A Distributed Routing Protocol for Predictable Rates in Wireless Mesh Networks[†]

Behnaz Arzani, Roch Guerin, and Alejandro Ribeiro
Department of Electrical & Systems Engineering, University of Pennsylvania
(barzani, guerin, aribeiro)@seas.upenn.edu

***Abstract*—Wireless mesh networks hold the promise of rapid and flexible deployments of communication facilities. This potential notwithstanding, the often erratic behavior of multihop wireless transmissions is limiting the range of applications that such networks can target. In this paper we investigate the feasibility and benefits of a routing protocol explicitly aimed at making wireless mesh networks more predictable while preserving their efficiency and flexibility. The protocol's basic premise is the classical idea that a *multipath* solution can offer resiliency to unexpected link variations. The paper's contributions are in demonstrating how this can be effectively realized in a wireless context, and in offering initial evidences of its efficacy. In particular, the paper illustrates how routing decisions that account for link variability can be computed in a distributed fashion, and the benefits they afford in improving the stability of end-to-end transmission rates even in the presence of random network fluctuations.**

I. Introduction

Wireless mesh networks boast many advantages over traditional wired networks, but the wireless medium is often unpredictable. This makes delivering reliable communication difficult. Traditional solutions for overcoming network changes involve protocols designed to *react* to those changes. Routing protocols are a prime example, as they adapt forwarding decisions once notified of network changes. However, even the most efficient protocol has a finite reaction time, which limits its efficacy especially in settings where changes occur relatively frequently, as is the case in wireless mesh networks.

Awareness of those limitations is in part behind the recent interest in multi-path solutions. This interest has manifested itself across most layers of the protocol stack. At the link layer, the combination of (diversity) coding schemes and multipath was for example explored to facilitate recovery from network failures [4]. At the network layer, there have been numerous protocols that have sought to leverage multiple paths to improve either performance or reliability; see *e.g.,* [13]. At the transport layer, multipath (MP)TCP and related efforts [5], [6], [8], [17], are among the more visible such activities. They have investigated various performance and implementation aspects of congestion control when using several paths simultaneously, and demonstrated the benefits this affords. There are also many works that explore the combination of coding schemes and multipath to improve transmission stability over lossy networks [14], [15], especially for real-time applications such as video [3].

In short, there is plenty of evidence that multipath solutions increase resiliency to disruptions and links variability, because it is unlikely that all paths are simultaneously affected. However, blindly seeking multiple paths can have disadvantages as well; *e.g.,* it may be that not all links have the same quality and combining them may produce an overall poorer result. It is, therefore, important to ensure an approach to multipath computations that is principled and incorporates clear objectives. The challenge is to realize a reasonable trade-off between the rigor of the protocol's objectives and the complexity of implementing it. For example, the DIV-R protocol of [11] sought robust multipath solutions by maximizing the number of next-hops at each node, and relied on an approximate objective function amenable to a distributed optimization solution. As shown in [10], while this offered reasonable resiliency to link failures, it did not perform as well when confronted to node failures.

Our goals are, therefore, twofold. First, to formulate a clear objective function that reflects performance predictability in the face of the many uncertainties that prevail in wireless networks. Second, to develop a practical distributed algorithm that realizes this objective, and evaluate its performance to better gauge the benefits of such an approach. Our objective function calls for computing paths that meet target (average) rate guarantees while minimizing the *variance* of the rates. Rate guarantees account for a flow's long-term throughput. Minimizing rate variances fosters rate stability in the presence of random link fluctuations.

Our solution relies on a distributed optimization framework originally proposed in [18], which we extend to enable a practical protocol implementation. The implementation assumes the availability of local link statistics, *i.e.,* link reliability and transmission rates. In wireless networks, these depend on numerous factors; some exogeneous such as noise and fading, and others that are affected by the network and its operation, *e.g.,* interferences, collisions, etc. There is a vast literature, *e.g.,* [1], [7], [16], on how to carry out measurements to estimate the combined impact of these factors on link statistics. In this paper, we assume the availability of such information; at least to the extent that nodes can estimate the mean and variance of local transmission rates. Designing and implementing such estimation procedures is a topic for future work.

The paper describes the Distributed Reduced Variance Routing (DRVR) multipath computation protocol, and makes two contributions. The first is in developing and validating an effective distributed optimization solution, which accounts for a number of practical deployment aspects such as convergence and infeasibility detection. The second is in assessing the benefits of such an approach compared to solutions that primarily seek to maximize transmission rates. In particular, we show that while a

[†] The work was supported by NSF grant CNS-1116039.

978-1-4673-2447-2/12/$31.00 © 2012 IEEE

multipath approach yields a slightly lower long-term guaranteed rate, it substantially reduces rate fluctuations; especially at the time-scale of relevance to real-time, interactive applications (see Section IV-A2 for details)

The rest of the paper is organized as follows. Section II reviews the optimization framework, discusses issues that arise in the context of a distributed implementation, and presents properties of the proposed distributed optimization solution. Section III introduces our evaluation methodology, including metrics, scenarios, and an alternate approach that will be used as a benchmark. Section IV is devoted to evaluating the benefits of the proposed multipath scheme, and highlighting the impact of different parameters. Finally, Section V summarizes the paper's findings and discusses several ongoing extensions.

II. Minimum Local Variance Routing

Wireless networks are characterized by the variability of links. At small time scales ranging from microseconds to a few milliseconds, fading and interference are the dominant phenomena. At mesoscales ranging from around 10 milliseconds to a few seconds, channel variations appear in the form of shadowing and other slow variations in channel gains. At larger time scales, variations occur due to spatial reconfigurations and node failures. Wireless physical layers incorporate effective techniques to counteract microscale fading effects while medium access (MAC) layers incorporate contention protocols to handle microscale interference. Macroscale variations can be handled by reconfiguring tables at the routing layer. Mesoscale variations, however, are too fast to allow recomputation of routes and too slow to be handled by the physical or MAC layer. Our purpose is to introduce a routing protocol to handle variations in link quality at these intermediate time scales.

For a precise problem formulation, consider a wireless network with N nodes, and let R_{ij} denote the transmission rate from i to j as perceived at the routing layer. Implicit in this rate is an average of the small time scale rates perceived at the physical and MAC layers. As such R_{ij} represents an average across fading states and contention resolutions. Conventional routing protocols assume these rates as given, and deal with variations in R_{ij} by reconfiguring routing tables. This is effective for long time scale variations but ineffective for changes that happen at intermediate time scales. To account for these variations we explicitly model changes in these rates through a stochastic model. Specifically, rates R_{ij} are modeled as random variables whose means and variances

$$\widehat{R}_{ij} := \mathbb{E}\left[R_{ij}\right], \qquad \sigma_{ij}^2 := \mathbb{E}\left[\left(R_{ij} - \widehat{R}_{ij}\right)^2\right], \tag{1}$$

are available to the routing layer[1]. The set of nodes that can communicate with node i, *i.e.,* the nodes j for which $\widehat{R}_{ij} > 0$, is termed the neighborhood of i and denoted $n(i)$. Node i knows $\widehat{R}_{ij}$ and σ_{ij}^2 only for $j \in n(i)$.

