Non-pipelined Relay Improves Throughput Performance of Wireless Ad-hoc Networks

Aravind Velayutham, Karthikeyan Sundaresan and Raghupathy Sivakumar
School of Electrical and Computer Engineering
Georgia Institute of Technology
{vel,sk,siva}@ece.gatech.edu

***Abstract*— The communication model typically assumed in wireless ad-hoc networks is based on a traditional "pipelined relay" (*PR*) strategy. In *PR*, an end-to-end flow has multiple outstanding packets (or data units) along the path from the source to the destination. In this paper, we argue that due to several unique properties of wireless ad-hoc networks, *PR* can be fundamentally improved upon. We present a new non-pipelined relay (*nPR*) strategy, where end-to-end flows have exactly one outstanding packet (or data unit) along the end-to-end path. We show that *nPR* has the following properties: (i) Under idealized network conditions, it provides performance improvement, in terms of end-to-end throughput capacity and network transport capacity over *PR*, and achieves proportional fairness; and (ii) Under practical network conditions, it further increases the above performance improvements, both in terms of the throughput achieved, and in terms of the fairness between flows. Finally, we present a forwarding protocol that practically realizes *nPR*. Through analysis and ns2 based packet level simulations, we evaluate the performance of the proposed strategy, and that of the forwarding protocol.**

I. Introduction

Multi-hop wireless ad-hoc networks are severely capacity constrained both due to the inherent limitations in the wireless bandwidth, and because they are highly interference limited. In [1], a model is presented for the evaluation of static wireless ad-hoc networks, where every node acts as a source for one flow, destination for one or more flows, and possibly as a relay for other flows. Using the model, it is shown that the network transport capacity goes up approximately as $O(\sqrt{n})$, as n - the number of network nodes - grows to be large. Consequently, the per-node end-to-end throughput capacity goes down approximately as $O(\frac{1}{\sqrt{n}})$.

Over the last few years, several approaches have been examined to transcend the capacity limits established in [1]. In [2], an approach is presented to leverage multi-user diversity, in the presence of node mobility, to achieve $O(n)$ network transport capacity. However, the approach is appropriate only for delay insensitive applications, where the "insensitivity" requirements can be from a few seconds to a few hours [2]. In [3], [4], the results in [1] are extended for wireless ad-hoc networks with smart antennas. The conclusions are that only an $\Theta(log^2(n))$ improvement in throughput can be achieved even if arbitrarily complex signal processing in terms of beam forming capabilities is assumed. Finally, in [5], the results in [1] and [2] are extended through the derivation of delay bounds, and identification of optimal delay-throughput trade-offs.

In this paper, we revisit the network model considered in [1]. However, we show that, through a simple change in the network communication strategy, fundamental improvements in performance can be achieved for the end-to-end throughput capacity and network transport capacity, in an idealized network setting (with optimal centrally coordinated medium-access control, routing, and relaying). Perhaps, equally importantly, we show that the change also delivers significant additional performance benefits in a practical wireless ad-hoc environment using distributed protocols for medium-access control, routing and packet relaying. Essentially, we term the communication strategy modeled in [1] as *pipelined relay* (*PR*) strategy, where several packets belonging to an end-to-end flow simultaneously wait to be served at different stages along the flow's path. This strategy is the default assumed in most, if not all, related works on both the theory and practice of wireless ad-hoc networks.

In this context, we examine an alternate *non-pipelined relay* (*nPR*) communication strategy, where every end-to-end flow in the network, by default, has exactly one outstanding packet along its path. We establish that this simple change in the communication strategy can result in an $O(log(n))$ performance improvement in the end-to-end throughput capacity in an idealized network setting. In addition, under the same idealized setting, the strategy results in proportionally fair allocation of network resources, and improves the network transport capacity. Note that the latter improvement does not necessarily follow from the earlier mentioned increase in end-to-end transport capacity, and is an additive improvement. Equally importantly, we also demonstrate that *nPR* reduces the degree of contention in the network thereby considerably improving both the utilization and fairness properties of practical distributed medium access control (MAC) schemes such as CSMA and its variants. We also show how *nPR* ameliorates the impact of route failures on flows by virtue of the shorter time flows need to be active given the improved throughput capacity, and provides temporal decoupling between network flows that enables effective load balanced routing to be performed, unlike in *PR* where load balanced routing has been shown to be ineffective due to the high degree of coupling between flows [6].

Finally, we present a distributed forwarding protocol (DFP) that addresses the unique challenges required to realize the *nPR* strategy practically in a wireless ad-hoc network. Thus, the contributions of this paper are threefold:

- In an idealized network setting, we show that *nPR* can achieve fundamental improvements in end-to-end throughput capacity, and network transport capacity, while achieving proportional fairness.
- We show that *nPR* brings in additional performance benefits in a practical wireless ad-hoc network environment using distributed protocols for medium access control.
- Finally, we present the DFP distributed forwarding protocol for wireless ad-hoc networks that realizes the *nPR* strategy, and tackles the unique challenges that arise in the process.

The rest of the paper is organized as follows: In Section II, we present terminology and definitions for the models considered in the paper. In Section III, we describe the performance improvements achievable when using *nPR* and we also present quantitative results to demonstrate the improvements due to *nPR* under practical conditions. In Section IV, we present the distributed forwarding protocol that realizes *nPR*. In Section V, we study the performance of DFP. We present discussions on *nPR*'s impact on other networking concepts in Section VI. Finally, in Section VII we present related work, and conclude in Section VIII.

II. Definitions and Models

In this section, we outline the different paradigms of communication that are of interest in this work. However, to make the discussions convenient we first present certain basic definitions that shall be used in the rest of the paper.

A. Definitions

Network: A wireless ad-hoc network consisting of n nodes randomly and uniformly distributed. Each node in the network serves as a source node choosing a destination at random, resulting in n multi-hop flows unless stated otherwise.

End-to-end flow: A multi-hop flow f_i from source node s_i to destination node d_i with hop length l_i. Let the intermediate nodes of the end-to-end flow be $n_i^1, n_i^2, \ldots, n_i^{l_i-1}$. The end-to-end flow can thus be represented by the following sequence of nodes $\{s_i, n_i^1, n_i^2, \ldots, n_i^{l_i-1}, d_i\}$. We use the term "flow" to represent an end-to-end flow in the rest of this paper unless stated otherwise.

Mini-flow: Every flow f_i can be represented by a sequence of single hop flows ($\{s_i, n_i^1\}$, $\{n_i^1, n_i^2\}$, …, $\{n_i^{l_i-1}, d_i\}$) each of which is referred to as a *mini-flow*. This can also be represented as (h_i^1, h_i^2, …,$h_i^{l_i}$), where h_i^j represents the mini-flow $\{n_i^{j-1}, n_i^j\}$.

Time slot: Basic unit of transmission granularity in a time-slotted communication system.

Contention region: With respect to every link, one can define a region around the link where there can potentially be only one transmission in any given time slot.

B. Pipelined Relay (PR)

In ad-hoc networks, a source node s_i keeps transmitting packets into the network attempting to keep the *pipe* between the destination d_i and itself fully utilized by pumping packets into it at a rate sustainable by the path. We refer to this commonly adopted communication paradigm in ad-hoc networks as the *pipelined relay* strategy. Thus, every flow f_i contributes l_i contending mini-flows in the network.

Formally, *PR* can be captured as follows. Let n be the number of flows in the network and $S(t)$ represent the set of contending mini-flows in the network in any time slot, t. Thus, in steady state we have,

$$\begin{aligned} PR: & \quad \forall\, transmission\ slot\ t, \\ S(t) &= \{\{h_1^1, h_1^2, \ldots, h_1^{l_1}\}, \ldots, \\ & \quad \{h_i^1, h_i^2, \ldots, h_i^{l_i}\}, \ldots, \{h_n^1, h_n^2, \ldots, h_n^{l_n}\}\} \end{aligned}$$

Hence, in *PR* all the constituent mini-flows of every flow contend for channel access in any slot, resulting in a total of $\sum_{i=1}^{n} l_i$ contending mini-flows in the network.

C. Non-pipelined Relay (nPR)

In *nPR*, the source s_i does not perform pipelining of data packets to the destination d_i. Hence, the name *non-pipelined relay (nPR)* strategy. Instead, the source keeps track of the status of the last transmitted packet into the network. Only when the preceding packet p_{i-1} has reached the destination, does the source transmit the next packet p_i into the network. The source thus, does not attempt to keep the *pipe* between the destination and itself full. This in turn, ensures that a flow has only one outstanding packet in the network at any instant. In reality, the data unit can consist of multiple packets as long as the packets are relayed in a bunch. The number of packets in a data unit will have no impact on *nPR*'s performance (see Section VI). In the rest of the paper, we assume a data unit size of one packet for all discussions. Hence, in any time slot only one of the mini-flows belonging to the same flow will have a packet of the flow to contend for channel access. This reduces the number of contending mini-flows contributed by each flow to *one*.

