# Routing in a Highly Dynamic Topology

Yashar Ganjali, Nick McKeown
Computer Systems Laboratory, Stanford University
Stanford, CA 94305-9030
{yganjali, nickm}@stanford.edu

***Abstract*— Routing in mobile ad-hoc networks is hard because the topology can change very rapidly. By the time new paths are discovered, the network can change again – and in extreme cases, packets circulate endlessly and the system is unstable. Most attempts to solve this problem have required that the topology changes slowly. In this paper, we propose a routing algorithm called *Volcano Routing Scheme* (VRS) which will route packets successfully, even if the topology changes very rapidly. VRS doesn't need to discover routes, or exchange routing information; it simply balances the load locally between adjacent pairs of nodes. We show that under some loose conditions on network topology changes, VRS keeps the system stable. Simulations also suggest that VRS is stable for various models of mobility, different communication patterns, and different amounts of flow in the network. Interestingly, we prove that when the network topology is static, packets follow the shortest path.**

## I. Introduction

Traditionally, routing in a data network is considered to be a two-phase process. In the *route-discovery phase*, a route is established between all source-destination pairs. Then, in the *packet forwarding phase*, packets are forwarded along the chosen path toward their destination [1]. Route-discovery can be done periodically (*proactive routing*), or initiated on-demand (*reactive or on-demand routing*), that is, whenever a source node needs to send a packet toward a destination (Figure 1).

When the topology of the network is fixed or changing slowly, the path established between a source and destination node can be used by any packet going from the same source to the same destination. Therefore, even though the overall overhead of establishing the routing path is relatively high, the cost is paid only once.

In networks for which the topology changes frequently, the overhead of finding a route must be paid often, which can dramatically decrease the performance of the system. In an extreme case, one can imagine a network in which the topology changes (perhaps multiple times) before a route is found, making two-phase routing impractical. Delivering packets to their destinations seems hard, if not impossible, in such a dynamic network. This is a limitation of two-phase routing, *i.e.*, packet forwarding can be done only when the route-discovery phase is complete.

It is expected that some ad-hoc networks will be highly dynamic, with the topology changing quickly. For example, in a network of tiny environmental sensors (*e.g.* Smart Dust [2]) nodes might be floating in the air or in water, and are subject to rapid movements, and frequent changes in the topology.

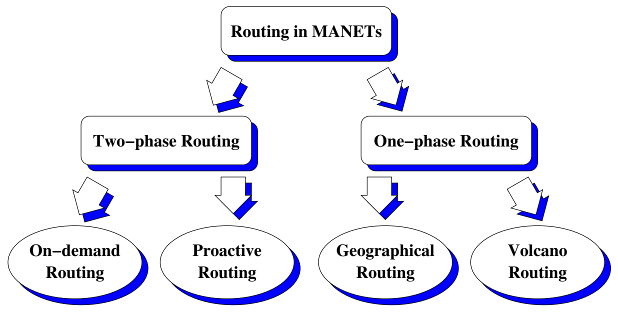


Fig. 1. Routing in mobile ad hoc networks.

One natural approach is to accelerate the route discovery phase [3]. For instance, on-demand routing protocols like AODV [4], TORA [5], and DSR [6], [7], use flooding to find routes as quickly as possible. But these protocols still use two phases, which means they can fail to route packets if the network changes too quickly.

An alternative approach, called *one-phase* routing, is to eliminate the explicit discovery of routes. Potentially, *one-phase* routing algorithms can find a path in a rapidly changing network. The most well-known one-phase routing schemes are *geographical routing* schemes. The idea is to use the location of a node as its address and forward the packets greedily toward the destination [8]. Although nodes move, routing is done toward a specific location rather than a specific node, and so there is no need for a separate route discovery phase. This property can be taken advantage of, whenever nodes of the network have access to some type of global positioning information.