We consider K information flows. Without loss of generality we assume the first K nodes, $1, \ldots, K$, are the respective destinations of flows $1, \ldots, K$. To handle rate variations we implement multipath routing, which we realize through the introduction of routing variables T_{kij} to control traffic splitting. Variable T_{kij} represents the fraction of transmission opportunities node i allocates for transmissions to neighboring node j of packets whose final destination is k. As such, we have for all nodes i,

$$\sum_k \sum_{j \in n(i)} T_{kij} \leq 1. \tag{2}$$

If the achievable rate on link $i \to j$ is R_{ij}, and a fraction T_{kij} of transmission opportunities is allocated to neighbor j and flow k, the traffic that node i sends to node j for this particular flow is $T_{kij}R_{ij}$. The total flow k traffic out of node i is then the sum $\sum_{j \in n(i)} T_{kij}R_{ij}$. Conversely, the total flow k traffic into node i includes packets $\sum_{j \in n(i)} T_{kji}R_{ji}$ received from neighbors. The (end-to-end) transmission rate a_{ki} available to *local* arrivals for flow k at node i therefore satisfies

$$a_{ki} = \sum_{j \in n(i)} T_{kij}R_{ij} - \sum_{j \in n(i)} T_{kji}R_{ji} \;\forall\, k, i. \tag{3}$$

Larger rates are favored, but the principal objective is to realize a minimum required rate of $a_{0,ki}$ that exceeds the local arrival rate for flow k at node i. If we guarantee $a_{ki} \geq a_{0,ki}$ for all nodes i and flows k, then all buffers are stable and information is eventually delivered to the destination. However, it is not possible to know whether $a_{ki} \geq a_{0,ki}$ because the link rates R_{ij} that determine a_{ki} in (3) are unknown at the routing layer. In fact, given the rate model in (1), the end to end rates a_{ki} are random variables. In this context we reinterpret the rate a_{ki} as the local belief that node i has regarding the actual end-to-end rate for flow k.

We therefore proceed to let node i work with its local belief a_{ki} in order to meet traffic requirements on average while minimizing the variance of the rates a_{ki}. To be specific, take expectation on both sides of (3) to write the expectation of a_{ki},

$$\mathbb{E}\left[a_{ki}\right] = \sum_{j \in n(i)} T_{kij}\widehat{R}_{ij} - \sum_{j \in n(i)} T_{kji}\widehat{R}_{ji}. \tag{4}$$

Likewise, the variance of a_{ki} follows from (3) as

$$\text{var}a_{ki} = \sum_{j \in n(i)} T_{kij}^2\sigma_{ij}^2 + \sum_{j \in n(i)} T_{kji}^2\sigma_{ji}^2. \tag{5}$$

To satisfy traffic requirements on average we impose the constraint $a_{0,ki} \leq \mathbb{E}\left[a_{ki}\right]$ and search for routing variables T_{kij} that result in the minimum possible sum variance $\sum_{i,k} \text{var}a_{ki}$. Using the explicit expressions in (4) and (5), we select optimal routing variables T_{kij}^* that solve the optimization problem

$$\begin{aligned} T_{kij}^* = \text{argmin} \; & \sum_{k,i,j} T_{kij}^2\sigma_{ij}^2 + \sum_{k,i,j} T_{kji}^2\sigma_{ji}^2 \\ \text{s.t.} \; & \sum_k \sum_{j \in n(i)} T_{kij} \leq 1, \quad T_{kij} \geq 0, \\ & a_{0,ki} \leq \sum_{j \in n(i)} T_{kij}\widehat{R}_{ij} - \sum_{j \in n(i)} T_{kji}\widehat{R}_{ji}. \end{aligned} \tag{6}$$

The problem defined in (6) is a convex quadratic program and so can be solved in polynomial time. However, solving the problem

[1] As mentioned earlier, a number of approaches are available to design effective estimation procedures for those variables. Selecting a particular one is, however, beyond the scope of this paper.

as stated, requires knowledge of the entire network, which forces a centralized implementation. Centralized computations are not desirable, as they induce the need to gather information about all network nodes at a central location. This results in numerous message exchanges, longer computation time, and frequent re-computations as a byproduct of local changes in the network. For this reason, a distributed solution is preferred. The next section outlines a solution introduced in [18].

A. Distributed implementation algorithm

Distributed implementation of convex optimization algorithms can be obtained by working in the dual domain. For that purpose, [18] introduces Lagrange multipliers λ_{ki} associated with the constraints $a_{0,ki} - \sum_{j \in n(i)} T_{kij}\widehat{R}_{ij} + \sum_{j \in n(i)} T_{kji}\widehat{R}_{ji} \leq 0$ in (6), and defines the Lagrangian as

$$\mathcal{L}(T_{kij},\lambda_{ki}) = \sum_{k,i,j} T_{kij}^2 \sigma_{ij}^2 + \sum_{k,i,j} T_{kji}^2 \sigma_{ji}^2 \\ + \lambda_{ki}\left(a_{ki} - \sum_{j \in n(i)} \widehat{R}_{ij} T_{kij} + \sum_{j \in n(i)} T_{kji}\widehat{R}_{ji}\right). \quad (7)$$

Define a time index n and let $\lambda_{ki}(n)$ be given multiplier values at time n. Primal Lagrangian minimizers are defined as

$$T_{kij}(n) = \underset{T_{kij} \geq 0,\ \sum_{k,j} T_{kij} \leq 1}{\operatorname{argmin}} \mathcal{L}(T_{kij}, \lambda_{ki}(n)). \quad (8)$$

Denote as $S_{ki}(n)$ the gradient component along the dual direction at $\lambda_{ki} = \lambda_{ki}(n)$ obtained by evaluating the slack of the constraint enforced by λ_{ki},

$$S_{ki}(n) = a_{0,ki} - \sum_{j \in n(i)} T_{kij}(n)\widehat{R}_{ij} + \sum_{j \in n(i)} T_{kji}(n)\widehat{R}_{ji}. \quad (9)$$

It can be observed that by reordering terms in (7) we can define local Lagrangians

$$\mathcal{L}_i(T_{kij},\lambda_{ki}) = \quad (10) \\ \sum_{k,j} 2T_{kij}^2\sigma_{ij}^2 + \lambda_{ki}\left(a_{ki} - \sum_{j \epsilon n(i)} T_{kij}\widehat{R}_{ij} - \widehat{R}_{ji}T_{kji}\right),$$

so as to decompose the (global) Lagrangian into the sum

$$\mathcal{L}(T_{kij}, \lambda_{ki}) = \sum_i \mathcal{L}_i(T_{kij}, \lambda_{ki}). \quad (11)$$

Thus, to compute the Lagrangian minimizers in (8), it suffices to find arguments that minimize local Lagrangians

$$T_{kij}(n) = \underset{T_{kij} \geq 0,\ \sum_{k,j} T_{kij} \leq 1}{\operatorname{argmin}} \mathcal{L}_i(T_{kij}, \lambda_{ki}). \quad (12)$$

The minimization in (12) can be expressed in closed form with the aid of the non-negative auxiliary variable $\delta_i(n) \geq 0$. This variable is chosen to guarantee that $\sum_{k,j} T_{kij}(n) \leq 1$ when the $T_{kij}(n)$ variables are given by

$$T_{kij}(n) = \left[\frac{1}{\sigma_{ij}^2}\left(-\delta_i(n) + \lambda_{ki}(n)\widehat{R}_{ij} - \lambda_{kj}(n)\widehat{R}_{ij}\right)\right]^+, \quad (13)$$

where $[x]^+ := \max(x, 0)$. If for $\delta_i(n) = 0$, the $T_{kij}(n)$ resulting from (13) satisfy $\sum_{k,j} T_{kij}(n) \leq 1$, these variables are the optimal solution[2] to (12). Otherwise, we determine the strictly positive $\delta_i(n) > 0$ that makes the $T_{kij}(n)$ resulting from (13) satisfy $\sum_{k,j} T_{kij}(n) = 1$.

With the gradients available as per (9), dual variables are updated along the gradient direction. This update is given by

$$\lambda_{ki}(n+1) = \left[\lambda_{ki}(n) + \epsilon_{ki} S_{ki}(n)\right]^+, \quad (14)$$

where ϵ_{ki} is the stepsize corresponding to the dual direction λ_{ki}.