Formally, *nPR* can be modeled as follows. Let n be the number of flows in the network and $S(t)$ represent the set of contending mini-flows in the network in any time slot t. Thus, in steady state we have,

$$\begin{aligned} nPR: & \quad \forall\, transmission\ slot\ t, \\ S(t) &= \{\{h_1^1 \| h_1^2 \| \ldots \| h_1^{l_1}\}, \ldots, \\ & \quad \{h_i^1 \| h_i^2 \| \ldots \| h_i^{l_i}\}, \ldots, \{h_n^1 \| h_n^2 \| \ldots \| h_n^{l_n}\}\} \end{aligned}$$

where the $\|$ operator indicates the presence of only one of the elements in the set. Hence, in *nPR*, only one of the constituent mini-flows of every flow contends for channel access in any slot, resulting in a total of n contending mini-flows in the network.

III. Theoretical Analysis

While intuitively it might appear that *nPR* should perform worse than *PR* due to its non-aggressive strategy, the goal of this section is to provide fundamental reasons as to why this is not so. In this direction, we provide analysis for the performance improvement provided by *nPR* over *PR*. We also study the performance advantages achieved using the *nPR* model in a practical ad-hoc network using distributed protocols.

The assumptions we make for the analysis are largely similar to those made in [1]. Specifically, we consider n nodes randomly and uniformly distributed on the surface of a three dimensional sphere of unit surface area. Every node serves as a source contributing n flows to the network. The destinations are randomly chosen and the sources are all assumed to be backlogged. Every node uses constant power to keep the transmission range constant and the value of the transmission range r is chosen to keep the network minimally connected ($r = \sqrt{\frac{\log n}{\pi n}}$).

We also supplement the theoretical analysis with practical packet-level simulation results obtained using the *ns-2* simulator. We vary the number of nodes from 100 to 500 in steps of 50, with the nodes distributed randomly in a square network grid of size that keeps the network density a constant. All nodes are sources, and destinations are randomly picked. The rest of the details about the simulation environment can be found in Section V. We use centralized flow scheduling [7] as the medium access control protocol for the simulations concerning throughput and network capacity analysis. The centralized MAC protocol performs per-flow scheduling as opposed to per-node scheduling adopted in CSMA/CA.

The notations used in the analysis are provided below:

- W is the capacity of any contention region, i.e. the number of transmitted bits that can be supported by the contention region in one second.
- M is the total number of contention regions in the network. Note that the different contention regions in the network can be obtained by identifying all the maximal cliques in the flow contention graph [8].
- C_π is the average contention level using the π model of communication. The contention level, C_π^ϕ, of a contention region, ϕ, can be expressed as the number of mini-flows sharing the capacity of the contention region, ϕ, in steady state.
- $\lambda(\pi)$ is the average throughput per-flow using the π model of communication.
- l_i is the hop-length of the flow f_i, and is assumed to be uniformly distributed, unless explicitly stated otherwise.
- $max_l = max(l_1, l_2, \ldots, l_n)$ where l_i is the hop-length of flow f_i.
- l_{av} is the *average hop-length* of the n flows in the network.

A. Throughput capacity (TC)

Throughput capacity (TC) is defined as the sum of the throughputs of all the flows in the network.

Proposition 1 : $TC(nPR) = TC(PR) \cdot O(\log(max_l))$.

Proof:

By the definition of the *PR* model, at steady state, the number of contending mini-flows contributed by any flow f_i is equal to the hop-length l_i of the flow. Hence, at steady state, the total number of mini-flows in the network, using *PR*, is given by $n \cdot l_{av}$. Let the probability mass function of hop length distribution of the flows in the network be given by $p_h(k)$. Then,

$$l_{av} = \sum_{k=1}^{max_l} k \cdot p_h(k) \tag{1}$$

The average contention level using the *PR* model is given by,

$$\begin{aligned} C_{PR} &= \frac{\text{No. of mini-flows in the network using the } PR \text{ model}}{\text{No. of contention regions in the network}} \\ &= \frac{n \cdot l_{av}}{M} \end{aligned} \tag{2}$$

Hence, we see that the capacity of each contention region, ϕ, is shared by $\frac{n \cdot l_{av}}{M}$ mini-flows on an average.

The throughput of a flow f_i can be obtained by determining the time taken by each bit transmitted by the source to traverse l_i hops in the network. However, in *PR*, the source aims to keep the pipe between the destination and itself full by constantly pumping packets into the pipe. This results in the number of bits in-transit at any instant being approximately equal to the number of hops. Hence the cost (time taken by each bit) has to be amortized over the hop length l_i of the flow. This, in turn, makes the throughput of the flow f_i dependent on the throughput of the mini-flows. Specifically, the throughput of a flow using the *PR* model reduces to the throughput of the bottle-neck mini-flow (mini-flow with the minimum throughput) in the path of the flow. Assuming the average contention level to be the same in all the contention regions, all the mini-flows of the end-to-end flow would obtain the same throughput. Hence, we have the throughput of an end-to-end flow using *PR* to be

$$\lambda(PR) = \frac{W}{C_{PR}} = \frac{W \cdot M}{n \cdot l_{av}}$$

Hence, we have the throughput capacity in *PR* as,

$$TC(PR) = n \cdot \lambda(PR) = \frac{W \cdot M}{l_{av}} \tag{3}$$

Using the *nPR* model, every flow f_i has only one contending mini-flow at any time instant. Hence the total number of contending mini-flows is the same as the number of flows (n) in the network. Hence the average contention level per contention region using the non-pipelined model is

$$\begin{aligned} C_{nPR} &= \frac{\text{No. of mini-flows in the network using the } nPR \text{ model}}{\text{No. of contention regions in the network}} \\ &= \frac{n}{M} \end{aligned}$$

Since in *nPR*, it is ensured that there is only one outstanding data unit belonging to a flow at any instant, the throughput achieved by each flow f_i can be calculated by determining the time taken for a bit to be transmitted from the source s_i to the destination d_i through l_i hops. The time taken for a bit to traverse from the source to the destination is the sum of the time taken for the bit to traverse each hop h_k, $k \, \epsilon \, [1, l_i]$, of flow f_i. Time taken for a bit to traverse through one hop in *nPR* is given as $\frac{C_{nPR}}{W}$. The throughput of a k-hop flow (τ_k) can be expressed as :

$$\begin{aligned}
\tau_k &= \frac{1}{\text{Time taken to traverse } k \text{ hops}} \\
&= \frac{1}{\frac{k \cdot C_{nPR}}{W}} = \frac{W}{k \cdot C_{nPR}} \\
&= \frac{W \cdot M}{k \cdot n} \qquad (4)
\end{aligned}$$

It can be seen from the above equation that *nPR* tends to favor shorter hop flows at the cost of longer hop flows.

Given the hop length distribution of the flows to be $p_h(k)$, the total throughput using *nPR* can now be derived as

$$\begin{aligned}
TC_{nPR} &= n \cdot \sum_{k=1}^{max_l} p_h(k) \tau_k = \sum_{k=1}^{max_l} \frac{n \cdot p_h(k) \cdot W \cdot M}{k \cdot n} \\
&= \sum_{k=1}^{max_l} \frac{p_h(k) \cdot W \cdot M}{k} \qquad (5)
\end{aligned}$$

The ratio of the transport capacities of *PR* and *nPR*, ρ, from equations (3) and (5) can be obtained as,

$$\rho = \frac{TC_{nPR}}{TC_{PR}} = l_{av} \cdot \sum_{k=1}^{max_l} \frac{p_h(k)}{k}$$

When the hop lengths are uniformly distributed, we have,

$$p_h(k) = \frac{1}{max_l}, \quad \forall k \, \epsilon \, [1, max_l]$$

Substituting for $p_h(k)$ in Equation (1), we have

$$l_{av} = \sum_{k=1}^{max_l} \frac{k}{max_l} = \frac{max_l + 1}{2}$$

Further, we have,

$$\sum_{k=1}^{max_l} \frac{p_h(k)}{k} = \sum_{k=1}^{max_l} \frac{1}{max_l \cdot k} \qquad (6)$$

In the asymptotic case of $n \to \infty$, we have

$$\lim_{n \to \infty} \sum_{k=1}^{n} \frac{1}{k} = \log(n)$$

So, in the asymptotic case of $\lim_{max_l \to \infty}$, Equation (6) becomes,

$$\sum_{k=1}^{max_l} \frac{p_h(k)}{k} = \frac{\log(max_l)}{max_l}$$

Hence, in the asymptotic case, this results in an improvement factor of,

$$\begin{aligned}
\rho &= l_{av} \cdot \sum_{k=1}^{max_l} \frac{p_h(k)}{k} \\
&= \frac{max_l + 1}{2} \cdot \frac{\log(max_l)}{max_l} = O(\log(max_l)) \qquad (7)
\end{aligned}$$

for *nPR* over *PR*. The improvement in throughput in *nPR* comes as a result of it favoring the shorter hop flows at the cost of the longer hop flows, and hence increasing the aggregate throughput capacity of the system. Thus, we have shown that the throughput capacity using non-pipelined *nPR* is greater than the throughput capacity achieved using the pipelined *PR*. ■

In order to compare and corroborate the throughput capacity achieved using *PR* and *nPR* in practical scenarios, simulations were performed by varying the number of nodes in the network. The results are presented in Figure 1. It can be observed that the throughput capacity in *nPR* is indeed higher than that of *PR*. Note that due to the use of a centralized scheduler in scheduling transmissions, the simulation results do represent actual throughput capacity and are not affected by the potential inefficiencies of distributed protocol operations.