In this paper, we propose an alternative *one-phase* routing algorithm, called the *Volcano Routing Scheme* (VRS), which does not require geographic information. VRS is a simple and fast method for routing packets in a highly dynamic network. VRS routes messages by locally balancing the load between adjacent pairs of nodes. It has no explicit route discovery phase, and can tolerate rapid changes in the network topology. Below, we show that under some loose conditions on how fast the topology changes, VRS keeps the system stable. We especially prove that the length of the path taken by packets under VRS is near optimal (*i.e.* shortest path), when the network topology is fixed. We also verify the stability of the system (for various mobility processes, different values of communication range, and different amount of flow in the network) using simulations.

The main advantages of VRS are as follows:

- VRS is completely distributed.
- The overhead of running VRS is low compared to other routing algorithms. The running time of the algorithm at each node $v$ is $O(KL_v)$, where, $L_v$ is the number of links connected to $v$ and $K$ is the number of flows in the networks.
- The amount of control traffic exchange between nodes is low.
- No matter how often the topology of the network changes, and no matter what the movement model of the nodes is, as long as some loose constraints on the topology are satisfied, VRS guarantees to deliver packets to their destinations.
- If the topology of the network is fixed (*i.e* when the nodes do not move), the length of the path taken by each packet is close to the shortest path from the source to the destination.

On the other hand:

- VRS performs best in a networks in which a continuous stream of packets are generated at each source node. If the source node generates packets sporadically, VRS does not perform well.
- Some packets might remain in the network for a long time before reaching their destination.[1] Therefore, packets might be re-ordered, which makes VRS unsuitable for applications that cannot tolerate packet re-ordering.

The rest of this paper is organized as follows. In Section II we briefly overview the related work. Section III explains the underlying network and traffic model. Then, we give a general description of how Volcano Routing works in Section IV. Sections IV-A, and IV-B describe VRS in more detail together with an analysis of the stability of the system. In Section V we show that packets take near optimal paths under VRS. Section VI is dedicated to simulations, which study the performance of VRS. Finally, Section VII concludes the paper.

## II. Related Work

Mobile ad-hoc networks (MANETs) have dynamic topologies. In MANETs, the topology changes when nodes move. In recent years, researchers have proposed numerous routing algorithms for MANETs. If the topology changes often, table-driven proactive routing protocols, that require periodic exchange of updates regarding the topology of the network, are unsuitable [9], [10]. On-demand routing protocols are more efficient since they only maintain the routes which are currently needed [5], [11], [12]. A route discovery process is initiated only when a new route is needed for transferring a packet. This approach was first presented by Johnson in 1994 [13] followed by a series of refinements, including AODV [4] and DSR [6]. Hybrid routing protocols combine on-demand and proactive routing to allow a trade-off between latency and overhead [14].

An alternative approach is to use geographic routing in which the location of a node is its address. Packets are forwarded greedily towards their destination [8], [15], [16]. There is no explicit discovery of routes, and they are more suitable for dynamic networks. The major drawback of geographical routing schemes is the need for a global positioning system, which is not always feasible.

All these techniques can cope with changes in the topology to some extent. However, their performance can be seriously degraded in the presence of extreme changes in the topology. Recently, Awerbuch *et al.* presented a simple routing scheme for an adversarial system [17], *i.e.*, a system in which an adversary can decide which links go down and which links remain up at each point of time. In the worst case, we can consider a highly dynamic network as an adversarial system. Therefore, Awerbuch's scheme is a perfect routing strategy for such a network. The major building block of this scheme is local load-balancing, introduced by Awerbuch, Mansour, and Shavit [18], and further studied and improved by Awerbuch and Leighton [19].

Awerbuch's method guarantees an upper bound on the number of packets that can be in the system at any time, even if the injection process is exactly at the limit of what the network can handle. However, this scheme only works in a network with a single receiver (destination). In this paper, we propose VRS, a generalization of Awerbuch's scheme, which can handle different flows destined to different destinations.