The algorithm determined by iterative application of (13)-(14) computes $T_{kij}(n)$ that converge to optimal routing variables T_{kij}^* as the iteration index n grows. To compute the primal iterates in (13) node i requires access to local multipliers $\lambda_{ki}(n)$ and multipliers $\lambda_{kj}(n)$ available at neighboring nodes $j \in n(i)$. To compute the dual iterates in (14), node i accesses local routing variables $T_{kij}(n)$ and $T_{kji}(n)$ of all neighbors $j \in n(i)$. We transform (13)-(14) into the DRVR protocol in the next section.

B. DRVR Protocol

A distributed implementation of (13)-(14) requires synchronizing primal and dual updates among all nodes in the network. This is unrealistic in practice, and motivates an asynchronous implementation in the form of a protocol, DRVR, whose operation is summarized in Algorithm 1. In DRVR, each node updates routing variables $T_{kij}(n)$. Node i also updates Lagrange multipliers $\lambda_{ki}(n)$ that serve as auxiliary variables used to determine suitable values for $T_{kij}(n)$. In this asynchronous distributed implementation we further introduce local versions of neighboring variables. We denote as $\lambda_{kj}^{(i)}$ the values of the multipliers of node j as stored in node i. Likewise we denote as $T_{kji}^{(i)}$ the values of the routing variables of node j as stored in node i. Algorithm 1 also assumes that average rates, $\widehat{R}_{ij}$ and $\widehat{R}_{ji}$, and variances, σ_{ij}^2 and σ_{ji}^2, for all links in and out of node i are available at the routing layer of node i.

1) **Protocol Iterations:** The core steps in DRVR are operations that correspond to (13)-(14) but using local versions of neighboring variables. The update of the primal variables is performed in Step 4 of Algorithm 1. The update coincides with the expression in (13) except for the substitution of the local copies $\lambda_{kj}^{(i)}(n)$ for the neighboring multipliers $\lambda_{kj}(n)$. The dual variable updates correspond to steps 8 and 9, which compute the gradient component $S_{ki}(n)$ and the new value of the local dual variables $\lambda_{ki}(n+1)$, respectively. In the computation of the gradients $S_{ki}(n)$ we use (9) with $T_{kji}^{(i)}(n)$'s used in lieu of their actual values $T_{kji}(n)$. The dual variable updates in (14) and Step 9 look the same, except that the gradient $S_{ki}(n)$ in (14) is replaced in Step 9 by the local constraint slack computed in Step 8.

Whenever a multiplier or a dual variable is updated, it is immediately scheduled for transmission to neighboring nodes. When the routing variable $T_{kji}(n)$ is updated in Step 4, a message with the new value of $T_{kji}(n)$ is scheduled for transmission to node $j \in n(i)$ as indicated in Step 5. When this new primal variable is received at node j, it triggers a recomputation of dual variables, but this is beyond the purview of node i. When the multipliers $\lambda_{ki}(n+1)$ are updated in Step 9, the new values

[2] See [18] for details and a proof of optimality.

Algorithm 1: DRVR;
protocol at node i

```
Data: Link mean rates $\widehat{R}_{ij}$ and $\widehat{R}_{ji}$ for $j \in n(i)$
Data: Link variances $\sigma_{ij}^2$ and $\sigma_{ji}^2$ for $j \in n(i)$
Result: Near optimal routing variables $T_{ij}(n)$ for $j \in n(i)$
1  while $S_{ki}(n) \geq \gamma$ do
2      while $\lambda_{kj}^{(i)}(n-1) = \lambda_{kj}^{(i)}(\tau)$ for all $j \in n(i)$ and $\tau \leq \delta$ do
3      Update neighboring multipliers $\lambda_{kj}^{(i)}(n) = \lambda_{kj}^{(i)}(\tau)$
4      Update primal variables $T_{kij}$ as per (13)
           $$T_{kij}(n) = \left[\frac{1}{\sigma_{ij}^2}\left(-\delta_i(n) + \lambda_{ki}(n)\widehat{R}_{ij} - \lambda_{kj}^{(i)}(n)\widehat{R}_{ji}\right)\right]^+$$
5      Transmit primal variable $T_{kij}(n)$ to node $j \in n(i)$
6      while $T_{kji}^{(i)}(n-1) = T_{kji}^{(i)}(\tau)$ for all $j \in n(i)$ and $\tau \leq 2\delta$ do
7      Update neighboring routing variables $T_{kji}^{(i)}(n) = T_{kji}^{(i)}(\tau)$
8      Compute constraint slacks as per (9)
           $$S_{ki}(n) = a_{0,ki} - \sum_{j \in n(i)} T_{kij}(n)\widehat{R}_{ij} + \sum_{j \in n(i)} T_{kji}^{(i)}(n)\widehat{R}_{ji}$$
9      Update dual parameters as per (14)
           $$\lambda_{ki}(n+1) = [\lambda_{ki}(n) + \epsilon_{ki} S_{ki}(n)]^+$$
10     Transmit $\lambda_{ki}(n+1)$ to all nodes $j \in n(i)$
11     Update discrete time $n = n + 1$. Reset analog time $\tau = 0$
12 end
```

Algorithm 2: Global stopping criteria for DRVR;
at node i

```
1  while $\tau \leq \delta'$ do
2      while $\lambda_{kj}^{(i)}(n) \neq \lambda_{kj}^{(i)}(\tau)$ for all $j \in n(i)$ do
3          Run DRVR;
4          at node $i$ (cf. Algorithm 1)
5      end
6      Reset analog time $\tau = 0$
7  end
```

are transmitted to *all* neighboring nodes $j \in n(i)$ as stated in Step 10. Reception of these new dual variables at neighboring nodes triggers recomputation of their primal variables. This recomputation is, again, beyond the scope of action of node i.

The counterpart of the transmission steps are the reception loops in steps 2 and 6. Here we use $\lambda_{kj}^{(i)}(n-1)$ to denote the local version of neighboring dual variables used in the last step and $\lambda_{kj}^{(i)}(\tau)$ to denote their current values. Step 2 locks execution of DRVR until an updated dual variable $\lambda_{kj}^{(i)}(\tau)$ is received from some neighbor $j \in n(i)$. When this happens we update the current value of the local multipliers $\lambda_{kj}^{(i)}(n)$ as indicated by Step 3 and proceed to the recomputation of primal variables in Step 4. Observe that the dual reception loop in Step 2 is unlocked by the transmission of a dual variable by some neighboring node currently running Step 10 of its local DRVR state machine. Step 6 locks execution at node i to wait for updated primal variables from some neighbor. When a new primal variable $T_{kji}^{(i)}(\tau)$ is received, its value is recorded into $T_{kji}^{(i)}(n)$ as per Step 7, and we proceed to the recomputation of dual variables. Observe that the loop in Step 6 is unlocked when some neighboring node updates the routing variable $T_{kji}^{(i)}(n)$ due to the execution of Step 5 of its local DRVR state machine.

2) **Protocol Convergence***:* Convergence is detected when the constraint slacks $S_{ki}(n)$ drop below a given tolerance γ. This is a suitable stopping criteria because constraint slacks are elements of the gradient of the dual function. The stopping criteria $S_{ki}(n) \leq \gamma$ is a local check on the norm of the gradient, which we know should vanish as the routing variable iterates $T_{kij}(n)$ approach optimal values T_{kij}^*. As DRVR nears convergence variable updates stop. To avoid software blocks in this situation, steps 2 and 6 include timers of duration δ. If δ time units elapse without a variable update, the corresponding wait loop is unlocked to proceed to the variable recomputations in steps 4 and 9.

