting on the Packet Extrinsic Cost, Ω_i, to adjust n_i's emphasis on the external factors. Since the location of the nodes are frequently beyond its control, it would be unfair to penalize or reward the nodes on the basis of location. For example, if a node is the only junction between two clusters of nodes, it will have a monopoly on the traffic across clusters and it can price its forwarding efforts for outrageous amounts. We argue that it is in the interest of the geographically advantageous nodes to also adapt this feedback because of rational reasons. Since FAIR uses a hop by hop routing strategy, a node only pays and controls its packets until the next hop node. Once the packet arrives in another node's queue, the source node has no more control over the packet's activity. If the next node has sufficient virtual credits, the packet will be relayed; otherwise, the packet may be in the next hop's sending queue for an extended period of time. Hence, it is not in a node's interests to be selfish and siphon all of the virtual credits as it will cause its neighboring nodes to be extremely poor, creating an "Island Effect" of a cluster of poor nodes surrounding a rich one. In this situation, the packets from the rich node will become stagnant in the ring of poverty and unable to reach their ultimate destination. Therefore, it is in the node's best interest to "share the wealth" (within reason) to allow other nodes to forward the source node's packets to the destination. The regional feedback achieves this goal of sharing the wealth within a region.

To this end, we identify nodes that are rich or poor based on geographical reasons and seek to remedy it by changing the weighting factor ρ_i. We still use only local information to deduce and induce regional behavior. A finite state machine is introduced to change between the various operational states of the feedback.

Before proceeding to the finite state machine, the feedback mechanism first determines the system state, which is related to the percentage of time, S_i, that n_i spends in the virtual credit accumulation phase or spending phase. S_i can be obtained by examining the change in the n_i's "wallet", w_i, which holds its virtual credits. By measuring the percentage of time that w_i is increasing and decreasing over the period T, FAIR can easily deduce the state of the node. The calculation is shown in Eq. (5).

$$S_i = \frac{1}{T} \int_t^{t+T} \Phi \left(\frac{\mathrm{d}w_i(t)}{\mathrm{d}t} \right) \mathrm{d}t \quad (5)$$

where

$$\Phi(f(t)) = \begin{cases} f(t)/|f(t)| & \text{if } f(t) \neq 0 \\ 0 & \text{if } f(t) = 0 \end{cases} . \quad (6)$$

The two extremes of S_i are 1 and -1, where $S_i = 1$ represents a pure virtual credit accumulation phase and $S_i = -1$ represents a pure spending phase. The finite state machine uses S_i of the current period to transition between its 5 states based shown as follows. The transitions are shown in Fig. 2.

- Normal — n_i is not appreciably gaining or losing wealth
- Geographically rich (GrN$_i$) — c_i is generally increasing due to n_i's geographically advantageous position
- Opportunistically rich (OrN$_i$) — c_i is generally increasing due to n_i's pricing strategy
- Geographically poor (GpN$_i$) — c_i is generally decreasing due to n_i's geographically disadvantageous position
- Opportunistically poor (OpN$_i$) — c_i is generally decreasing due to n_i's pricing strategy

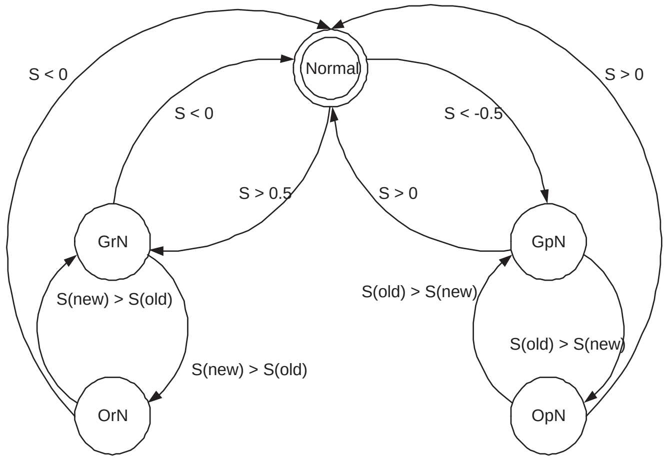


Fig. 2. FAIR Transition

For each node n_i, the state transitions occur every T discrete periods and hysteresis is used to prevent the node from rapidly switching between states. Note that with only local data, it is initially impossible to distinguish n_i state between GrN and OrN, and likewise for GpN and OpN. The regional feedback mechanism first transitions the normal state to GrN or GpN depending on the value of S_i. After the state machine is in one of the geographical states, we try to tune ρ_i to maximize fairness and use the result to confirm the feedback state. For instance, if n_i is classified as GrN_i, it is because it is thought to have a monopoly over a certain path. Hence, n_i should reduce ρ_i to lower its profit margin to spread the wealth. If this decrease in ρ_i leads to more profit, we can conclude that our initial state assumption is incorrect and the node is actually in the OpN_i state, which can gain more profit by decreasing the price (attracts more nodes to forward through it). The state machine then transitions its state accordingly and increase ρ_i to decrease its profit. The rest of the transitions can be understood similarly.