## III. Network and Traffic Model

Consider a network of $n$ nodes, with a unique identifier from the set $\{1, 2, \dots, n\}$ assigned to each node. The connectivity of the network can change over time, and is represented by the undirected graph $G = (V, E(t))$, where $V$ is the set of nodes, and $E(t)$ is the set of links at time $t$. Topology changes can occur as a result of node movements and variations in channel strength due to loss via distance attenuation, shading by obstacles, or interference from other nodes. In this paper, we are solely interested in topology variations regardless of the underlying cause. Despite changes in topology, each node is assumed to have a complete view of its adjacent neighbors, *i.e.*, at any time $t$ any given node $v$ knows any node $w$ such that $(v, w) \in E(t)$.

We assume that time is divided into discrete time slots, all nodes start simultaneously, and work synchronously. In any time slot, each link can transfer one packet. Each packet belongs to one of the $K$ flows in the network. The $i$-th flow ($1 \leq i \leq K$) is identified by a tuple $(S_i, D_i, F_i)$, where $S_i$ and $D_i$ are respectively the source and destination of the flow, and $F_i$ is the maximum number of packets which flow $i$ can generate in any time slot. We assume that all source and destination nodes are distinct. This is not a restrictive assumption since any node which is the source or destination of more than one flow can be decomposed into a number of virtual nodes. Packets are generated in the source nodes at the

[1]The number of such packets is very small, and by adding a time to live field we can simply remove old packets from the system and retransmit them if needed.

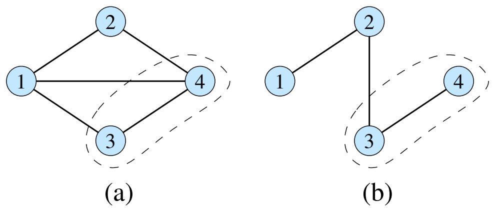


Fig. 2. An example of a cut.

beginning of each time slot. At the end of each time slot, any packet that has arrived to its destination is removed from the system. The following definitions will be used throughout the paper.

**Definition 1:** In a network whose node set is $V$, a *cut* $C = (U, V-U)$ is a partitioning of the set of nodes in the network into two sets $U$ and $V-U$. The cut $C$ is said to *separate* two nodes $v$ and $w$, if $v \in U$ and $w \in V-U$ or vice versa.

**Definition 2:** The *minimum capacity* of a cut $C = (U, V-U)$, denoted by $Min\_Cap(C)$, is the minimum number of links connecting $U$ and $V-U$ at any time. Similarly, the *average capacity* of $C$, denoted by $Avg\_Cap(C)$ is the average over time of the number of links connecting $U$ and $V-U$.

Figure 2 depicts a network with four nodes whose connectivity changes between parts (a) and (b) of the figure. In this example, $C = (\{1,2\},\{3,4\})$ is a cut which separates the nodes 1 and 4, $Min\_Cap(C) = 1$, and $Avg\_Cap(C) = 2$.

**Definition 3:** A node $v$ in the network is said to be *strictly stable* if the number of packets buffered in the node is always finite. Similarly, if the rate of flows entering the node equals the rate of outgoing flows, the node is said to be *rate-stable*. We will call a network strictly stable (rate-stable), if all nodes in the network are strictly stable (rate-stable).

We notice that in a rate-stable network, it is possible for the number of packets buffered in some node to become arbitrarily large, but if a network is strictly stable, the rate of flows entering each node must be equal to the rate of flows leaving the node.

**Observation 1:** Any network which is strictly stable, is rate-stable too; the reverse is not necessarily true.

**Definition 4:** We consider a positive real number $F$. Any network with a single flow with source node $S$ and destination node $D$ is said to be *$F$-min-provisioned* if for any cut separating $S$ and $D$

$$Min\_Cap(C) \geq F. \tag{1}$$

Similarly, if the average capacity of any cut $C$ separating $S$ and $D$ is at least $F$, *i.e.*

$$Avg\_Cap(C) \geq F, \tag{2}$$

the network is said to be *$F$-average-provisioned*.