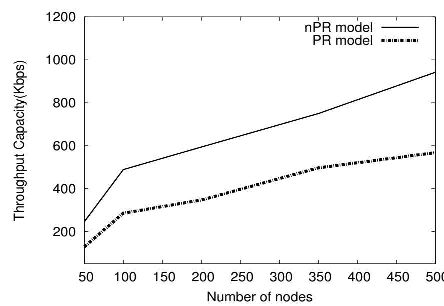


Fig. 1. Throughput capacity comparison between *PR* and *nPR*

In summary, *nPR* allows for only one packet in transit for a flow on its end-to-end path. Despite the strategy, the utilization of network resources does not go down since the number of outstanding packets (n) is still larger than the number of contention regions O($\frac{n}{log(n)}$). In *nPR*, the end-to-end throughput of flows are inversely-proportional to their respective hop-lengths. Hence, shorter hop flows enjoy proportionally more end-to-end throughput than longer hop flows. Thus, the overall end-to-end throughput capacity of *nPR* is larger than that of *PR*.

B. Fairness

We define fairness as the measure of deviation among the throughputs achieved by the flows in the network.

Proposition 2 : *nPR is proportionally fair.*

Proof: *nPR* favors shorter flows to longer flows in providing throughput. We use the proportional fairness analysis presented in [9] to show how *nPR* follows a proportional fairness model.

Consider a utility-based network with a set J of resources, and let C_j be the finite capacity of resources j, for $j \epsilon J$. Let a route r be a non-empty subset of J, and let R be the set of possible routes. Set $A_{jr} = 1$, if $j \epsilon r$, so that resource j lies on route r, and set $A_{jr} = 0$ otherwise. This results in a 0-1 matrix $A = (A_{jr}, j \epsilon J, r \epsilon R)$. Associate with each user, a route r and assume that if a rate x_r is allocated to a user r then the user has the utility $U_r(x_r)$. Assume that the utility $U_R(x_r)$ is an increasing, strictly concave and continuously differentiable function of x_r over the range $x_r \geq 0$. Now, the optimal rates of the system are obtained by solving the following optimization problem,

$$max \sum_{r \epsilon R} U_r(x_r)$$

subject to,

$$Ax \leq C, \; over \; X \geq 0$$

Moreover, it has been shown in [9] that in order to obtain proportionally fair rate allocation, the utility functions must be logarithmic functions of the allocated rate. In addition, if weights are associated with each user w_r, then the network optimization problem now reduces to solving for a weighted proportionally fair rate allocation vector, given by,

$$max \sum_{r \epsilon R} w_r \log(x_r)$$

subject to,

$$Ax \leq C, \; over \; X \geq 0$$

For this system, it can be shown through Lagrangian methods that the unique optimum to the optimization problem is given in [9] to be,

$$x_r = \frac{w_r}{\sum_{j \epsilon r} \mu_j}$$

where μ_j represents the prices (or otherwise called feedback signals) at each resource of the route r. Now, it can be seen that if the prices on each of the resources are the same ($\mu_j = \mu_{av}, \forall j \epsilon r$), then,

$$x_r = \frac{w_r}{l_r \cdot \mu_{av}}$$

where l_r represents the hop length of flow r. Thus, if the converged allocated flow rates are inversely proportional to their hop lengths, then it can be shown that the allocation is proportionally fair. Now, it follows that *nPR* adheres to proportional fairness since in steady state, it ensures that the flows have a throughput given by,

$$x_r = \frac{W}{l_r \cdot C_{nPR}}$$

assuming that the average contention level is the same across all contention regions. When the contention level is not the same, *nPR* provides,

$$x_r = \frac{W}{\sum_{k=1}^{l_r} C_{nPR}^k}$$

where C_{nPR}^k represents the contention level in the contention region with respect to the k^{th} hop of flow r. This in turn is of the form $x_r = \frac{w_r}{\sum_{j \epsilon r} \mu_j}$. Thereby, we show that *nPR* conforms to a proportional fairness model.

■

Since any proportionally fair rate allocation is known to maximize the aggregate utility (proportional fairness index), $\sum_{i=1}^{n} \log(x_i)$, we perform simulations to compare $\sum_{i=1}^{n} \log(x_i)$ using both *nPR* and *PR*. It can be seen from Figure 2 that *nPR* is better than *PR* in attempting to be proportionally fair. Note that this is not to be taken as a critique of *PR*'s fairness properties as *PR* is not meant to provide proportional fairness. *PR* can be shown to provide max-min fairness.

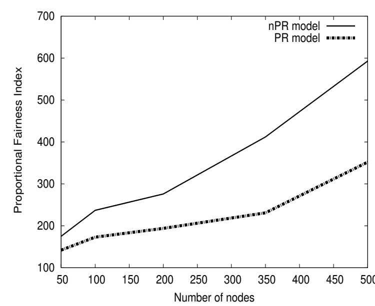


Fig. 2. Proportional fairness comparison between *PR* and *nPR*

While *nPR* is proportionally fair, we also study *nPR*'s fairness properties against that of *PR* using a standard metric such as the normalized standard deviation. The normalized standard deviation (standard deviation normalized by the mean) among the flows using *nPR* and *PR* is presented in Figure 3, where it can be seen that the normalized standard deviation using *nPR* is much smaller than that achieved using *PR*. The deviation in throughput due to the hop length bias is off-set by (i) the increase in the average throughput of the flows, and (ii) decrease in the location dependent max-min unfairness within a contention region by virtue of reducing the number of mini-flows per contention region. The latter reason can be explained further as follows: in max-min fairness, when there is location dependent deviation in the contention, flows traversing larger bottlenecks will receive lower throughput than flows that traverse smaller bottlenecks. However, *nPR* reduces the expected number of mini-flows per contention region by a factor of l_{av}. Hence, the ratio of the maximum-flows-in-contention-region to the minimum-flows-in-contention-region can be expected to go down by a factor of l_{av}, thus improving the fairness among flows.

In summary, in *nPR*, the *raw* network resources are allocated in a hop-length independent fashion to the flows

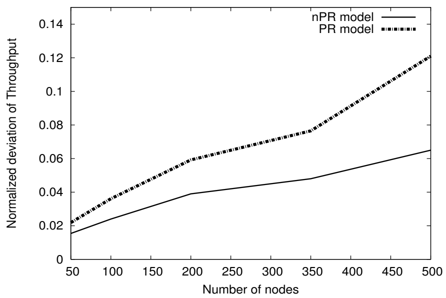


Fig. 3. Normalized deviation in the throughput achieved by the flows using PR and nPR model

in the network. This results in the flows enjoying *end-to-end throughput* inversely-proportional to their respective hop-lengths, and hence achieving proportional fairness.

C. Transport Capacity of the Network

The transport capacity $NC(\pi)$ using a model of communication, π, is defined as the number of bit-meter transported in one second by the network.

Proposition 3 : $NC(nPR) \geq NC(PR)$.

Proof:

Firstly, it can be shown that the probability of any contention region being devoid of flows is very small for both *nPR* and *PR* and hence the probability that a contention region will go completely un-utilized is very small. For the network model assumed in the beginning of this section, given that the transmission range is maintained at minimum connectivity, it can be shown that the number of contention regions in the network is O($\frac{n}{\log(n)}$). In *PR*, we know that the number of mini-flows in the network is $l_{av} \cdot n$, where l_{av} is O($\sqrt{\frac{n}{\log(n)}}$). Hence, the number of mini-flows per contention region in *PR* is O($\sqrt{n \cdot \log(n)}$). In *nPR*, we know that the number of mini-flows in the network is n. Hence, the number of mini-flows per contention region in *nPR* is O($\log(n)$). Thus, in both *PR* and *nPR*, the probability that a contention region will go un-utilized is very small.