The final feedback mechanism has the global goal of adjusting the average packet forwarding price, $\frac{1}{N}\sum_{i=1}^{N}\Gamma_i$, to match with the average virtual credit supply, $\frac{1}{N}\sum_{i=1}^{N}c_i$. This is an important goal because over or under pricing of packet forwarding will nullify the effects of FAIR. If the Γ_i^k are generally too high, the throughput of the system will suffer because few nodes have sufficient amount of credit to send packets; if Γ_i^k are generally too low, the pricing strategy is ineffective since the nodes will no longer be bounded by the need to earn credits. As nodes power down as the result of lack of power, certain amount of credits will be "locked" up and inaccessible to the rest of the system. Hence, the average packet forwarding price should decrease to take this into account. On the other hand, as new nodes join and introduce new credits, the average packet forwarding price should also adjust accordingly. Note that this overall feedback mechanism makes the initial amount of credits in each of the nodes, n_i unimportant because the Γ_i^k will automatically adjust to the right pricing level.

To affect the overall pricing, each of the nodes, n_i, must tune their prices according to their knowledge of global state. To determine the global state, each node runs a distributed algorithm that logs the average price in the estimation table for all destinations. It periodically calculates the trend of the average prices and acts to counteract the changes. However, since the final destination does not charge the previous node for the receipt of packets, estimates of the node's immediate neighbors are not available. To compensate for this, we assume a steady state system and the amount that n_i charges its neighbors is similar to the amount the neighbors charge n_i for the forwarding effort. Thus, we can approximate the average of n_i's neighbor's forwarding charges by the average of our own forwarding charges, $\frac{1}{K}\sum_{k=1}^{K}\Gamma_i^k$. We are interested in calculating Q_i, the percentage change in the average packet forwarding pricing, Γ_i^k, in the overall system, as perceived by n_i. The calculation for Q_i, which will be used to adjust the overall pricing parameter θ_i, is presented as follows.

$$Q_i = \frac{\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{N}\sum_{j=1\wedge i\neq j}^{N}(x_{(i,j)} + \sigma_{(i,j)}\frac{1}{K}\sum_{k=1}^{K}\Gamma_i^k)\right)}{\frac{1}{N}\sum_{j=1\wedge i\neq j}^{N}(x_{(i,j)} + \sigma_{(i,j)}\frac{1}{K}\sum_{k=1}^{K}\Gamma_i^k)} \qquad (7)$$

where $x_{(i,j)}$ is the estimate of how much n_j will charge n_i to forward a unit size packet. Since $x_{(i,j)} = 0$ if n_i and n_j are neighbors, $\frac{1}{K}\sum_{k=1}^{K}\Gamma_i^k$ needs to be added in those cases. Hence $\sigma_{(i,j)} = 1$ if n_i and n_j are neighbors and 0 otherwise.

To react to changes in the system, the overall pricing parameter, θ_i, is adjusted in the opposite direction as the Q_i. To ensure θ_i does not fluctuate widely, the change is θ_i is capped at both ends. The modified θ_i will then be applied back to the Γ_i^k calculation in Eq. (3).

$$\theta_{i_l} = \begin{cases} 0.75\theta_{i_{l-1}} & \text{if } Q_i \geq 0.25 \\ \theta_{i_{l-1}}(1 - Q_i) & \text{if } Q_i < 0.25 \wedge Q_i > -0.25 \\ 1.25\theta_{i_{l-1}} & \text{if } Q_i \leq -0.25 \end{cases} \qquad (8)$$

To summarize, we present the distributed algorithm implemented in the FAIR middleware architecture in Table II. We emphasize that one of the salient features of FAIR is its extremely low overhead — there will be no extra control messages sent and received between nodes except for the "bill" for packets delivered (and therefore services performed). Especially, "long-haul" signaling messages over multiple wireless hops are completely eliminated. Such a design minimizes the complexity of the protocol, and ensures that FAIR may be efficiently implemented and executed over long periods of time.