There is a distinction between local convergence, in the sense of having local constraint slacks $S_{ki}(n) \leq \gamma$ smaller than the given γ tolerance, and global convergence in the sense of having $S_{ki}(n) \leq \gamma$ for all nodes i. It is only the latter that guarantees proximity of $T_{kij}(n)$ and T_{kij}^*, but only the former that can be checked locally. It is possible that variables stop changing in a certain neighborhood, making DRVR comply with the exit criteria in Step 1 at some particular nodes, while iterations continue in other places. Eventually however, changes that occur in some part of the network propagate to other parts, but this is moot if DRVR has declared convergence in these nodes. To avoid this problem we introduce a global stopping criteria for DRVR as shown in Algorithm 2. This forces the algorithm to wait for δ' time units before declaring convergence. If during this waiting time a change is detected (cf. Step 2 of Algorithm 2) the loop in Algorithm 1 is restarted. Note that the performance of the protocol is not sensitive to an exact optimal choice of δ' as long as the value is chosen to be sufficiently large.

Though [18] proves convergence of a synchronous version of DRVR, the proof does not extend to an asynchronous implementation. Nevertheless, empirical results show that even for relatively small δ values that allow asynchronous computations that are more likely to involve outdated values, DRVR converges to the same final results, albeit in a larger number of iterations. In particular, a series of experiments were performed on networks described in Section III-C for values of δ equal to $200{,}000, 20{,}000, 2{,}000, 200$ and 20ns. As δ decreased from $20{,}000$ns to 200ns, the average number of iterations required before DRVR converged increased from around 900 to around $6{,}000$, but the final results were identical. It is only in the last scenario with an unrealistically small value of $\delta = 20$ns that nodes claim local convergence before global convergence had occurred. This results in incorrect final values, and highlights that in practice a reasonable delay should be imposed before allowing nodes to start a new computation cycle. This ensures sufficient time to receive updates from enough neighbors.

Finally, note that although once computed the multipath routes of DRVR minimize the need to update routing after every network change, DRVR's computations will occasionally have to be restarted, *e.g.,* when adding or removing nodes. When those computations end, forwarding decisions must be updated, and updates should be performed without jeopardizing network stability. In particular, this calls for avoiding the formation of transient loops that could quickly exhaust link bandwidth. Fortunately, this can be readily realized through a simple adaptation of the DIV protocol of [11], which guarantees that loops never form. Due to space limitations, it is not possible to include the detailed description of this adaptation, but the key to its feasibility is that the local Lagrange multipliers of DRVR map directly onto the "values" that DIV relies on to avoid loop formation. Specifically, DIV allows forwarding from node i to node j only if the value (for a given destination) of node i is larger than that of node j. Similarly, DRVR's routing variables result in forwarding from node i to node j only if node i's multipliers are larger than those of node j. This allows a direct mapping of DIV's update rules onto DRVR's operation.

III. Evaluation Methodology

The previous section introduced DRVR that seeks predictable transmissions in mesh networks, where the availability and quality of wireless links vary faster than routing can adapt to.

The computational complexity of an exact solution together with the need for a distributed approach led to a formulation that replaced the end-to-end route selection problem by a concatenation of local decisions, where rate variance was minimized under the constraints of meeting average rate guarantees. Our goal is next to evaluate the extent to which this approach succeeds in minimizing end-to-end rate variations while securing end-to-end rate guarantees.

Given the intractability of an optimal benchmark, we resort to comparing DRVR to a heuristic that simply targets maximizing end-to-end path availability, and therefore the odds that rate guarantees are met. In other words, rate variations are not explicitly accounted for, and the focus is instead on selecting paths that are least likely to experience disruptions and, therefore, rate variations.

Next, we outline and justify the use of this "most reliable path" (MRP) heuristic as a benchmark, and identify the metrics we rely upon to compare MRP to DRVR. The scenarios used for the purpose of this comparison are also introduced together with parameters whose impact is explored.

A. Most reliable path

When links experience random quality fluctuations, a natural option is to favor paths whose end-to-end stability/reliability is maximal. Such paths experience the least disruptions, and can be readily computed in a distributed fashion using only local estimates of link reliability. For example, links can be represented as a Gilbert-Elliot channel [9] that alternates between "good" and "bad" states. Assuming that the traffic load is low (high) compared to link capacities when in a good (bad) state and that link "failures" are independent, a most reliable path is a shortest path with link weights set to minus the log of the probability that the link is in its good state. The distributed computation of MRP can, therefore, be readily accomplished using a standard distance-vector approach, *e.g.,* Bellman-Ford.

B. Evaluation metrics and parameters

Given our target of minimizing rate variations while offering rate guarantees, it is natural for the evaluation to focus on metrics that capture these goals. Two such metrics are the mean and the standard deviation of the transmission rates of individual flows under both DRVR and MRP. In particular, we are interested in the extent to which DRVR's lack of end-to-end path awareness affects its average rate performance. Conversely, MRP's focus on paths that experience the least disruptions independent of rate variability should yield average rates that outperform DRVR, but at the cost of greater variations.

1) Average rate and rate variance: A flow's transmission rate is measured at the flow's destination node and tracks both the different rates at which the destination receives data and the durations of the periods when each rate is in effect. More formally, for an experiment of duration T, the average rate α_{ki} and variance σ_{ki}^2 for a flow from node i to node k are given by

$$\bar{\alpha}_{ki}(T) = \frac{1}{T}\int_0^T \alpha_{ki}(t)dt \tag{15}$$

$$\bar{\sigma}_{ki}^2(T) = \frac{1}{T}\int_0^T \alpha_{ki}^2(t)dt - \alpha_{ki}^2\,, \tag{16}$$

where $\alpha_{ki}(t)$ denotes the end-to-end rate at which destination node k receives data from source node i at time t.

Comparing variances $\bar{\sigma}_{ki}^2(T)$ or standard deviations $\bar{\sigma}_{ki}(T)$ for different protocols may yield unfair metrics as it is possible to reduce the standard deviation of a random variable simply by reducing its mean. To avoid this, we rely on the normalized standard deviation $\tilde{\sigma}_{ki}(T)$ defined as

$$\tilde{\sigma}_{ki}(T) = \frac{\bar{\sigma}_{ki}(T)}{\bar{\alpha}_{ki}(T)}. \tag{17}$$

In Section IV, we report empirical values for mean rates $\bar{\alpha}_{ki}^{\text{DRVR}}(T)$ and $\bar{\alpha}_{ki}^{\text{MRP}}(T)$ of DRVR and MRP, as well as empirical estimates of normalized standard deviations $\tilde{\sigma}_{ki}^{\text{DRVR}}(T)$ and $\tilde{\sigma}_{ki}^{\text{MRP}}(T)$ for DRVR and MRP.

2) Buffer space, spare capacity, and delays: In most communication systems there is a trade-off between buffer space and transmission capacity; at least within the system's capacity region. Our system is no exception, and the introduction of increasingly large buffers can allow any routing to realize rate guarantees (below capacity) when averaged over a long enough period of time. Hence, when buffers are large enough and enough spare capacity exists to drain buffers that have filled up during periods of link unavailability, both DRVR and MRP should realize the same average rate performance. Buffers can, therefore, help DRVR mitigate its inability to account for end-to-end transmission rates. Assessing the impact of buffers on both a flow's realized average rate and its rate variance[3] is explored in Section IV.

Large buffers, however, come at a cost. In particular, they introduce additional delays while draining link down periods. These delays can become large and amplify the impact of disruptions when network load is high so that buffers drain slowly. The impact of these factors on DRVR's and MRP's ability to benefit from larger buffers is also assessed in Section IV, which further illustrates differences in the two protocols' sensitivity to network load. Because MRP selects paths solely based on their end-to-end reliability without accounting for their capacity, it

[3] As a flow's larger average rate is realized by allowing higher transmission rates while buffers are drained, this can increase the flow's rate variance.

tends to pack flows on the more reliable links. This can result in overloading those. In contrast, the rate constraints built into the optimization behind DRVR ensure that link overloads are avoided when feasible. On the flip side, MRP tends to favor shorter paths (fewer links) as they are usually more reliable. This consumes resources on fewer links, while DRVR multipath choices often result in longer, less efficient paths. Given our focus on minimizing rate variations while meeting (as opposed to optimizing) average rate guarantees, we focus on scenarios that avoid penalizing either protocols because of overloaded links, *i.e.,* we avoid high load configurations.