Given, that all contention regions are utilized, it can now be shown that the transport capacity for *nPR* is larger than that for *PR*. In *nPR*, since every source maintains only one out-standing packet in the network, as long as a contention region has at-least one mini-flow, the contention region will be fully utilized. However, this is not the case in *PR*. In *PR*, since the source pipelines its data toward the destination, the throughput obtained by the flow is determined by the throughput of the bottle-necked mini-flow. Hence, the other mini-flows belonging to the same flow cannot use the capacity of their respective contention regions to more than that of the bottle-necked throughput. Hence, as long as *all* the mini-flows in a contention region have their corresponding bottle-neck elsewhere along the flow, there would be under-utilization of the capacity of the contention region under consideration and hence reduction in the transport capacity of the network.

The probability that all the mini-flows in a contention region have their corresponding flows' bottle-neck elsewhere along the flow can be characterized as follows. Assume that the distribution of the contention level in a contention region is uniform. Let c_{max} represent the maximum value of contention level in any contention region. Now consider a contention region with contention level c. This means that there are c mini-flows contending in the contention region under consideration. Further, assume the average hop length of each flow to be h. The probability that each of these mini-flows experiences bottle-necks elsewhere along the $h-1$ hops is given by $1 - (\frac{c}{c_{max}})^{h-1}$. Hence, the probability that all the c mini-flows experience bottle-necks elsewhere is given by $\{1 - (\frac{c}{c_{max}})^{h-1}\}^c$. Thus, the probability (p) that a contention region is under-utilized in *PR* is given by,

$$\sum_{c=1}^{c_{max}} \frac{c}{c_{max}} \cdot \{1 - (\frac{c}{c_{max}})^{h-1}\}^c \tag{8}$$

As long as there is a deviation in the contention levels of the contention regions, the transport capacity of *nPR* will be larger than that of *PR*. However, when the contention levels of all the contention regions are the same (deviation = 0), then the transport capacity of *nPR* will be the same as that of *PR*.

■

We provide simulation studies performed using a 500 X 500 m topology with 50 nodes. Every node serves as a source and all the sources are backlogged. The location of the nodes in the topology and the destination of the flows are biased to achieve deviation in the average contention level distribution in the network. Figure 4 presents the results for different deviations in the channel contention levels and how it impacts the transport capacity achieved using *nPR* and *PR*. It can be seen that as the deviation in the contention level increases, there is a drop in the transport capacity achieved using *PR*, while it almost remains unperturbed in *nPR*. This in turn is due to the transport capacity of *nPR* being independent of the deviation in the contention level in the network.

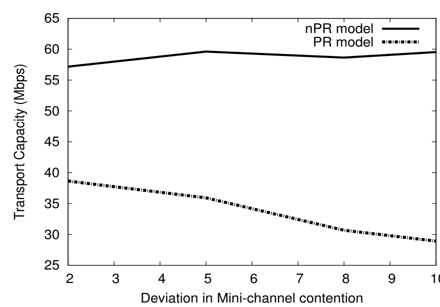


Fig. 4. Comparison of Transport Capacity achieved using the flows using *PR* and *nPR* model

In summary, in *PR*, a flow bottlenecked at one contention region cannot use any additional resources available at the other contention regions it traverses. Hence, there is a finite probability for contention regions being under-utilized when there is a deviation in the distribution of mini-flows over contention regions. However, in *nPR*, the service experienced by a flow is a function of the cumulative service it experiences in each of the contention regions it traverses. Hence, even when a flow passes through a heavily contended region, it can still use the resources in other regions to the fullest extent possible, thus eliminating the chances of under-utilization.

D. MAC protocol utilization

Property: *nPR reduces the number of contending mini-flows per contention region and hence improves utilization.*

nPR ensures that each source has only one out-standing data packet in the network at any time instant. Hence, it reduces the total number of contending mini-flows to the actual number of flows in the network. Recall from our discussions earlier in the section that this is a reduction in the number of mini-flows (and hence contention in the network) by a factor of l_{av} (average hop length) when compared to *PR*.

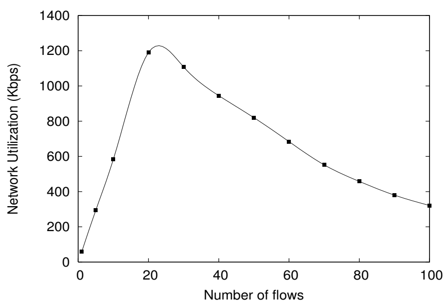


Fig. 5. Channel utilization for IEEE 802.11 CSMA/CA protocol

The utilization curve for the CSMA/CA MAC protocol is a bell-shaped curve, with the utilization of the contention region increasing till the number of contending stations (flows) saturates the channel capacity, and then starts to decrease as the number of flows keeps increasing. This can be observed from Figure 5. In general, this is true for any *contention based MAC protocol* which can be explained as follows. In any random channel access mechanism, the probability of successful transmission in a time slot can be considered as representative of the channel utilization. This successful probability is in turn composed of two components: one that increases the successful transmission probability with increasing load, and the other that decreases it with increasing load. However, the second component starts to dominate once the load saturates the channel capacity and hence the channel utilization starts to decrease. However, the slope of the increasing and decreasing function is dependent on the nature of the channel access mechanism employed.

In ad-hoc networks where the flows are in general multi-hop flows, different flows experience bottle-necks at different regions of the network. Since the throughput of a flow in *PR* is determined by its bottle-necked mini-flow's throughput, the utilization of a contention region can be reduced by virtue of its contending flows being bottle-necked elsewhere, in addition to the reduction contributed by the increasing load. However, *nPR* reduces the number of contending flows and thereby helps operate in the optimal region of the utilization curve for a larger range of higher loads. For example, consider 100 nodes in a 1000m X 1000m grid. Let the transmission range be 250m and the carrier-sense range be 500m. Hence, the total number of contention regions is approximately between 2 and 4. Assuming the number of contention regions to be 4, the average hop length to be 3 and the number of sources to be 100, the number of contending mini-flows per contention region is about 75 in *PR* while it is only about 25 in *nPR*. Hence, referring back to Figure 5, while *PR* operates on the down-slope of the utilization curve, *nPR* operates in the optimal region.

One aspect of *nPR* that might be of serious concern is how to prevent it from operating in the under-utilization region. While the probability of under-utilization is small, occurring only at very low number of flows, nevertheless we address it in the context of the design of the distributed forwarding protocol described in Section IV.

IV. Distributed Protocol

Thus far, we have discussed the merits of *nPR* with respect to *PR*, in terms of throughput capacity, transport capacity, fairness and MAC utilization. In this section, we present the Distributed Forwarding Protocol (DFP) that realizes *nPR* under practical conditions. DFP sits atop the routing layer, and beneath the transport layer in the protocol stack. Every packet forwarded by an intermediate node goes through DFP.

DFP consists of the following three key elements: (i) proactive acknowledgments - to maintain the one-packet-per-flow principle; (ii) proportional rate adaptation - to proportionally use unused spatial-reuse resources, possible when the network load is low; and (iii) load balanced routing - to exploit the increased potential for load balanced routing that *nPR* enables. In the rest of the section, we motivate the rationale behind each of the above elements, and describe the corresponding approach.

A. Proactive Acknowledgments

nPR is based on the strategy of having exactly one packet in transit along the end-to-end path for every flow. This ensures that there are exactly n packets in transit in the network, where n is the number of nodes (and flows) in the network. This results in the expected number of contending flows per contention region to be $O(log(n))$, which is a considerable reduction from the contention levels of *PR*. We need a mechanism that ensures the one-packet-in-transit principle to prevent *nPR* from under-utilizing the network resources.

While straightforward destination-originated ACKs is a potential solution, this would reduce the number of packets in the network by approximately a factor of *two*, since during the time the ACKs travel back to the sources, the sources will remain idle. DFP solves this problem by using the notion of *proactive acknowledgments*.

Essentially, the temporal mid-point of a flow (in terms of delay) is dynamically kept track of, and ACKs are sent back from the mid-point back to the source. This ensures that at the time the data packet is delivered to the destination, the source initiates the next packet transmission. At every intermediate node, the time t_i taken by each node i from the time it received the packet till it manages to transmit the packet is piggybacked on every packet header. The time t_i is appended to the packet header along with the identifier i of the intermediate node.