IV. SIMULATION RESULTS

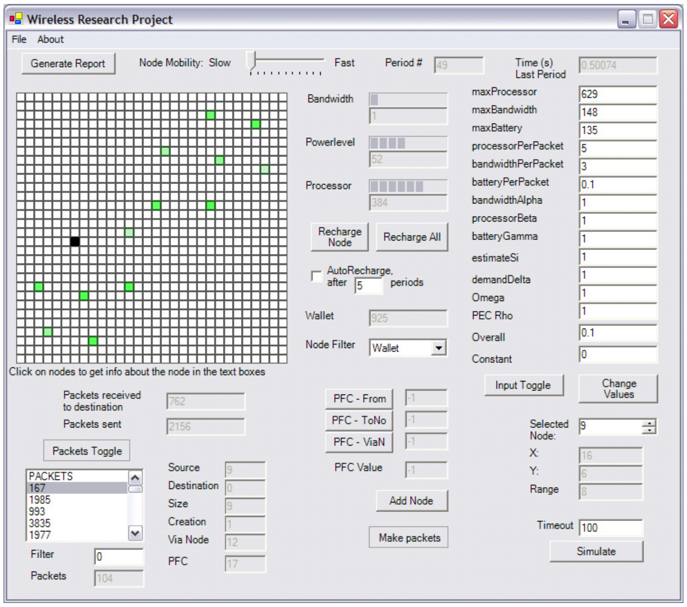
We conducted simulations of FAIR via *PriceSim*, a network simulator shown in Fig. 3, for two stationary configurations of nodes — a 6x6 mesh and 2 clusters connected via a bridge, as well as a mobile configuration consisting of nodes in random placements. The system topology used in the simulation is a 30 x 30 grid that wraps around the edges to eliminate corner nodes in the simulation. The grid is an area that wraps around horizontally and vertically. It creates an ideal sphere where nodes can communicate and traverse to the other end of the grid. The elimination of corner nodes allows the modeling of an infinite field in addition to finite topologies. The system

TABLE II
FAIR ALGORITHM WITH FULL FEEDBACK

```
Node $n_i$ with packet $k$ to send:
  forwarder$_{\text{list}}$ ← Obtain possible next hops
  $n_{\text{forwarder}} \leftarrow \text{forwarder}_{\text{list}}[0]$
  $x_{\text{forwarder}} \leftarrow x^k_{(i,\text{forwarder})}$
  **while** forwarder$_{\text{list}}$ has unvisited nodes
    // $x^k_{(i,j)}$ for the unvisited node, $n_j$, return 0
    **if** $x^k_{(i,\text{current})} < x_{\text{forwarder}}$ **then**
      $n_{\text{forwarder}} \leftarrow \text{forwarder}_{\text{list}}[\text{current}]$
      $x_{\text{forwarder}} \leftarrow x^k_{(i,\text{forwarder})}$
  Sends packet $k$ via $n_{\text{forwarder}}$
  // Node continue other operations while waiting for the bill
  Receives bill for packet $k$ from $n_{\text{forwarder}}$
  Logs transaction in *History*
  Deducts amount from wallet $w_i$

Node $n_i$ received packet $k$ to forward:
  Stores $k$ in Queue$_i$
  Calculates $\Gamma^k_i$ and bills $n_{\text{sender}}$
  Logs $\Gamma^k_i$ calculated for global feedback

Node $n_i$ executes feedback every period:
  Calculates resource utilization $B_i$, $P_i$ and $E_i$ for local feedback
  Use Eq. (4) to calculate the new $\alpha_i, \beta_i, \chi_i$
  Every $T$ periods, execute regional feedback:
    Calculates $S_i$ with Eq. (5)
    Compare $S_i$ with $S_{i_{\text{old}}}$ and transition between state
    $S_{i_{\text{old}}} \leftarrow S_i$
    Adjust $\rho_i$ according to state
  Calculate $Q_i$ by Eq. (7) and adjust $\theta_i$ according to Eq. (8)
```



Period # 49
Time (s) Last Period 0.50074
Bandwidth 1
Powerlevel 52
Processor 384
☐ AutoRecharge, after 5 periods
Wallet 925
Node Filter Wallet
maxProcessor 629
maxBandwidth 148
maxBattery 135
processorPerPacket 5
bandwidthPerPacket 3
batteryPerPacket 0.1
bandwidthAlpha 1
processorBeta 1
batteryGamma 1
estimateSi 1
demandDelta 1
Omega 1
PEC Rho 1
Overall 0.1
Constant 0
Click on nodes to get info about the node in the text boxes
Packets received to destination 762
Packets sent 2156
PFC - From -1
PFC - ToNo -1
PFC - ViaN -1
PFC Value -1
Selected Node: 9
X: 16
Y: 6
Range 8
PACKETS
167
1985
993
3835
1977
Filter 0
Packets 104
Source 9
Destination 0
Size 9
Creation 1
Via Node 12
PFC 17
Timeout 100