C. Topologies, link capacities, and simulation scenarios

To compare DRVR and MRP, we generate a number of representative network topologies using a method similar to that of [12]. Specifically, for a network consisting of N nodes, we distribute nodes uniformly at random in a square of unit dimension. Links are then sorted by distance and the first $\frac{Nd}{2}$ are kept, where d is the average node degree targeted for the network topology. The resulting network corresponds to a graph $\mathcal{G}(V, E)$ of $|V| = N$ vertices with average degree d, and $|E| = \frac{Nd}{2}$ edges that reflect a node's ability to communicate with neighbors at a distance less than a certain maximum value. The reliability of communications on a link is, however, based on the link's length. We assign values of P_b and $P_d < P_b$ to the shortest and longest links in the network, respectively[4], and as suggested in [2] take a link's reliability to decrease quadratically between these two values as a function of its length.

The next step in constructing a network topology is to assign realistic capacities to links. All links are initially assumed to have the same raw transmission rate C, but their actual capacity should reflect interactions arising from transmissions among nodes within transmission range of each other, and the resulting selection of a reasonable transmission schedule, *e.g.,* as would be discovered by the 802.11 MAC protocol. To capture these effects and realize a reasonable assignment of link capacities, we introduce the standard assumption that interferences prohibit any receiver from being in the transmission range of two or more transmitting nodes or from simultaneous transmission and reception at any given node. Writing these constraints explicitly schedules δ_{ij} are required to satisfy

$$\begin{aligned}
&\sum_{k=1}^{n} \delta_{jk} + \sum_{i=1}^{n} a_{ij}\delta_{ij} \leq 1, \\
&\sum_{i=1}^{n} a_{ij} \sum_{k=1}^{n} \delta_{ik} + \sum_{i=1}^{n} a_{ij}\delta_{ij} \leq 1 \qquad \delta_{ij} \in [0,1]. \qquad (18)
\end{aligned}$$

In (18) δ_{ij} represents the fraction of transmission opportunities in which the link $i \to j$ is active and the a_{ij} are elements of $\mathbf{A}$, the adjacency matrix of $\mathcal{G}(V, E)$. The corresponding average rate of the link $i \to j$ would be $\widehat{R}_{ij} = P_{ij}C\delta_{ij}$ where P_{ij} is the link reliability computed according to the description in the previous paragraph and C is the raw transmission rate. We then proceed to choose schedules δ_{ij} that maximize a proportional fair rate utility $U_{ij} = \log(P_{ij}C\delta_{ij})$, i.e.,

$$\begin{aligned}
\max_{\delta_{ij}} \quad & \sum_{i,j} U_{ij} = \log(P_{ij}C\delta_{ij}) \qquad (19) \\
\text{s.t.} \quad & \text{constraints in (18)}
\end{aligned}$$

The utility U_{ij} seeks to approximate a reasonably fair transmission schedule across nodes. We emphasize that our formulation of the maximum weight optimization in (19) is a dual relaxation of the actual integer program that needs to be solved to find optimal schedules. This relaxation has been observed to be a reasonable approximation to the NP-hard original problem.

Once network topologies and link capacities have been generated, we consider two different operating environments.

In the first, small buffer configuration, nodes can store just a few packets waiting to be transmitted, but those buffers are not sufficient to hold data while a link is unavailable. In such a setting, a flow realizes a non-zero rate only when *all* links on its path are operational. As discussed earlier, the lack of buffers to average out transient interruptions is expected to have a greater impact on DRVR, because of its lack of explicit awareness for end-to-end path reliability.

In the second, large buffer configuration, we introduce progressively larger buffers that can store data while transmissions are interrupted due to a link's unavailability. Again as mentioned earlier, this should improve the average rate performance of both DRVR and MRP, with both eventually (once buffers are large enough) yielding similar average rates. With large buffers, access to excess link capacity to drain a flow's buffer is in proportion to the flow's original rate requirement.

IV. Performance comparison of DRVR and MRP

As mentioned earlier, their respective designs should transalate in MRP outperforming DRVR in terms of expected rates, whereas the converse should hold when it comes to rate variances. In this section, we quantify these differences in performance, and show that DRVR offers an attractive compromise.

For that purpose, we rely on networks generated according to the method of Section III-C. Given our initial intent of using mesh networks as an alternative to traditional wired *infrastructures*, we focus on small topologies, *i.e.,* around 10 nodes, which correspond to common deployment sizes for the type of light infrastructure we target. We expect the benefits of DRVR to, however, remain or even increase in larger topologies, as its ability to construct multipaths that meet rate guarantees while minimizing variance improves with the greater path diversity commonly available in larger topologies. To facilitate comparisons (between DRVR and MRP), we also rely on configurations involving a small number of flows, *i.e.,* either 4 flows (2 random destinations each with 2 random sources) or 8 flows (2 random destinations each with 4 random sources). These correspond to light and medium load scenarios, respectively.

The nominal link capacities introduced in Section III-C are normalized to a unit capacity, while each individual flow contributes a traffic intensity of $1/25^{\text{th}}$ that value. Actual link capacities are set based on the solution of the optimization problem (19) that seeks to account for the impact of interferences, fading, collisions, etc. Similarly, link reliabilities are set

[4] In Section IV, we use $P_b = 0.95$ and $P_d = 0.7$.

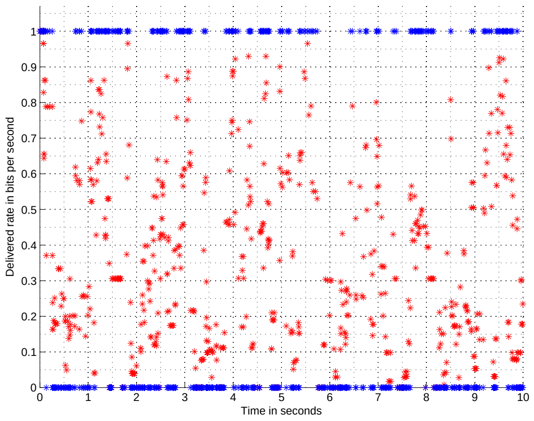

Fig. 1. "Instantaneous" realized end-to-end flow rates of DRVR (red dots) and MRP (blue stars) for a sample flow in a 4-flows configuration.

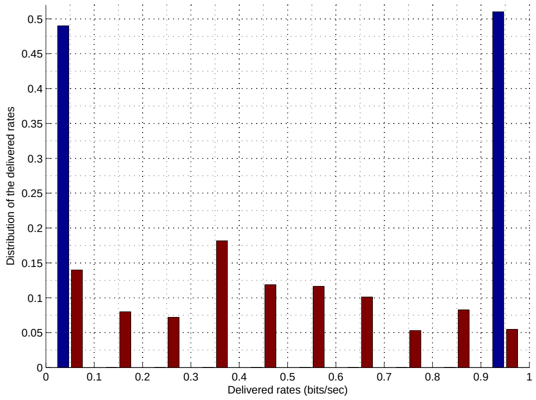

Fig. 2. Probability distribution of a sample flow's "instantaneous" rates $\alpha_{ki}(t)$ for DRVR (right red bars) and MRP (left blue bars) in a 4-flows configuration.