**Definition 5:** We consider a network together with $K$ flows. For the real vector $\overline{F} = (F_1, \ldots, F_K)$ the network is said to be *$\overline{F}$-rate-provisioned*, if for any cut $C$ we have:

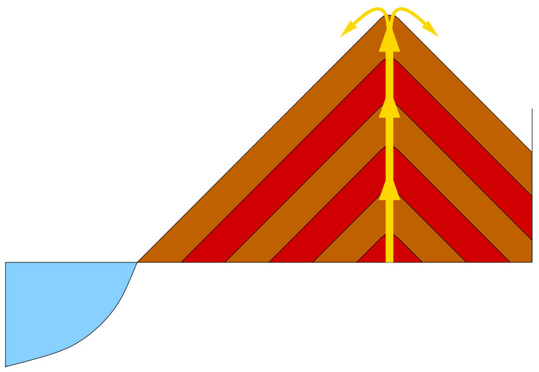


Fig. 3. The lava issued from a volcano flows in every direction till it hits a barrier, or reaches the sea.

$$Avg\_Cap(C) \geq \sum_{i \in \mathrm{Sep}(C)} F_i. \tag{3}$$

Here, Sep$(C)$ is the set of flows whose source and destination nodes are separated by $C$.

In the network depicted in Figure 2 for instance, if there is a single flow originating at node 1 and destined to node 3, then the network is 1-min provisioned, and 2-average-provisioned. If in addition to this flow, we have a second flow originating at node 2 and destined to node 4, then the network is $(1,1)$-average-provisioned.

## IV. Volcano Routing Scheme

The Volcano Routing Scheme (VRS) is a *potential-based* routing scheme [20]. The key idea in potential-based routing is to define $K$ scalar fields on the network, one for each destination node. More formally, a single-valued potential, denoted by $P_v^i$, is associated with any flow $i$ at a given node $v$. At each node $v$ of the network, packets destined to $D_i$, the destination of the $i$-th flow, are routed in the direction (*i.e.* the next hop) that the potential field decreases most for flow $i$.

Normally, the potential function depends on the topology of the network, and is chosen so that each packet is directed toward its destination. In VRS, the potential function is quite different, and is simply based on the number of packets buffered at each node of the network, and not on the connectivity of the network. At a given node $v$ and for a given flow $i$, the potential function $P_v^i$ equals the number of packets of flow $i$ that reside at node $v$. VRS forwards packets from nodes which have more buffered packets, to nodes which have fewer buffered packets, *i.e.*, VRS locally balances the load between adjacent nodes. If the load-balancing works perfectly, any destination node will have a chance to receive packets which are destined to it. Since we have assumed that packets are continuously injected into the network, the destination will receive a continuous stream of packets at all times.

Intuitively, VRS mimics the behavior of the lava flowing from a volcano and downhill towards the sea. When it encounters an obstruction (i.e. congestion), it builds up until it can overcome the obstacle (i.e. has greater potential); see Figure 3. Notice that even if the volcano and obstacles move, so long as the flow continues it will eventually overwhelm any obstacle and flow to the sea. This is the case even though neither the volcano nor the lava know a path to take; the lava simply follows a path of decreasing potential. Likewise, packets in VRS will reach their destination, regardless of how nodes move, so long as new packets continue to flow into the network. Obviously, when congestion builds up, the number of packets needed to keep flows moving can be quite large. But in the following sections, we show that the total number of packets in the system remains bounded. In other words, VRS can deliver all but a fixed number of packets to their destination.

### A. *Single-Flow Networks*

Single-flow networks are those with only one flow $(S, D, F)$, where $S$ is the source node, $D$ is the destination node, and $F$ is the number of packets generated at the source node during each time slot. In this section, we will describe how VRS works in single-flow networks and prove its stability. This prepares us for the more general multi-flow case that follows.