Fig. 3. PriceSim in action: two clusters connected via a bridge

operates in discrete time. In every time slot, each node, n_i, will attempt to send all its queued packets. The success or the failure of such attempts is determined by several factors: next hop availability, resource constraints (E_i, B_i, P_i) and virtual credit constraint (c_i). Each node is injected with the same number of new packets (with lengths in a Gaussian distribution) to send every period.

To provide a baseline for comparison, we propose a pricing scheme, *Naive*, as control. The Naive algorithm is also a "credits" trading algorithm, but it does not take the internal or external conditions of the node into account during pricing. The Naive's Packet Forwarding Cost, Γ^k_i, is simply given by $\Gamma^k_i = \text{const}$. In contrast, the FAIR algorithm without feedback will first be executed, and Naive's const will be set as the average of all the FAIR's Γ^k_i. This already gives the Naive protocol a slight advantage in the simulation because the determination of the const to use in the pricing is a non-trivial task. Except for the *Pricer* and the obvious absence of *Feedback*, the remainder of the Naive protocol is identical to the FAIR protocol. We compare the performance of the Naive protocol, basic FAIR (without any feedback), FAIR with local feedback, FAIR without local and regional feedback, and FAIR with full feedback.

Two metrics, Composite Credit Accumulation Rate (*CCAR*) and Credit Expenditure Rate (*CER*), are tracked in *PriceSim* to measure the fairness. We have previously defined fairness as proportionality, accountability and autonomy. Proportionality and accountability seeks to ensure the each node's ability to send packets is directly related to the amount of contribution that it makes to the overall system. To this end, we use CCAR$_i$, which measures the amount of resource that n_i expends to earn virtual credits. Since resource utilization scales linearly to the packet size, a small standard deviation of *CCAR*, $\sigma(\text{CCAR}_i)$, suggests both good proportionality and accountability, because each node needs to exert roughly the same amount of effort per packet size to earn the virtual credits. On the flip side, after n_i earns the virtual credits, each node should also spend roughly the same amount of virtual credits to send packets. The amount of resource spent per transmission is given by CER$_i$, which measures the amount of credit spent per unit length of packet forwarded. The concept of proportionality and accountability can also be measured by the standard deviation of CER$_i$, of which a lower value of $\sigma(\text{CER}_i)$ signifies superior fairness. The concept of autonomy is already encoded into the FAIR *Pricer*'s Γ^k_i by the term $\bar{c}_i/\bar{q}_i$, which modifies the willingness of n_i to forward other node's packets by the relative abundance of credits versus internal demand. We formally define *CCAR* and *CER* as follows.

$$\text{CCAR}_i = \sum_i \frac{\text{credit earned}_i}{\sum_j \text{resource expended}^j_i} \tag{9}$$

$$\text{CER}_i = \frac{\text{credit spent}_i}{\text{data transmitted}_i} \tag{10}$$

To measure collaboration, *PriceSim* tracks the Insufficient Credit Ratio, ICR$_i$, which measures the percentage of packets in n_i that are not transmitted due to credit constraint. As defined previously, collaboration is defined by necessity and sufficiency. Nodes with adequate credit will not be credit constrained, and will exhibit a low ICR$_i$; on the other hand, nodes need to refrain from becoming a credit "sink" to allow other nodes to have adequate credits for their needs. Hence, a low average of *ICR* for the entire network indicates that both goals of collaboration has been achieved.

A. 6x6 *Mesh Configuration*

This configuration is comprised of 36 nodes, arranged in a square lattice. Each node, n_i, can communicate with four of its neighbors (North, South, East and West). Since the simulation topology wraps around, there are no corner nodes. Fig. 4, 5, and 6 shows $\sigma(\text{CCAR}_i)$, $\sigma(\text{CER}_i)$ and average ICR respectively. It may be observed that any version of FAIR outperforms the Naive approach. Note that the artificial depression in $\sigma(\text{CCAR}_i)$ and $\sigma(\text{CER}_i)$ of Naive protocol is again due to nodes being unable to send packets, as indicated by average ICR ≈ 1, which is due to the extremely uneven distribution of credits. The performance of the various versions of FAIR are very similar because of the symmetric nature of the topology. This and the previous configuration illustrates that feedback functions best under heterogeneous topologies, where the regional and global feedback mechanisms have a chance at optimizations. It is remarkable that even under these unfavorable circumstances, all versions of feedback is still able to offer satisfactory performance. This observation is encouraging for the general deployment of feedback because it suggests that feedback can be applied generally in any topology without a negative impact on performance.