Then, when the destination receives the data packet, it computes the temporal mid-point as the node m that satisfies the following conditions:

$$\sum_{i=m}^{h} t_i \geq \sum_{i=1}^{m} t_i, \tag{9}$$

$$\sum_{i=m+1}^{h} t_i < \sum_{i=1}^{m+1} t_i \tag{10}$$

The mid-point identifier is then stamped on the ACK and sent back towards the source. The corresponding mid-point node notes itself as the temporal mid-point and lets the ACK propagate back to the source.

When the source sends the second data packet, the mid-point node responds with an ACK. From here-on, the destination originated ACK is intercepted by the next elected mid-point node (as a proactive ACK has been sent by the previous mid-point node). Note that if the network conditions are stable, the temporal mid-point will not change. However, the mid-point selection is performed on a per data-packet basis to keep up with any network dynamics.

Finally, as we discuss in Section IV-B, there might be certain operating conditions under which DFP may have multiple packets in transit for a given flow (say k). In such an event, the proactive ACKs then need to be sent on each *path-segment* (each segment starting from the source consists of $\frac{h}{k}$ hops), along the end-to-end path, and hence temporal mid-points are chosen on a path-segment basis. Since the number of packets in transit is taken on an end-to-end basis, and the corresponding path-segments are identified in a straightforward manner, the node at the end of each segment plays the role of a *pseudo-destination* by acting as the destination described earlier in the mechanism.

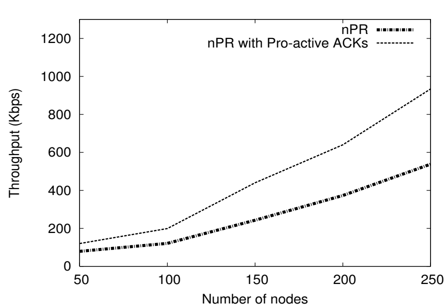


Fig. 6. Proactive acknowledgments advantage

Figure 6 shows the amount of performance improvement provided by the proactive acknowledgment mechanism for a topology with varying nodes but with constant node density, with all of them acting as sources and generating data at 800 Kbps.

B. Proportional Rate Adaptation

The *single data unit in transit* principle in *nPR* is what distinguishes it from *PR*. However, there are network conditions under which such a strategy will perform worse. An extreme example of such a condition is when there exists only one flow in the network. In such a scenario, having only one packet in transit is not the best strategy to employ as it will under-utilize the network resources. In other words, such a strategy provides us with performance benefits under conditions where there are sufficient number of flows in the network to not under-utilize network resources. A coarse lower bound for the number of flows required is $\frac{n}{\log(n)}$ flows, as there are same order of contention regions in the network.

While is is quite reasonable to expect the number of flows in the network to be of the above order (e.g. 16 flows for 100 nodes), it is still desirable for DFP to be able to adapt its point of operation to avoid under-utilizing the network under low-load conditions.

DFP uses a simple marking based feedback strategy by intermediate relay nodes to detect low-utilization conditions. The source reacts by appropriately changing the number of packets in transit for a flow. Note that there are two key requirements that need to be incorporated into such an adaptation scheme: (i) the adaptation should again lead to proportional fairness, given the fairness model established for *nPR*, and (ii) the increase should not lead to self-contention between different packets in transit for the same flow. We now describe the specific mechanisms employed at the DFP source and intermediate nodes to achieve the adaptive determination strategy for the number of packets in transit.

Each intermediate node keeps track of its own channel utilization information by maintaining an exponential average of its queue size q_{avg}. While this is similar to that of mechanisms employed in wired environments such as AQM [11], [12], there is a subtle difference in this specific context for how the queue length is monitored. In an environment such as a wireless ad-hoc network, the real queue size of interest is that of the cumulative queue of all nodes in a contention region [13], as the utilization of the contention region is truly determined by the cumulative value. Assuming that the queue size is averaged every transmission slot, DFP accounts for this variation by considering the queue size to be zero (in its averaging process) *only* when the queue size is empty *and* the corresponding transmission slot is left unused in the contention region (the channel remains idle). In other words, the instantaneous queue size considered by a node during the averaging process is the sum of its queue size and *one* if the transmission slot is used by a node other than itself.

Also, due to the fewer number of packets in transit (in steady state) in the network ($O(n)$), the queue occupancies on a per-node basis will typically be a small number ($O(1)$). Hence, the thresholds used for determining whether the contention region is under-utilized, over-utilized, or otherwise are relatively small values. Thus, using two thresholds α, and β, each intermediate node determines how its contention region is utilized:

$$\begin{aligned} q_{avg} &\leq \quad \alpha \rightarrow underutilization \\ q_{avg} &\geq \quad \beta \rightarrow over-utilization \\ otherwise & \quad \rightarrow optimal\ utilization \end{aligned}$$

Every data packet header has a 2-bit *utilization (U)* field, where the first field $U.1$ corresponds to the *under-utilization*, and hence *increase* feedback, while $U.2$ corresponds to the *over-utilization*, and hence *decrease* feedback. The initial value of the field is set to 10. Each intermediate node, depending upon its inferred region performs the following, where U_{in} is the incoming value for the utilization field, and U_{out} is the value this node changes it to before the packet leaves:

$$\begin{aligned} if\ (underutilized), &\quad U_{out} = U_{in} \\ if\ (overutilized), &\quad U_{out} = 01 \\ if\ (optimal\ utilized), &\quad U_{out}.1 = 0,\ U_{out}.2 = U_{in}.2 \end{aligned}$$

The destination piggybacks the U_{in} it receives on the acknowledgment it sends back to the source. In the event of there being multiple intermediate pseudo-destinations and proactive ACKing temporal midpoints, such information is propagated back on a sequence of ACKs.

The source, upon receiving the feedback, performs a *linear increase*, *maintain*, or *multiplicative decrease* of the number of packets it is allowed to have in transit. Note that the LIMD mechanism is to ensure fair sharing of the network resources [9].

Figure 7 shows the performance improvement achieved by using spatial re-use when the network is operating in the under-utilized region. We can observe that the improvement is achieved under low-load (smaller number of flows) conditions. This improvement is achieved by increasing the number of mini-flows in the network through the proportional rate adaptation mechanism.

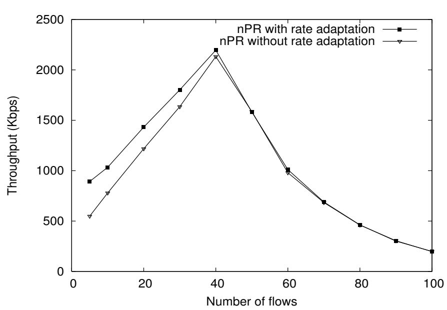


Fig. 7. *nPR* with rate adaptation

C. Load Balanced Routing

nPR can increase the time separation between the schedule of two mini-flows upto h_{av} time slots. We refer to this as the *temporal separation* between the mini-flows. This temporal separation in turn is responsible for the reduction in the total number of mini-flows by a factor of h_{av}. In the case of *PR*, since the number of mini-flows contributed by n flows is large $n \cdot h_{av}$, the probability of finding decoupled routes even at moderate loads is fairly small. Hence, the usage of Load Balanced Routing (LBR) protocols does not bring significant performance improvements at moderate to high loads [6] . On the other hand, in *nPR*, since the number of contending mini-flows is reduced by a factor of h_{av}, the coupling between the routes is correspondingly decreased. Hence, the load at which the probability of finding decoupled routes becomes small is increased. This helps the LBR protocols using *nPR* leverage performance improvements in larger loads than using *PR*. Thus, *nPR* has a better potential for load balanced routing than *PR*.

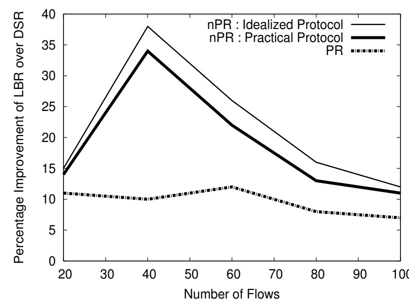


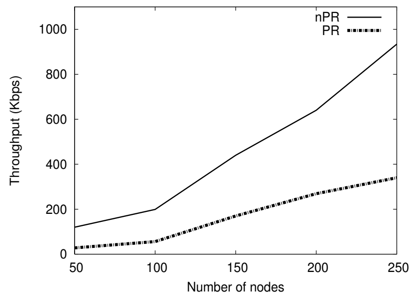
Fig. 8. Potential for load balanced routing

While DFP is not a routing protocol, we include a brief description of the load balancing we use with DFP in this section. An important aspect of the load balanced routing approach we describe is its simplicity. In the rest of the section, we describe the load balanced routing in the context of the DSR protocol. However, the routing strategy is simple enough to be realized along with any reactive routing protocol.