**Single-Flow VRS (SF-VRS).** We define the potential function associated with node $v$ at time $t$, denoted by $P_v(t)$, as the number of packets buffered at node $v$ at the beginning of time slot $t$. During any time slot $t$, the following steps are performed in order.

i) At the beginning of the time slot, $F$ packets are generated by the source node.
ii) Each link $(v, w)$ for which
$$P_v(t) - P_w(t) \geq \Delta, \tag{4}$$
is marked *active*. Here, $\Delta \geq 0$ is a pre-specified parameter which is called the *transfer threshold*. All other links are marked *inactive*.
iii) If the number of packets residing at node $v$, which is $P_v(t)$, is greater than or equal to the number of active outgoing links adjacent to $v$ then each such link transfers one packet from $v$ to its neighbors. Otherwise, node $v$ randomly chooses $P_v(t)$ of its active outgoing links, and each of these links transfer one packet from $v$ to a neighboring node.
iv) Any packet which has reached the destination node $D$ is immediately removed from the system.

Step (ii) is run in parallel on all links of the network, and step (iii) is run in parallel on all nodes and active links of the network. Note that the potential function $P_v(t)$ is updated instantaneously after a packet leaves node $v$ or if a packet arrives at $v$.

Figure 4 shows a simple example of how SF-VRS works in a network with a fixed topology. In this example, node 1 and 3 are the source and destination of the flow respectively, and one packet is generated at the source during each time slot. The transfer threshold, $\Delta$, is set to 2 in this example. The system becomes stable only after three time slots and transfers packets from the source node to destination node with a fixed rate of one packet per time slot. What happens if the topology of the system changes in this example? As Figure 5 shows, even if the link $(1, 3)$ fails, or if one or both of the links $(1, 2)$ and $(2, 3)$ fail, the system can still deliver packets to the destination with a fixed rate of one packet per time slot.[2] Only if $(1, 3)$ and one or both of the other two links fail together, packets are accumulated at the source node, and the system becomes unstable. Obviously, no routing scheme can deliver packets to the destination in this case, simply because there is not enough capacity to carry the packets from the source to the destination.

In general, we would like to know under what circumstances a network running SF-VRS remains stable. To answer this question, let us start with a sufficient condition for a network to be strictly stable under SF-VRS.

**Theorem 1:** Let us consider a network $G = (V, E(t))$ with a single flow $(S, D, F)$. If the minimum capacity of any cut $C$ separating the nodes $S$ and $D$ is at least $F$, then network remains strictly stable under SF-VRS.

Instead of proving Theorem 1, we will prove the following stronger result.

**Theorem 2:** Let us consider a single-flow network $G = (V, E(t))$ with flow $(S, D, F)$ running SF-VRS. We assume that the minimum capacity of any cut $C$ separating the nodes $S$ and $D$ is at least $F$. Then, for any subset of nodes $U$ such that $|U| = k$, and at any time slot $t$, the total number of packets residing in $U$, denoted by $P_U(t)$ is at most
$$B(k) = \Delta \times [Nk - \frac{k(k+1)}{2}] + k, \tag{5}$$
where $N$ is the total number of nodes in the network (*i.e.* $N = |V|$) and $\Delta$ is the transfer threshold of the SF-VRS.

The proof of this theorem is left to the Appendix.

### B. *Multi-Flow Networks*

In this section, we generalize the technique presented in previous section for multi-flow networks. It's interesting to note that simply extrapolating SF-VRS (*i.e.* treating all flows like a single flow) does not work directly, as shown by the following counter-example in Figure 6. In this network there are two flows, one from node 1 to node 4, and one from node 4 to node 1, each generating one packet every other time slot. Since the number of packets in nodes 2 and 3 are always the same, if we use SF-VRS, no packets make it across the middle link $(2, 3)$. Thus, the network is unstable.