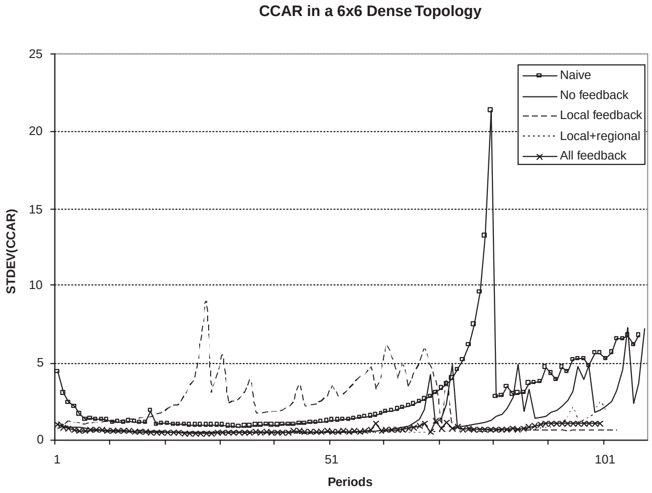


Fig. 4. $\sigma(\text{CCAR}_i)$ for 6x6 Mesh Configuration

B. *2 Cluster Configuration*

This configuration is comprised of 13 nodes arranged in a dumbell configuration. For nodes to communicate between the two clusters, packets must be relayed through the center node, which has been initialized with a higher energy capacity to prevent it from running out during the simulation period. Fig. 7, 8, and 9 shows $\sigma(\text{CCAR}_i)$, $\sigma(\text{CER}_i)$ and average ICR respectively. Like the previous configurations, the Naive protocol became a severe limiting factor by period 50, with ICR ≈ 1. All versions of FAIR again outperforms the Naive protocol by large margins. FAIR with all feedback is able to consistently outperform the other variants for FAIR across all three metrics. For instance, although the local-only feedback is able to achieve at lower average ICR by period 50, it has a much larger fluctuation when compared to all feedback. Moreover, the local-only feedback has a much higher $\sigma(\text{CCAR}_i)$ and $\sigma(\text{CER}_i)$. The results have matched with our hypothesis that the feedback mechanism functions best in a heterogeneous environment with many alternate paths, which is the case for most real world ad hoc wireless network topologies.