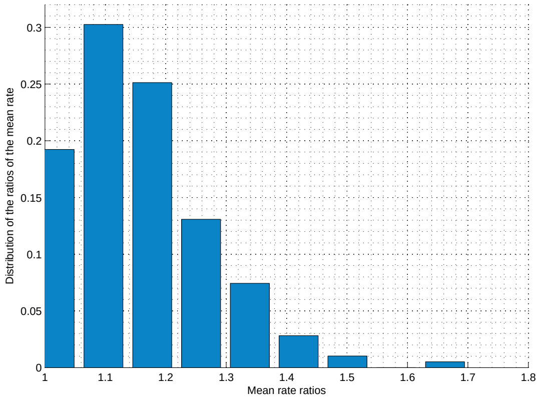

Fig. 3. Distribution of the ratio $\bar{\alpha}_{ki}^{\text{MRP}}(T)/\bar{\alpha}_{ki}^{\text{DRVR}}(T)$ for experiments of duration 3,000 secs across 1,000 different topologies (low load configuration).

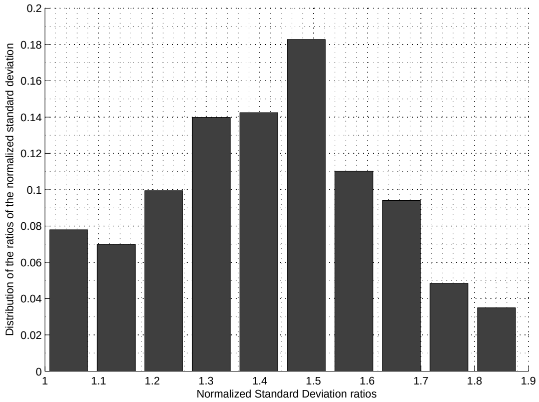

Fig. 4. Distribution of the ratio $\tilde{\sigma}_{ki}^{\text{MRP}}(T)/\tilde{\sigma}_{ki}^{\text{DRVR}}(T)$ for experiments of duration 3,000 secs across 1,000 different topologies (low load configuration).

as discussed in Section III-C, with exponentially distributed up and down times. The expected time between failures of link $i \to j$ is equal to ξP_{ij} sec, and conversely the expected duration of failures of link $i \to j$ is $\xi(1 - P_{ij})$ sec. The values P_{ij} are assigned as described in Section III-C, while ξ is a constant selected so that the average channel coherence time is 122msec. This corresponds to a pedestrian 802.11 channel operating in the 2.4 GHz band. The average rates and variances used as inputs to Algorithm 1 are also computed as per Section III-C.

Next, we discuss results obtained for the small and large buffer configurations mentioned earlier.

A. Small buffer scenarios

In the case of small buffers, flows realize non-zero end-to-end rates only when there is an uninterrupted path from source to destination. In the case of MRP that typically selects a single (most reliable) path, a flow realizes its full target rate when all links in the path are up, and a rate of 0 when at least one link in the path has failed (faded). For DRVR, a flow's realized rate is the sum of rates achieved across all the (multiple) paths currently open between source and destination.

1) Instantaneous rates: Fig. 1 offers an illustrative example. It plots the "instantaneous" end-to-end rates realized by a sample flow over a period of 10 secs. Rates are recorded at random sample times and the figure displays the recorded values for DRVR (red dots) and MRP (blue stars). The figure highlights MRP's bimodal rate behavior and the rate fluctuations of smaller amplitude displayed by DRVR. This difference is further highlighted in Fig. 2 that displays a histogram of normalized (to a flow's maximum rate) realized end-to-end flow rates for MRP and DRVR over a total experiment duration of now $T = 1,000$sec. The histogram consists of 10 bins, with the $x-$axis of Fig. 2 identifying bin boundaries, *e.g.,* the first bin tracks the number of normalized rate samples in the range $[0, 0.1]$. MRP rate samples alternate more or less equally between values of either 0 or 1, *i.e.,* in the first and last bins, while DRVR boasts a wider support for its rate samples as suggested in Fig. 1. Specifically, under MRP the flow suffers a rate of 0 close to 50% of the time, while the bulk of DRVR's probability mass is in the range $\alpha_{ki}(t) \in [0.3, 1]$. We also note that for the sample flow of Fig. 1, both MRP and DRVR realize similar average end-to-end rates (around 0.5). This is, however, somewhat atypical, as MRP's focus on selecting paths that experience the least disruption usually results in a higher average rate than DRVR, but at the cost of a correspondingly higher (normalized) variance. We explore this issue next.

Specifically, we carry out a set of experiments across a total of $1,000$ different topologies under both light (4 flows) and

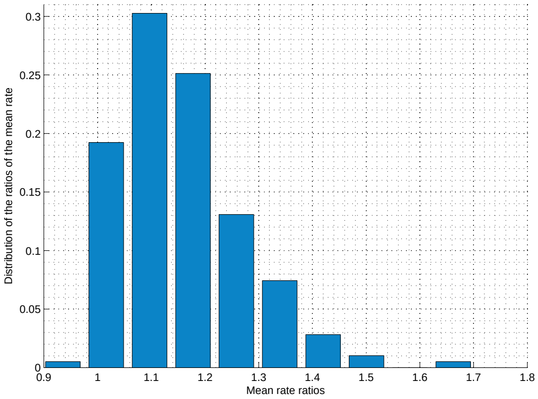

Fig. 5. Distribution of the ratio $\bar{\alpha}_{ki}^{\text{MRP}}(T)/\bar{\alpha}_{ki}^{\text{DRVR}}(T)$ for experiments of duration 3,000 secs across 1,000 different topologies (medium load configuration).

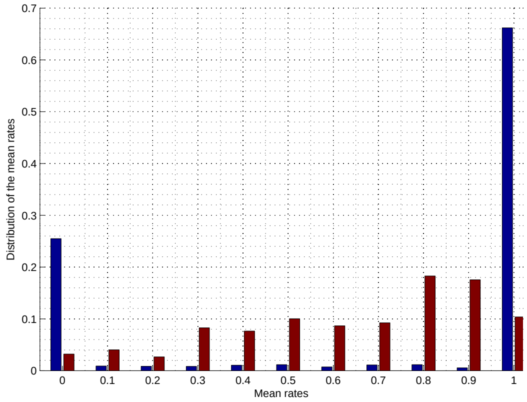

Fig. 7. Distribution of DRVR (right red) and MRP (left blue) rates averaged using a $W = 200$ msec sliding window.

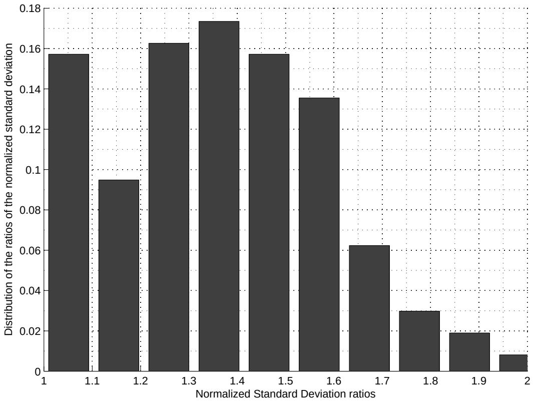

Fig. 6. Distribution of the ratio $\tilde{\sigma}_{ki}^{\text{MRP}}(T)/\tilde{\sigma}_{ki}^{\text{DRVR}}(T)$ for experiments of duration 3,000 secs across 1,000 different topologies (medium load configuration).