We consider a network carrying $K$ flows $(S_i, D_i, F_i)$ (for any $i$, $1 \leq i \leq K$). A simple generalization of SF-VRS to work in such a network is as follows.

**Time Division VRS (TD-VRS).** We divide the time between the $K$ flows, *i.e.* at time slot $t$ the network will be

[2]As we will see later, disruptions can happen but only for a limited number of times during the entire system lifetime.

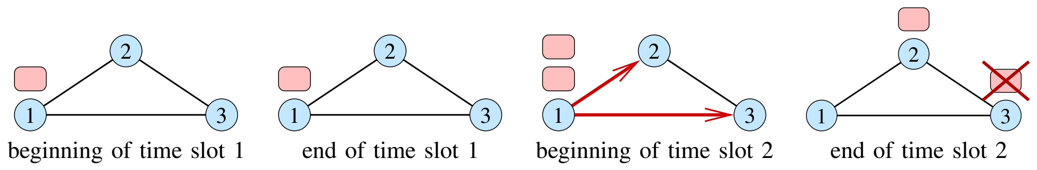
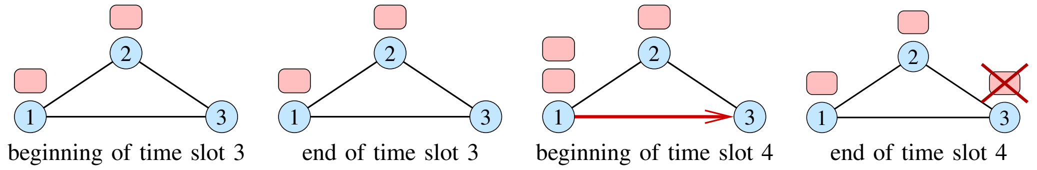


Fig. 4. An example of how SF-VRS works. At the beginning of each time slot, one packet is generated in the source node 1. All active links are shown by a red arrow while the rest of links are plotted as black lines.



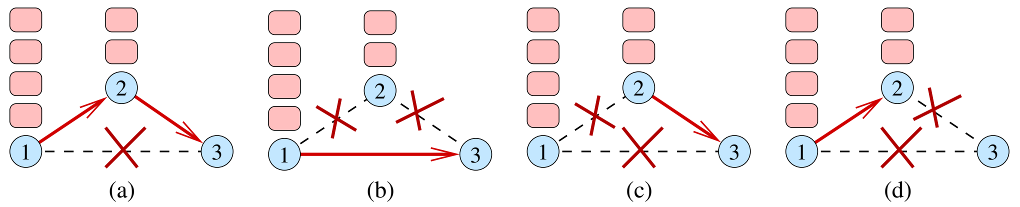


Fig. 5. The system can deliver packets from the source to the destination in cases (a) and (b). However, in case (c) and (d) no packet reaches the destination.

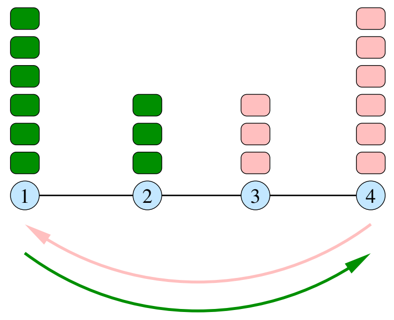


Fig. 6. An example showing why SF-VRS does not work in multi-flow networks.

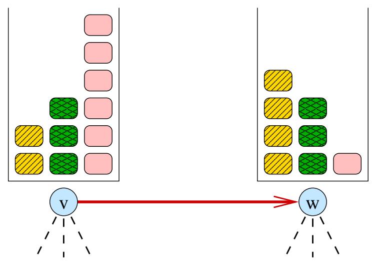


Fig. 7. An example of how MP-VRS works.

completely dedicated to the $(t \mod K+1)$-th flow, and use SF-VRS to route packets belonging to this flow.