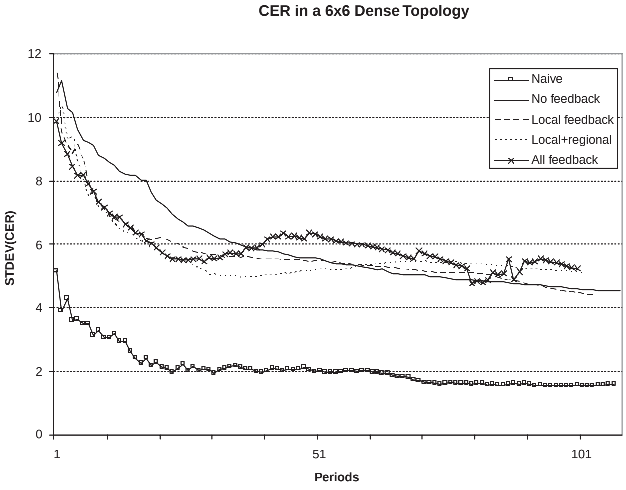


Fig. 5. $\sigma(\text{CER}_i)$ for 6x6 Mesh Configuration

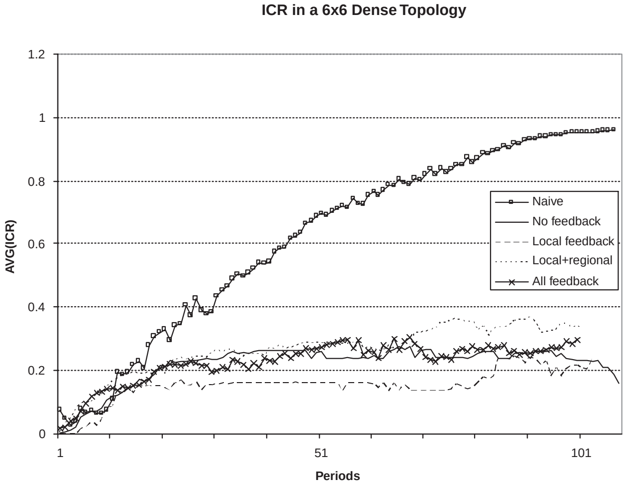


Fig. 6. AVG(ICR) for 6x6 Mesh Configuration

C. *Mobile Configuration*

This configuration is comprised of 13 nodes, associated with the random waypoint mobility model, randomly distributed in the network. Fig. 10, 11, and 12 shows $\sigma(\text{CCAR}_i)$, $\sigma(\text{CER}_i)$ and average ICR respectively. The various versions of FAIR is again superior to the Naive protocol. Since the nodes are in motion and their neighbors keep changing, the nodes must continuously adapt to the new environment. The feedback mechanism is of particular benefit in mobility since it can adapt the *Pricer* to the region and simultaneously maintaining the stability of the average price, thus leading to a high level of fairness (as shown by the low $\sigma(\text{CCAR}_i)$ and $\sigma(\text{CER}_i)$) as well as the high level of collaboration (as demonstrated by the low average ICR). Since this configuration most closely matches with real world deployments of ad hoc wireless networks, this excellent result suggests that FAIR with the

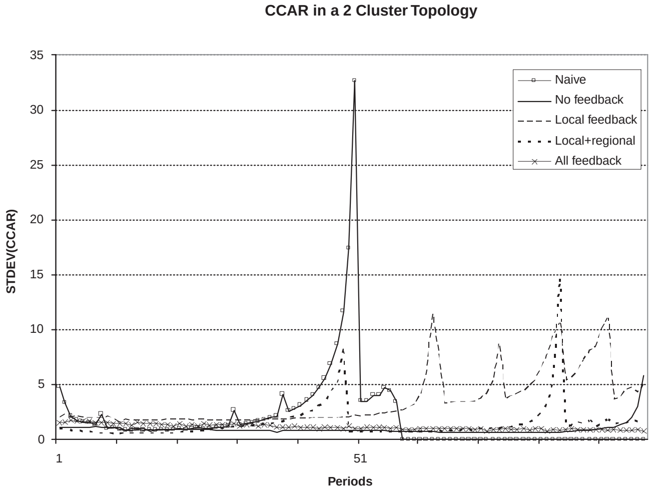


Fig. 7. $\sigma(\text{CCAR}_i)$ for 2 clusters

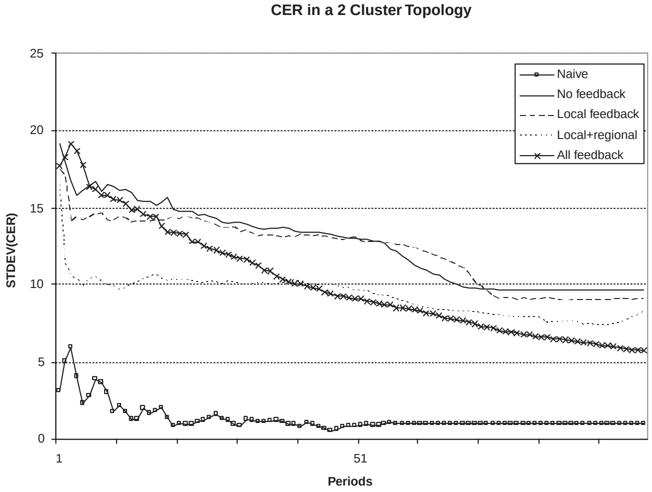


Fig. 8. $\sigma(\text{CER}_i)$ for 2 clusters

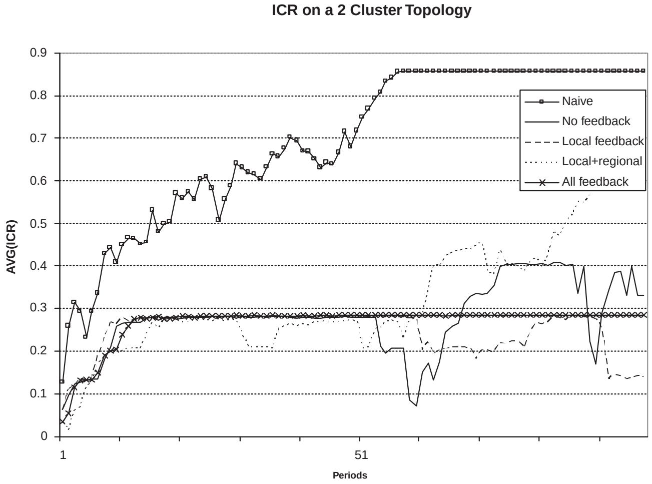


Fig. 9. AVG(ICR) for 2 clusters

feedback mechanism will have superior performance in real world applications.