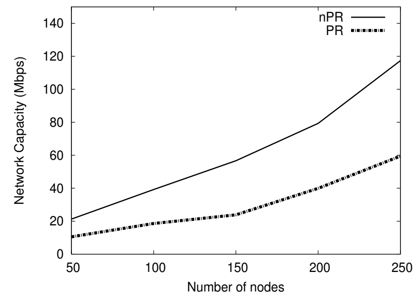
To participate in the load balanced routing process, every node in the network keeps track of the number of active flows traversing it on a periodic basis. Essentially, every node maintains a list of active flows. When a packet arrives from a particular flow, the time associated with the corresponding entry in the list is refreshed with the current time. Periodically, elements in the list, not refreshed since the last time the list was monitored, are removed. The refreshing is done at a granularity of one-second.

When a route-request (RREQ) is sent by the source, every intermediate node that forwards the RREQ message by stamping its identifier on the packet header (just as in DSR). In addition, the packet header is extended to have a *max-contention* (MC) field that is set to *zero* by the source. When the intermediate node forwards the packet, it checks to see if the number of active flows it serves is more than the MC value on the incoming packet header. If yes, it sets the MC field to the number of active flows served.

When routes are selected by the source, instead of merely using the hop-count to differentiate routes, the tuple ($\frac{1}{MC}$,hop-length) is used to lexicographically compare the different routes. The route with the minimum lexicographic value for the tuple is selected for use.



a) Throughput



b) Network Capacity

Fig. 9. Varying number of nodes

Figure 8 shows the amount of performance improvement when using the simple LBR in *nPR*. We also present the performance improvements achieved when using the idealized LBR in *nPR* and *PR*. It is to be noted that even when using a simple load balancing strategy, the same degree of benefits established for the idealized load balancing scheme can be attained.

V. Performance Evaluation

A. Simulation Environment

We evaluate the performance of DFP (distributed realization of *nPR*) in this section. The *ns2* [14] network simulator is used for the experiments. CBR (Constant Bit Rate) is used as the data generating application and UDP (User Datagram Protocol) is used as the transport protocol. DSR (Dynamic Source Routing) serves as the routing protocol. IEEE 802.11 in DCF (Distributed Co-ordination Function) mode is used as the MAC protocol. Two-ray ground model is assumed to be the propagation model with a constant transmission radius of 250m. DFP sits atop the routing layer, but beneath the transport layer. The DFP realization includes all three elements discussed in the last section. The number of nodes is maintained at 100 in a topology of 1000m X 1000m. Each of the sources generates data at the rate of 400 Kbps.

The scenarios considered for the simulation studies can be classified as follows: (i) varying number of nodes maintaining constant node-density, (ii) varying load, (iii) impact of mobility, and (iv) impact of traffic patterns. DFP is compared with the conventional protocol stack (CBR/UDP/DSR/802.11) which is the distributed realization of *PR*. We refer to DFP as *nPR* and the conventional pipelined approach as *PR* in all our simulation results. The metrics used for comparison are:

- Aggregate throughput: measured as the sum of the throughputs obtained by all the flows in the network (Kbps)
- Network capacity: measured as the total number of successful transmissions in the network over the simulation duration (Mbps)
- Normalized standard deviation: measured by dividing the standard deviation of the throughput of all the flows by the mean throughput (no unit)
- Number of route errors

Each of the data points in the results presented is averaged over 20 random seeds.

B. Performance Evaluation

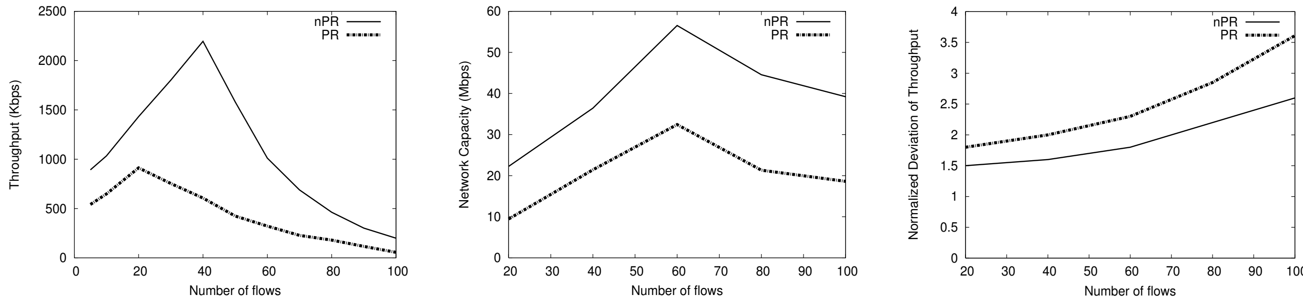
1) Varying number of nodes: First, we study the impact of the number of nodes on the performance of *nPR*. The size of the network is appropriately increased with increasing number of nodes in order to maintain constant node density. With increasing network size, spatial reuse and hence performance increases for both *nPR* and *PR*. However, the increase in the number of nodes also brings in more opportunity for biasing the shorter hop flows using *nPR* and hence improves performance further. This can be observed in Figure 9(a).

As the network size increases, there is an increase in the number of contention regions in the network. This in turn increases the probability of finding under-utilized contention regions. Recall from discussions in Section III that as the number of under-utilized contention regions increases, the network capacity using *PR* will go down which in turn explains the performance improvement obtained by *nPR* in 9(b).

2) Varying load: We study the impact of load on the operation of *nPR* and *PR*. As seen from Figure 10(a), peak throughput of *nPR* occurs at a higher load than *PR* indicating that a higher fraction of the available network capacity is achieved in the case of *nPR*. The proportional rate adaptation of *nPR* exploits available spatial reuse and saves it from under-utilizing the network capacity at low loads. In the optimal region, *nPR* performs significantly better than *PR* due to the reduction in the number of mini-flows contending in each contention region. However, as the number of flows increases, both *nPR* and *PR* enter the over-utilized region and hence they tend to perform similarly.

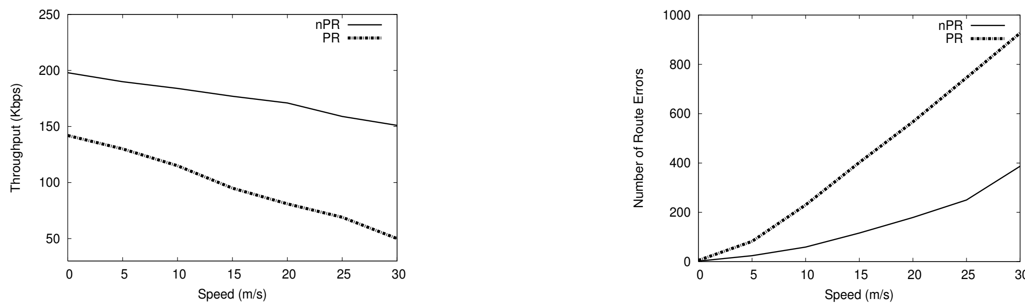
The capacity of the network also follows a similar trend as the throughput results as seen in Figure 10(b). The performance improvement of *nPR* over *PR* can be attributed to the reduction in the number of mini-flows in the network and the consequent reduction in the distributed overheads such as (channel) resource wastage due to back-offs.

It is interesting to note that *nPR* achieves lower normalized standard deviation than *PR* in Figure 10(c). The expected deviation in throughput due to the hop length bias can be explained to be offset by (i) the increase in the average



a) Throughput b) Network Capacity c) Normalized standard deviation

Fig. 10. Varying number of flows



a) Throughput b) Route errors

Fig. 11. Impact of mobility

throughput of the flows, and (ii) decrease in the max-min unfairness within a contention region by virtue of reducing the number of mini-flows per contention region.

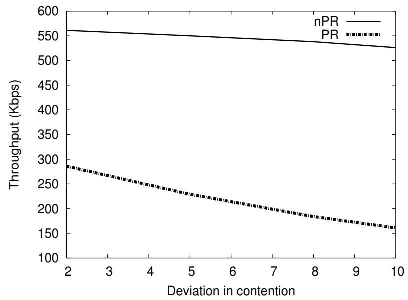
3) Impact of mobility: The impact of mobility on the performance of *nPR* and *PR* is observed by varying the speed of the nodes in the network. The random-walk mobility model was used where nodes pick a random point in the topology and choose a random speed to move towards the chosen point. The throughput results are plotted in Figure 11(a). It can be seen that the degradation in throughput is lesser for *nPR* than for *PR*. This is because the reduction in channel contention and hence the increased throughput using *nPR* effectively reduces the lifetime of data transfer for the flows. Hence, the vulnerability of the flows to mobility-induced route failures is significantly reduced using *nPR*, thereby increasing the throughput. This in turn can be corroborated by recording the number of route errors observed in the two cases in Figure 11(b) where the number of route errors increases at a much faster rate for *PR* than for *nPR* with increasing mobility.