**Theorem 3:** We consider a multi-flow network $G = (V, E(t))$ with $K$ flows $(S_i, D_i, F_i)$. We let

$$F = \max_{i=1}^{K}\{F_i\}. \tag{6}$$

If for any cut $C$ that separates $S_i$ and $D_i$ for some $i$ $(1 \leq i \leq K)$, we have $Min\_Cap(C) \geq KF$ then the network is stable under TD-VRS.

We prove this theorem in the Appendix. Note that if the number of packets injected by each flow is equal (or very close) to $F$, TD-VRS performs optimally (or near-optimally). However, if some of the flows generate very small number of packets (compared to $F$), TD-VRS wastes the capacity by not forwarding any packets at time slots assigned to such flows. In other words, even though the system remains stable, we are not using its full capacity. The following variation of VRS is intended to fix this problem.

**Maximum Pressure VRS (MP-VRS).** During the time slot $t$, we denote the number of packets corresponding to flow $i$ at node $v$ with $P_v^i(t)$. At time slot $t$, each link $(v, w)$ computes

$$i^* = \arg\max_i\{P_v^i(t) - P_w^i(t)\}. \tag{7}$$

The link $(v, w)$ transfers one packet belonging to flow $i^*$ from node $v$ to node $w$ if

$$P_v^{i^*}(t) - P_w^{i^*}(t) \geq \Delta. \tag{8}$$

Figure 7 depicts an example of how MP-VRS works. In this example, $\Delta = 2$. Here, $i^* = 3$ and since the difference between the number of packets of flow 3 residing at $v$ and $w$ is at least $\Delta$, one packet belonging to flow 3 will be transferred from $v$ to $w$.

Despite TD-VRS, which ignores the individual demands of each flow, MP-VRS gives priority to the flow with the highest

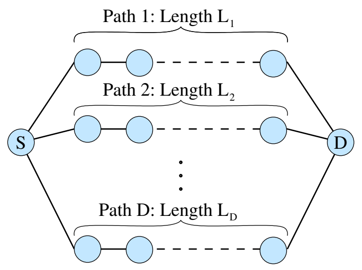


Fig. 8. There are $D$ disjoint paths $P_1, \ldots, P_D$ between the source node $S$ and the destination node $D$, such that $L_1 \leq L_2 \leq \cdots \leq L_D$.

pressure at each link. The key idea here is that if at any time slot, we balance the flow with the highest pressure at the two ends of a given link, overall all flows will become balanced throughout the network. Like SF-VRS and TD-VRS, we need some restrictions on the network connectivity so that MP-VRS keeps the system stable. The following lemma presents a necessary condition on the network topology.

**Lemma 1:** We consider a multi-flow network $G = (V, E(t))$ carrying flows $(S_1, D_1, F_1), \ldots, (S_K, D_K, F_K)$. If there is a routing scheme under which this network is rate-stable, then the network is $\overline{F} = (F_1, \ldots, F_K)$-rate-provisioned.

The proof of Lemma 1 is trivial and is omitted. We were not able to prove a sufficient condition for the stability of the system similar to that of Theorem 3 for MP-VRS. However, we will study the stability of the system under MP-VRS using simulations in Section VI.

## V. Routing Path Length

In this section we determine the length of the path taken by a packet compared to shortest-path routing. We note that no matter what the routing scheme is, it is not always possible to route all packets on the shortest path. In the network depicted in Figure 8 for example, if the demand between the source node $S$, and the destination node $D$ is two packets per time slot, no matter what the routing algorithm is, at least half of the packets cannot take the shortest path. On the other hand, if the demand between the source and destination node is less than or equal to one packet per time slot, a simple shortest-path routing scheme can route all packets on the shortest path. The following theorem, shows that the same is true with high probability in VRS.