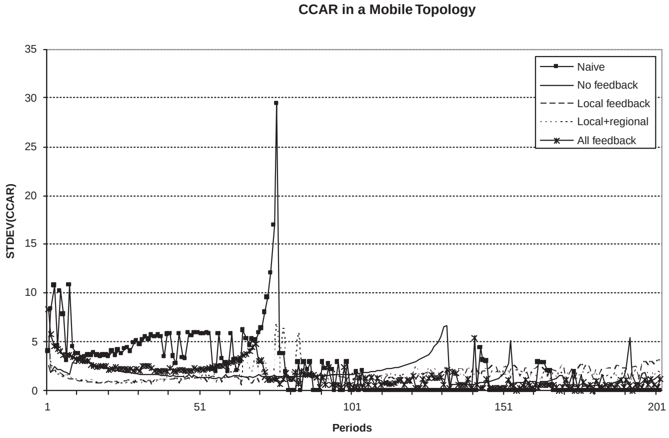


Fig. 10. $\sigma(\text{CCAR}_i)$ for mobile configuration

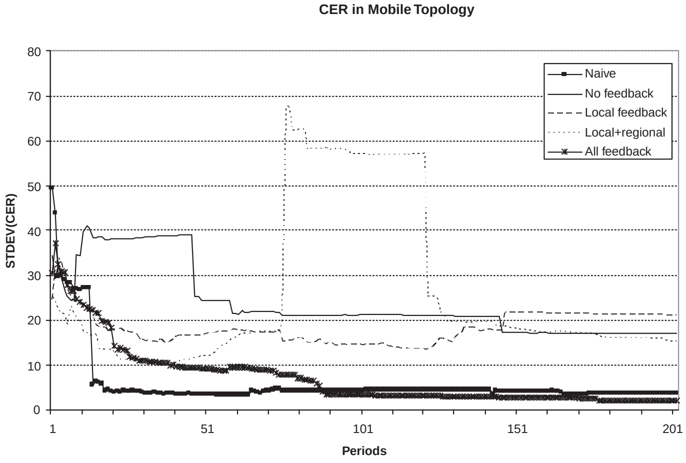


Fig. 11. $\sigma(\text{CER}_i)$ for mobile configuration

V. RELATED WORK

Most of the existing work in the research area of ad hoc network pricing focus on one of the following objectives: *security* and *incentives* towards collaborations. We briefly discuss the related works in the context of FAIR.

Towards the *security* directions, security and tamper resistant software/hardware support are a related area of research in pricing. To ensure that the nodes do not modify their behavior in order to maximize their utility (e.g., by creating fake virtual credits), combinations of hardware [2] and software [1], [3] mechanisms or protocols have been suggested to safeguard the incentive-based pricing and exchange system. Granted, our approach in this paper assumes that security-related mechanisms have been implemented and in full effect in the lower layers. This assumption is valid since we focus on application-level algorithms to make decisions on pricing. The previous work on the security aspect is inherently complementary to our proposal in this paper.

Towards *incentives for collaborations*, the role of pricing is to provide adequate user incentives to forward packets for