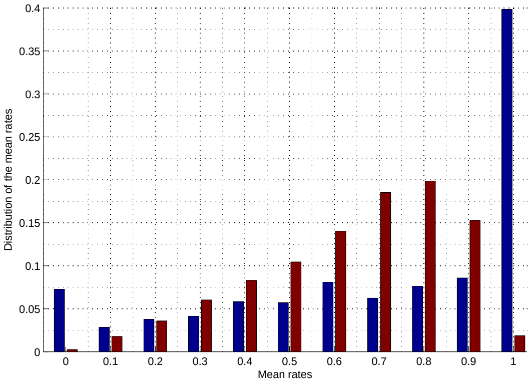

Fig. 8. Distribution of DRVR (right red) and MRP (left blue) rates averaged using a $W = 2$ sec sliding window.

medium (8 flows) loads. Each experiment runs for a total of $T = 3,000$ secs, at the end of which the mean and normalized standard deviation of each flow's realized rate is recorded for both DRVR and MRP, *i.e.,* $\bar{\alpha}_{ki}^{\text{DRVR}}(T)$ and $\bar{\alpha}_{ki}^{\text{MRP}}(T)$, as well as $\tilde{\sigma}_{ki}^{\text{DRVR}}(T)$ and $\tilde{\sigma}_{ki}^{\text{MRP}}(T)$ as given by (15)-(17). The distributions of the ratios (MRP/DRVR) of both quantities are reported in Figs. 3 and 4 for a 4-flow configuration, and in Figs. 5 and 6 for an 8-flow configuration. The findings from both sets of figures are comparable. They illustrate the (slight) penalty that DRVR pays compared to MRP when it comes to average realized rate, and at the same time demonstrate its advantage in terms of lowering rate variations.

The data from Figs. 3 and 5 gives median rate reductions of 0.81 and 0.87, respectively, for DRVR over MRP. In other words, a penalty of less than 20% in both cases. Similarly, the probability of a rate reduction exceeding 30% under DRVR is less than 15% in the low load configuration of Fig. 3, and even lower at barely 3% in the medium load configuration of Fig. 5. The slightly lower average rate penalty that DRVR experiences at medium load is in part due to the impact of load on MRP's ability to route all flows on their most reliable path. In particular, as load increases the most reliable links tend to become over-subscribed, which then forces MRP to select the next most reliable path for the affected flows. In contrast, because DRVR distributes flows across multiple paths, it is less likely to be forced to use a less desirable path because of rate feasibility constraints. The effect of load notwithstanding, DRVR does offer, as expected, meaningful improvements in rate variability. Figs. 4 and 6 show median improvements of about 43% and 36%, respectively, in the normalized standard deviation of flows' rates. The extent to which the reduction in rate variability that DRVR affords is worth the slight loss in average rate is likely to vary across applications. Investigating this for several representative applications is a topic we are actively investigating. However, this trade-off is in general likely to depend on both the magnitude and the time-scale of the reductions in rate variations that DRVR produces. This is a topic we explore next.

2) Rates and rate variations across time-scales: Because most applications are sensitive to rate variations over a specific time-scale, we introduce a rate averaging window to compare DRVR and MRP across different time-scales. The averaging window measures a flow's rate over a time interval of duration W. We select two sample values for W, namely, $W = 200$ msec and $W = 2$ secs. These correspond roughly to time-scales of relevance to interactive, real-time applications, *e.g.,* audio or videoconf, and interactive, non-real-time applications, *e.g.,* web browsing. For both values of W, we compute average rate values using a sliding window of duration W and using a sliding step

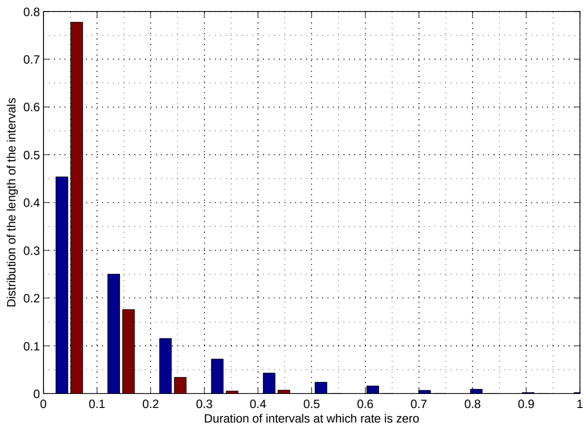

Fig. 9. Distribution of zero-rate durations in secs for DRVR (right red) and MRP (left blue) over a sample $3,000$ secs experiment.

of size $W/20$.

Comparisons between DRVR and MRP for representative flows are presented in Figs. 7 and 8 for $W = 200$msec and $W = 2$sec, respectively. In both cases, the network operates at light load and links coherence time is kept[5] at 122msec. Fig. 7 shows that at a time-scale of 200msec, MRP subjects flows to significant rate variations, while DRVR offers more even performance. In particular, the odds that a flow receives less than 30% of its nominal rate, an arguably severely degraded level of performance, is less than 16% under DRVR and about 28% under MRP; a non-trivial difference. As illustrated in Fig. 8, this difference, shrinks when the time-scale increases to 2secs,. In this case, the odds that the flow receives less than 30% of its nominal rate is 7% under DRVR and 15% under MRP. When looking at the converse, namely the odds that the flow experiences a rate of 90% or more of its nominal rate, MRP has an edge (44% vs. only 8% for DRVR). Qualitatively similar results were observed across different topologies and for different network loads.

Figs. 7 and 8 illustrate what may be the most important advantage of DRVR over MRP. User perception is typically governed by average communication performance during application-specific time scales. For infinite time horizons the metrics for DRVR and MRP coalesce to their respective (long-term) means. As the mean of MRP is larger than that of DRVR , the former is preferred in this scenario. However, the smaller variance in instantaneous rates $\alpha_{ki}(t)$ implies that the convergence of DRVR to its mean performance is faster than the convergence of MRP. Thus, for smaller time windows, DRVR should offer better user experience due to the reduced rate variability. Again, assessing to what extent this is indeed realized for different applications, is a topic of ongoing work.

A related perspective is presented in Fig. 9. It again considers a given (random) topology, and gives the distribution of the duration of zero rate periods, *i.e.,* time intervals during which $\alpha_{ki}(t) = 0$, under DRVR (right red bars) and MRP (left blue bars). The figure illustrates the benefits of DRVR in ensuring much shorter periods of complete transmission interruption. The average length of service interruption is 47msec for DRVR and interruption periods longer than 300msec have essentially zero probability. For MRP the average length of service interruption is 139msec, and service interruption periods longer than 300msec occur with probability 0.19. Repeated experiments of duration $T = 3,000$secs yielded an average time of total communication interruption of 14secs for DRVR vs. 88secs for MRP.

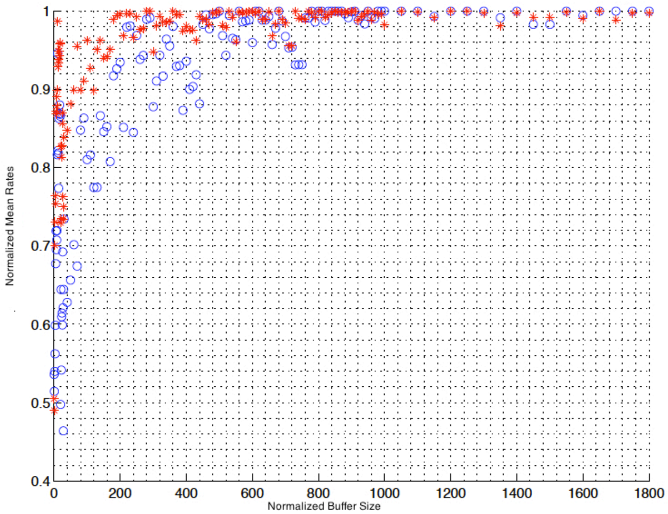

Fig. 10. Normalized flow rates as a function of normalized buffer size for DRVR (red stars) and MRP (blue circles).

B. Large buffer scenarios

The previous section offered evidence in support of the benefits that DRVR offers when it comes to reducing rate variability, especially at short time-scales. These benefits were at the cost of lower average rates than a solution such as MRP. However, the decrease was observed under the assumption of a network configured with small buffers, *i.e.,* with insufficient storage to hold data during link down times and transmit it once the link comes back up. This is not unreasonable since large buffers can introduce substantial delays given standard link speeds in wireless mesh networks. Nevertheless, it is an assumption that penalizes DRVR and it is, therefore, of interest to explore if/how larger buffers may improve its performance.