4) Impact of traffic patterns: While the above experiments were conducted by choosing the source-destination pairs randomly, to study the impact of non-uniform traffic patterns, we consider two components that help us deviate the traffic pattern from a uniform distribution, namely (i) deviation in the contention level, and (ii) deviation in the hop length of the flows.

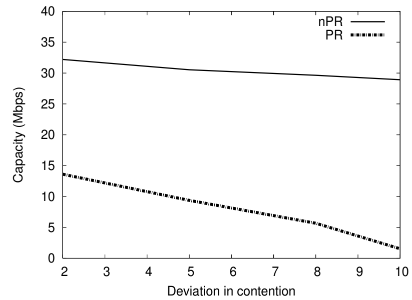
Deviation in contention level: The number of contending flows in each contention region is varied. This is achieved using a combination of three methods : (i) weighted choice of destination for the flows, (ii) weighted distribution of nodes in the network, and (iii) creating artificial routing hot-spots

The throughput results are plotted in Figure 12(a). The throughput of a flow in *PR* depends on the bottleneck (maximum) contention level along its path. Hence, when the deviation in contention level increases, there is an increase in the maximum contention level possible in the network and hence indirectly a decrease in throughput. *nPR* on the other hand, is not impacted by the deviation in the contention level in the network. In the case of network capacity, increasing the contention level increases the maximum possible contention level in the network and hence increases the probability of under-utilized contention regions in *PR* as observed in Figure 12(b). This is also true for the normalized standard deviation results for *PR* recorded in Figure 12(c). In addition, the reasons attributed for *nPR*'s better fairness performance in Section III-B are still applicable here.

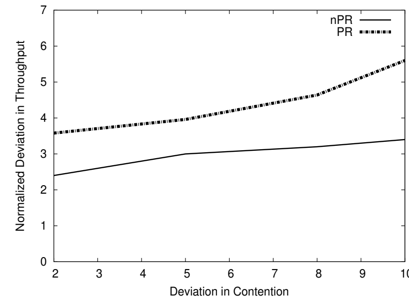
Deviation in hop length: The range of hop lengths used by the flows in the network is varied to introduce deviation in the hop lengths of the flows. Note that the performance of *nPR* is sensitive to the range of hop lengths of the flows. However, *PR* on the other hand is not impacted by the hop lengths of the flows but only on the bottle-neck contention level along the path. Hence, in Figure 13(a) the throughput of *nPR* increases with increasing range of hop lengths due to the ability to bias shorter hop flows more. *PR* on the other hand, does not exhibit any variation in throughput as expected. However, changing the range of hop lengths does not change the network capacity as observed in Figure 13(b). The



a) Throughput



b) Network Capacity



c) Normalized standard deviation

Fig. 12. Varying contention level

improvement in network capacity of *nPR* over *PR* comes from being able to fundamentally utilize network resources better in the presence of different contention levels in the network and not because of biasing shorter hop flows.

Figure 13(c) shows an interesting result, wherein the normalized standard deviation for *nPR* increases with increasing hop length range to increase beyond that of *PR* for higher hop lengths. This is because at higher hop lengths, the bias toward shorter hops is significant to introduce large deviations in the throughputs enjoyed by different flows of varying hop lengths. This in turn increases the normalized standard deviation such that even the off-setting factors outlined in Section III-B, namely the increase in the average throughput of the flows, and decrease in the location dependent max-min unfairness within a contention region, are unable to help lower the normalized deviation. In the case of lower range of hop-lengths, we still observe *nPR* to perform better than *PR* with help from the off-setting factors. Since the average hop lengths are typically not as high as 10, this explains why *nPR* is able to provide better fairness even in terms of normalized standard deviation when compared to *PR*.

VI. Discussions

A. nPR and TCP

The impact of *nPR* on the TCP transport layer protocol is an interesting issue. TCP is by far the most dominant transport layer protocol used in the Internet, and hence is given considerable importance when it comes to backward compatibility issues for new strategies and protocols. However, note that the need (or lack thereof) to use TCP in wireless ad-hoc networks on an as-is basis is firmly dependent upon the type of ad-hoc network applications considered. In [15], the authors argue for not using TCP as the transport layer mechanism by showing how each of its mechanisms are fundamentally inappropriate for the target environment.

However, we now discuss the potential impact on TCP because of using a communication strategy such as *nPR*. If TCP is used with *nPR*, there is exactly one mechanism - TCP's retransmission timeout (RTO) calculation mechanism - in TCP that will be negatively impacted. Specifically, TCP sets its RTO value as follows:

$$RTO \leftarrow rtt_{avg} + 4 * rtt_{mdev}$$

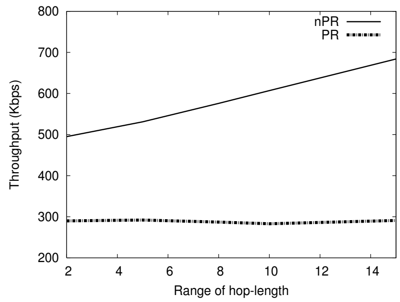
where rtt_{avg} is the average end-to-end Round Trip Time (RTT) and rtt_{mdev} is the mean deviation in the RTT values. Considering the use of TCP with *nPR* and assuming the network is loaded enough to allow only one packet to be in transit, every packet sent out by TCP will traverse the rtt delay independently. In other words, when TCP sends out a burst of $cwnd$ packets in a congestion window, the first packet will experience an rtt of rtt_{actual}, while packet i will experience an rtt of $rtt_{actual} * i$.

Such a variation in the rtt values experienced by the different packets in a congestion window will increase rtt_{mdev} substantially, which will finally result in highly inflated RTO values. The inflation in RTO values will negatively impact a connection's performance if the connection experiences a timeout, as TCP will wait for RTO amount of time before actually inferring the corresponding loss. Timeouts in TCP can occur under one of the following conditions (i) more than $cwnd - 4$ packets are lost, (ii) there are suffix losses, and there is no more application data to send, or (iii) a retransmitted packet is lost. If timeouts do not occur, then TCP's performance will not be adversely affected because of *nPR*.

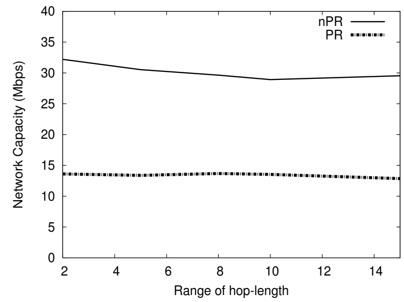
Finally, if the variation in rtt samples needs to be avoided, a simple strategy that would involve the holding back of the incoming ACKs by the sender side DFP till all ACKs within a congestion window worth of packets are received will address the problem. In such a mechanism, only the receipt of DUPACKs will be released to the TCP layer without having to wait for all ACKs. Essentially, such a scheme would allow for TCP to experience the same rtt for all packets within a congestion window. Note that the obvious concern in such a scheme of potential bursts (because of the ACK bunching) into the network is not valid in *nPR* because *nPR* performs implicit rate shaping due to its one-packet-in-transit principle.

B. Nature of applications

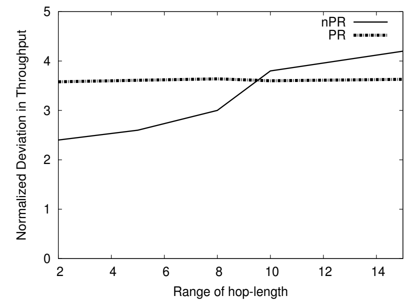
While it may appear that *nPR* is not suitable for real-time applications because of its one-packet-in-transit (non-pipelined) principle, note that the effective end-to-end throughput is better than that of *PR*. Hence, its inter-packet separation at the receiving end will be better than that of *PR*. In *PR*, despite the fact that packets are pipelined, the bottlenecks on



a) Throughput



b) Network Capacity



c) Normalized standard deviation

Fig. 13. Varying hop length

the path prevent it from attaining the end-to-end throughput possible in *nPR*, and consequently results in larger inter-packet separation for the arriving packets at the receiving end. The above arguments hold when the real-time application is generating packets at a rate greater than or equal to the available end-to-end throughput. However, when the application itself is bottlenecked (non-backlogged), the maximum rate achievable by both *nPR* and *PR* is the application rate itself. In summary, the *nPR* strategy, unlike other strategies to improve on network performance [2] does not limit the kind of applications that can be supported by the ad-hoc network.