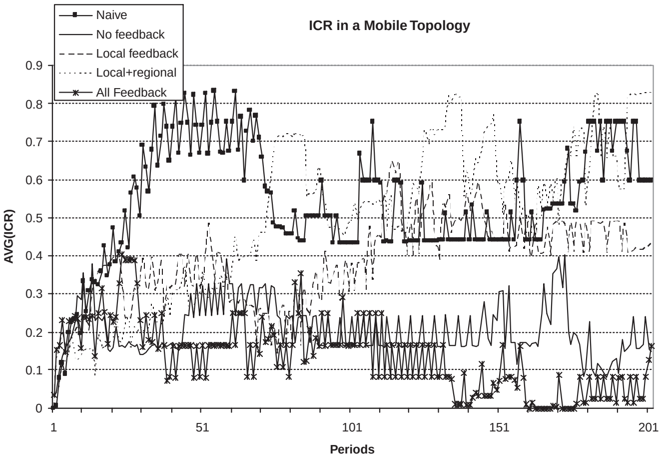


Fig. 12. AVG(ICR) for mobile configuration

other users. For one example, in Qiu *et al.* [4], The goal of optimal price setting at each node is to maximize its net benefit, which reflects its utility gain, its revenue from packet relays and its cost for paying other nodes to relay its own packets. It is shown that the bandwidth allocation driven by the relay-incentive pricing can achieve the global optimum, where the total utility of all users can be maximized, based on a simplified network model. It is apparent that the objective of this work is different from FAIR: it seeks to use pricing mechanisms as a signal to achieve globally optimal bandwidth allocations. In contrast, the focus and objectives of FAIR are *fairness* and degrees of *collaborations* from the users' point of view, rather than globally optimal resource allocation strategies, which take the point of view of the global system. By focusing on the proportionality, accountability and autonomy aspects of fairness, we elevate the discussions of *fairness* to the application level, and treat resource availability as observable parameters when pricing decisions are made.

For a second example, Srinivasan *et al.* [5] has proposed a game theoretic framework to model the rational (but non-cooperative) behavior of nodes in wireless ad hoc networks, and to analyze the optimal trade-off between throughput and lifetime of energy-constrained nodes. A distributed algorithm has been proposed to propel each node, so that subject to its power lifetime to relay packets for the other nodes, the optimal throughput can be achieved. There are two significant differences between the objectives and approaches of FAIR and [5]. First, the design objectives are fundamentally different. FAIR uses virtual credit pricing to promote fairness and collaboration, while [5] systematically analyzes the trade-off between lifetime and throughput. Second, FAIR remains a fully distributed algorithm with only localized algorithms and no additional overhead of exchanging per-node information with other nodes (including neighbors). [5] assumes a static game with complete information of all other nodes, requiring information exchanges. Other game theoretical works [6], [7] follow similar paths as well and are significantly different from FAIR.

Beyond these two examples, there exist other related work towards similar directions. The work closest to our proposal may be from the Terminodes project (e.g., Buttyan *et al.* [2] and many others). These works are based on the concepts of pricing and exchanging virtual credits or nuggets for service. Within these existing works, we have failed to locate a detailed mechanism of making pricing-related decisions, and to answer the critical question of "how much I should charge (pay)?" from a user's point of view. We adopt the virtual credit based pricing mechanism as an underlying foundation of FAIR, but our contributions focus on detailed algorithms that calculate prices using purely localized data, as well as the systematic simulation-based evaluation with respect to the effectiveness of FAIR to achieve its goals. These practical and application-level issues have not been previously addressed in any game theoretical or resource-centric work.

VI. CONCLUDING REMARKS

This paper proposes the Fee Arbitrated Incentive Architecture (FAIR) to enhance the fairness and collaboration in an ad hoc wireless network. In particular, various feedback proposals to dynamically tune the performance of FAIR is studied for different configurations to yield insights into the general applicability of FAIR and feedback. FAIR, even without feedback, is already significantly more advanced than other simple credit-based protocols (modeled by the naive protocol in this work) outlined in Sec. V. As seen in all of the performance graphs, any version of FAIR is superior in all performance metrics than the naive protocol. We observe that FAIR performs convincingly and effectively in improving the fairness and collaboration of the global system with strictly localized algorithms, the two goals that this work has focused on. One of the salient features of FAIR is its minimal protocol overhead, since the "quotation" phase during the pricing process is not included, and protocol control messages over multiple hops are completely avoided. Even with the collection of existing work on this very topic, we believe that this is the first work that practically proposes a credit-based pricing architecture that focuses on promoting both fairness and degrees of collaboration in ad hoc wireless networks.

REFERENCES

[1] S. Zhong, J. Chen, and Y. Yang, "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks," in *Proceedings of IEEE INFOCOM*, 2003.

[2] L. Buttyan and JP. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," *ACM Mobile Networks and Applications*, vol. 8, no. 5, October 2003.

[3] S. Marti, T. J. Giuli, K. Lai, , and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, 2000.

[4] Y. Qiu and P. Marbach, "Bandwith Allocation in Ad-Hoc Networks: A Price-Based Approach," in *Proceedings of INFOCOM*, 2003.

[5] V. Srinivasan, P. Nuggehalli, C. Chiasserini, and R. Rao, "Cooperation in Wireless Ad Hoc Networks," in *Proceedings of IEEE INFOCOM*, 2003.

[6] E. Friedman and D. Parkes, "Pricing WiFi at Starbucks: Issues in Online Mechanism Design," in *Proceedings of ACM Conference on Electronic Commerce*, 2003.

[7] L. Buttyan and JP. Hubaux, "Rational Exchange: A Formal Model Based on Game Theory," in *Proceedings of 2nd International Workshop on Electronic Commerce*, 2001.