The benefits of buffers (for either DRVR or MRP) depend on both the size of the buffers and the spare capacity available in the network to (quickly) drain data that accumulated during link down times. Fig. 10 reports on the result of an experiment carried out at low load for different buffer sizes. The $x-$axis is the buffer capacity in units normalized to a flow's original transmission rate, *i.e.,* if flows transmit at a rate of 1Mbps, buffers are in units of Megabits. The $y-$axis reports normalized mean rates. A normalized rate of 1 would indicate that the buffers have been successful at entirely eliminating the impact (on the average rate) of link disruptions. The results (red stars and blue circles correspond to normalized rates of an individual flow during a sample experiment under DRVR and MRP, respectively) illustrate that as expected, increasing buffer size eventually helps both DRVR and

[5] Changing the link coherence time simply amounts to rescaling the results, *e.g.,* a decrease by a factor 10 in link coherence time transforms Fig. 7 into Fig. 8, and conversely an increase by a factor 10 transforms Fig. 8 into Fig. 7.

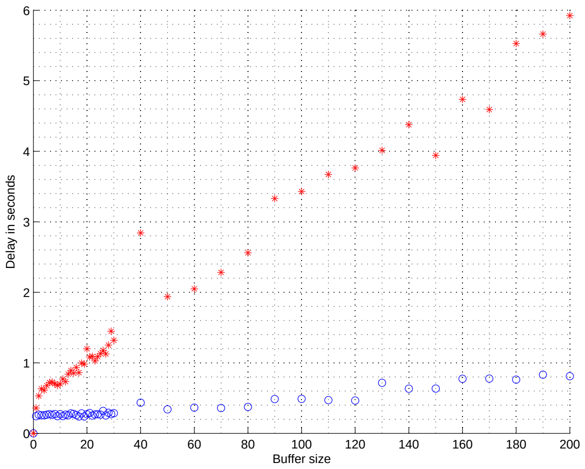

Fig. 11. Average flow delays as a function of normalized buffer size for DRVR (red stars) and MRP (blue circles).

MRP[6] realize average rates close to the their nominal rate. This means that DRVR's disadvantage vanishes, and it can be verified that its advantage when it comes to reducing rate variability remains[7]. This being said, the buffer sizes required to realize average rates close to a flow's nominal rate for both DRVR and MRP are large, *i.e.,* a few 100 Megabits assuming 1Mbps flows. Hence, such values are unlikely to be practical, especially since as presented next, they translate into substantial delays.

Fig. 11 reports average delays experienced by flows in a sample topology during a 1,000secs experiment under low load conditions, as a function of normalized buffer sizes for DRVR (red dots) and MRP (blue circles). For DRVR, the delay is the worst average delay across all of a flow's paths. This reflects that packets delayed on one path affect the delays of packets on all other paths, and is in part responsible for the steeper growth of DRVR's delay with buffers. This, however, ignores the possibility that excessively delayed data may be retransmitted on other paths and arrive before the original copy, or that coding be used [3], [14], [15] and allow the recovery of delayed data. Hence, the data of Fig. 11 represents a worst case scenario for DRVR. The combination of Figs. 10 and 11 nevertheless illustrates that increasing buffers to eliminate the average rate penalty caused by link rate fluctuations, translates into delays that for both DRVR and MRP quickly become comparable to the response time of reactive solutions, *e.g.,* re-routing. Avoiding such delayed reactions was one of the motivations behind DRVR, which confirms that its preferred use is in a small buffer setting.

V. Conclusion

The paper introduced DRVR, a routing protocol that seeks to deliver rate guarantees while minimizing rate variability in wireless mesh networks. The protocol was evaluated and compared to a solution, MRP, that focuses on maximizing long-term throughput, independent of short-term rate variations. The evaluation demonstrated the benefits of DRVR in delivering reasonably stable rates over short time-scales.

There are two major directions in which the work is being extended. The first seeks to better quantify the benefits of DRVR for actual applications. Of particular interest is the evaluation of TCP performance to better understand the extent to which it can benefit from reduced short-term rate variability. A second equally important extension involves an actual experimental evaluation. This requires not only a complete implementation of DRVR on a wireless mesh network platform, *e.g.,* OpenWrt (see https://openwrt.org.), but also the design and implementation of the link estimation procedures needed to acquire the necessary link statistics information. Once completed, this work should offer a complete and more accurate assessment of DRVR's potential as a protocol for wireless mesh networks.

References

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. *ACM SIGCOMM Computer Communication Review*, 34(4):121–132, 2004.

[2] A. Aguiar and J. Gross. Wireless channel models. *Telecom. Netw. Group, Technische Universität Berlin, Tech. Rep. TKN-03-007*, 2003.

[3] J. Apostolopoulos. Reliable video communication over lossy packet networks using multiple state encoding and path diversity. In *Proc. SPIE Conf. Visual Comm. & Image Proc.*, Los Angeles, CA, January 2001.

[4] E. Ayanoglu, I. Chih-Lin, R. Gitlin, and J. Mazo. Diversity coding for transparent self-healing and fault-tolerant communication networks. *Communications, IEEE Transactions on*, 41(11):1677–1686, 1993.

[5] S. Barre and J. Iyengar. Architectural guidelines for multipath TCP development. *draft-ietf-mptcp-architecture-05*, 2011.

[6] S. Barré, C. Paasch, and O. Bonaventure. Multipath TCP: from theory to practice. *NETWORKING 2011*, pages 444–457, 2011.

[7] R. Draves, J. Padhye, and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. *ACM SIGCOMM Computer Communication Review*, 34(4):133–144, 2004.

[8] T. Dreibholz, H. Adhari, M. Becke, and E. Rathgeb. Simulation and experimental evaluation of multipath congestion control strategies. In *Proceedings of the 2nd International Workshop on Protocols and Applications with Multi-Homing Support (PAMS), Fukuoka/Japan*, 2012.

[9] E. Gilbert. Capacity of a burst-noise channel. *Bell System Technical Journal*, 39(5), September 1960.

[10] K. W. Kwong, L. Gao, R. Guérin, and Z.-L. Zhang. On the feasibility and efficacy of protection routing in IP networks. *IEEE/ACM Trans. Netw.*, 19(5):1543–1556, October 2011.

[11] S. Ray, R. Guérin, K. Kwong, and R. Sofia. Always acyclic distributed path computation. *IEEE/ACM Trans. Netw.*, 18(1):307–319, February 2010.

[12] I. Stojmenovic and X. Lin. Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(10), 2001.

[13] J. Tsai and T. Moors. A review of multipath routing protocols: From wireless ad hoc to mesh networks. In *Proc. ACoRN Early Career Researcher Workshop on Wireless Multihop Networking*, July 2006.

[14] A. Tsirigos and Z. Haas. Analysis of multipath routing - Part I: The effect on the packet delivery ratio. *IEEE Trans. Wireless Comm.*, 3(1), January 2004.

[15] E. Vergetis, E. Pierce, M. Blanco, and R. Guérin. Packet-level diversity: From theory to practice. an 802.11-based experimental investigation. In *Proc. MOBICOM 2006*, Los Angeles, CA, September 2006.

[16] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz. Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer. *IEEE Trans. Industrial Electronics*, 49(6), 2002.

[17] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath TCP. *Proc. Usenix NSDI 2011*, 2011.

[18] Y. Wu, A. Ribeiro, and G. Giannakis. Robust routing in wireless multihop networks. In *Information Sciences and Systems, 2007. CISS'07. 41st Annual Conference on*, pages 637–642. IEEE, 2007.

[6] The fact that MRP requires larger buffers is in part because its single path solution translates into higher buffer filling rates during link down times.

[7] A companion figure of Fig. 10 showing DRVR's benefits when it comes to the standard deviation of flows' rates is not shown due to lack of space, and neither are figures showing similar outcomes at medium and high load configurations.