C. *nPR and Wireline Networks*

An important question to answer in the context of this paper is: Is *nPR* the right strategy to employ in wireline networks also, or is its efficacy specific to multi-hop wireless ad-hoc networks? The answer to the question is the latter. *nPR*'s efficacy is solely due to the unique nature of wireless ad-hoc networks, and will not perform better than *PR* in a wireline environment.

Specifically, the non-pipelined nature of *nPR* reduces the number of outstanding packets in the network to n, where n is the number of nodes and sources in the network. This translates to an offered load of $O(log(n))$ mini-flows per contention region (given that there are $O(\frac{n}{log(n)})$ contention regions in a minimally connected network [1]). This represents a sufficient enough load per contention-region to keep network resources from being under-utilized (see Section III for illustration of the capacity results for *nPR*). However, in a wireline network, the number of contention regions is greater than or equal to the number of nodes in the network. Conservatively, if a node can transmit and receive only one packet at a time, the number of contention regions is n. However, if multiple independent processors are assumed to serve the different links at every node, the number of contention-regions can be upto $O(n * log(n))$. Under either condition, if the number of outstanding packets is reduced to merely n, that represents exactly 1 outstanding packet per contention region in the earlier scenario, and less than 1 packet per contention region in the latter.

Thus, the key characteristic of wireless ad-hoc networks that enables *nPR* to function efficiently due to its non-pipelined nature is its property that the number of contention regions is significantly smaller than that of the number of nodes in the network, which in turn keeps the network capacity of *nPR* at-least as high as that of *PR*.

D. *Size of in-transit data*

In earlier sections, we indicated that the specific size of the in-transit data in *nPR* has no bearing on its performance. Specifically, the amount of in-transit data for a flow can be multiple packets *as long as exactly one packet is in flight at any given point in time*. This can be further visualized as follows: Let the amount of in-transit data for a flow be B packets. In *nPR*, each intermediate node along the path would wait to receive all B packets, before they forward the data. This would ensure that exactly one packet is in flight on the end-to-end path, although the amount of data in-transit along the path is actually B packets.

Note that B has no bearing on the throughput performance for flows (as transmitting B packets would entail B times longer delay for the in-transit data to reach the destination when compared to a single packet). However, the proper choice of B might have practical applications. For example, the overheads due to the ACK packet sent back by the destination can be amortized with a larger B. However, there are some drawbacks in using a large B. These include : more number of packets lost for every route failure (assuming no route salvaging by intermediate nodes), and potential buffer availability problems at intermediate nodes to store B packets of data from multiple hosts. However, the optimal value of B can be determined to be the maximum value such that the expected number of lost packets due to route failures is the same as that in *PR*. Thus, the impact on *nPR* due to route failures (on a per route-failure basis) will still be as bad as that of *PR*, while the cost of the ACK packets are amortized. Note that the above discussion is completely orthogonal to the advantage *nPR* has in terms of minimizing the expected number of route failures per connection due to its superior end-to-end throughput performance, which still holds.

E. *Self contention*

Finally, there is one practical by-product of non-pipelining that we did not focus in this paper: alleviation of self-contention. The presence of self-contention has been explored in related work [16], and has been shown to be detrimental, specifically in the context of TCP. Although not the objective of this work, *nPR* implicitly alleviates self-contention, and hence will have similar benefits to those identified in [16]. However, note that self-contention is not a problem when using a centralized scheduler, and hence the fundamental benefits of *nPR* established in this work are not due in any way contributed to because of the alleviation of self-contention.

VII. Related Work

There have been several works that have contributed to the understanding of network capacity and utilization in multi-hop wireless networks [1], [2], [17], [18], [19], [5]. [1] identifies the capacity bounds for arbitrary and random static ad-hoc networks and shows that the per-node throughput does not scale for increasing node densities. On the other hand when mobility is considered, [2] showed that it is possible for the per-node throughput to scale. However, the delay incurred in the data transfer is not bounded by the mechanism. [19] has tried to bridge the gap between [1] and [2] by proposing a routing algorithm that aims to help the per-node throughput scale, while at the same time bounding the delay incurred in the delivery of data packets. [5] has recently analyzed the optimal order of delay incurred for a given throughput in random wireless networks with the incorporation of mobility, and has also suggested a scheme that helps achieve this optimal order of delay. [18] is another work in this category that analyzes the capacity regions of ad-hoc networks in the presence of multiple hops, power control and time division. Since the goal of most of these works is to obtain fundamental scaling laws and bounds for parameters of interest, the network configurations and traffic patterns are assumed to be homogeneous in these works. Furthermore, a pipelined packet forwarding model is considered in all of these works. The focus of our work, on the other hand, is a non-pipelined packet forwarding model, which in idealized and practical network configurations with different sets of traffic patterns, shows much better aggregate capacity and fairness than that obtained by the pipelined model.

VIII. Conclusion

In this work, we argue that the default pipelined model of communication assumed for ad-hoc networks is inappropriate. In this regard, we have presented a new communication strategy called the non-pipelined relay (*nPR*) model that provides improvements in throughput and transport capacity and conforms to the proportional fairness paradigm. Further, we provide reasons and arguments for improvements in practical network conditions. Finally, we present a simple forwarding protocol that realizes the proposed *nPR* paradigm and evaluate its performance comprehensively over a variety of network configurations and parameters.

Acknowledgment

This work was partially funded by the Georgia Tech Broadband Institute, and NSF grants ECS-0428329, ANI-0117840, ECS-0225497, and CCR-0313005.

References

[1] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, Mar 2000.

[2] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, Aug 2002.

[3] C. Peraki and S. D. Servetto, "On the Maximum Stable Throughput Problem in Random Networks with Directional Antennas," in *Proceedings of ACM MOBIHOC*, Jun 2003.

[4] S. Yi, Y. Pei, and S. Kalyanaraman, "On the Capacity Improvement of Ad Hoc Wireless Networks Using Directional Antennas," in *Proc. of ACM MOBIHOC*, Jun 2003.

[5] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-Delay Trade-off in Wireless Networks," in *Proceedings of IEEE INFOCOM*, Hong Kong, Mar 2004.

[6] H-Y. Hsieh and R. Sivakumar, "IEEE 802.11 over Multi-hop Wireless Networks: Problems and New Perspectives," in *Proceedings of IEEE Vehicular Technology Conference (VTC)*, Sep 2002.

[7] H. Luo, S. Lu, and V. Bhargavan, "A New Model for Packet Scheduling in Multi-hop Wireless Networks," in *Proceedings of ACM MOBICOM*, Boston, MA, 2000.

[8] T. Nandagopal, T-E. Kim, X. Gao, and V. Bhargavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," in *Proceedings of ACM MOBICOM*, Aug 2000.

[9] F. P. Kelly, A. Maulloo, and D. Tan, "Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, Mar 1998.

[10] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug 1993.

[11] W-C. Feng, K. G. Shin, D. D. Kundlur, and D. Saha, "The BLUE Active Queue Management Algorithms," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 513–528, Aug 2002.

[12] K. Xu, M. Gerla, L. Qi, and Y. Shu, "Enhancing tcp fairness in ad hoc wireless networks using neighborhood red," in *Proceedings of ACM MOBICOM*, 2003, pp. 16–28.

[13] "NS2: The Network Simulator," in *http://www.isi.edu/nsnam/ns*.

[14] K. Sundaresan, V. Anantharaman, H-Y. Hsieh, and R. Sivakumar, "ATP: A Reliable Transport Protocol for Ad-hoc Networks," in *Proceedings of ACM MOBIHOC*, Jun 2003.

[15] Zhenqiang Ye, Dan Berger, Srikanth Krishnamurthy, Michalis Faloutsos, Satish K. Tripathi, and Prasun Sinha, "Alleviating MAC Layer Self-Contention in Ad-hoc Networks," in *Student Poster ACM MOBICOM*, San Diego, September 2003.

[16] J. Li, C. Blake, D. De Couto, H. I. Lee, and R. Morris, "Capacity of Ad Hoc Wireless Networks," in *Proceedings of ACM MOBICOM*, 2001, pp. 61–69.

[17] S. Toumpis and A. Goldsmith, "Capacity Regions for Wireless Adhoc Networks," in *Proceedings of International Symposium on Communication Theory and Applications*, 2001.

[18] N. Bansal and Z. Liu, "Capacity, Delay and Mobility in Wireless Ad Hoc Networks," in *Proceedings of IEEE INFOCOM*, Apr 2003.