**Theorem 4:** In a single-flow network with a fixed topology, if the amount of flow injected to the network is $1-\epsilon$ per time slot (for an arbitrarily small $\epsilon$), then there is a transfer threshold $\Delta$ for which SF-VRS almost surely (*i.e.* with probability one) routes packets on the shortest path from the source to the destination.

In general, if the demand of the flow is $D-\epsilon$, by choosing appropriate $\Delta$ we can show that packets take the first $D$ disjoint shortest paths with probability one.

**Theorem 5:** In a single-flow network with a fixed topology, if the amount of flow injected to the network is $D-\epsilon$ per time slot (for an arbitrarily small $\epsilon$), then there is a transfer threshold $\Delta$ for which SF-VRS almost surely routes packets on the first $D$ disjoint shortest paths between the source and the destination (assuming that such paths exist).

The proof of Theorem 4 and Theorem 5 comes in the Appendix.

Finally, note that since the maximum number of packets in the network increases with the transfer threshold, $\Delta$, there is a trade-off between the length of the path taken by each packet, and the maximum number of packets in the system. By reducing the threshold $\Delta$, we can reduce the number of packets that are roaming. However, this will increase the ratio between the length of the path taken by each packet and the length of the corresponding shortest path.

## VI. Simulations

In this section, we evaluate the performance of VRS using simulations. All nodes are distributed on a square surface of unit area with either a random walk or a waypoint mobility process. In the waypoint mobility model, each node chooses a random target, which is uniformly distributed in the surface, and advances toward that target at a constant velocity. When a node reaches its target, it generates a new target and moves again. In the random walk model each node moves one step (of constant size) in one of the four cardinal directions (chosen randomly), and reflects at the boundary.

We examine the performance of VRS based on several metrics: packet loss, queue size distribution, and the length of the path taken by packets. Several factors can impact each of these performance metrics, namely, the number of nodes in the network, the number and amount of flows, the mobility process, the amount of the transfer threshold $\Delta$, communication range, and the algorithm used to route the messages. In order to evaluate the impact of each of these parameters on a given performance metric, in our experiments, we fix all but one or two of these parameters, and analyze the impact of changing these parameters on the system performance. In the experiments of this section, unless stated otherwise, the network consists of 100 nodes, each node has a communication range of 0.26, movement velocity ranges from 0.01 to 0.2, and $\Delta = 2$.

**Packet Loss Ratio.** The first performance metric we study is the packet loss ratio in the system, *i.e.*, the fraction of packets that are dropped due to insufficient buffer size at the nodes of the system. We study packet loss, since it is an indication of the stability of the system; clearly, if the system is strictly stable and the queue sizes are chosen appropriately, the packet loss is equal to zero.

- **Flow Demand.** Figure 9 (a) shows the packet loss ratio (and thus the stability region) of VRS in a single-flow network, as a function of the number of packets injected to the system at each time slot. We observe that the packet loss ratio of the system does not depend on the mobility process, and is almost the same for the random walk and the waypoint processes. An interesting point here is that

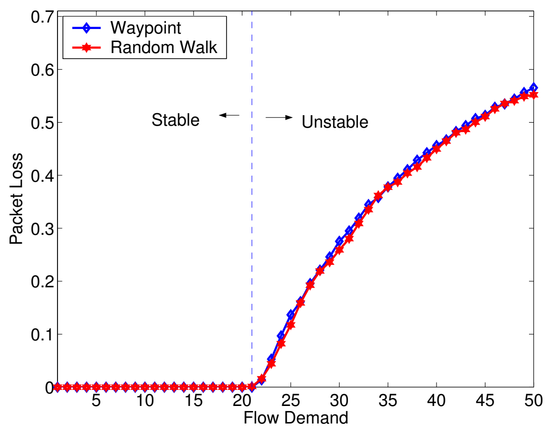


Fig. 9. Stability of VRS in a single-flow network.

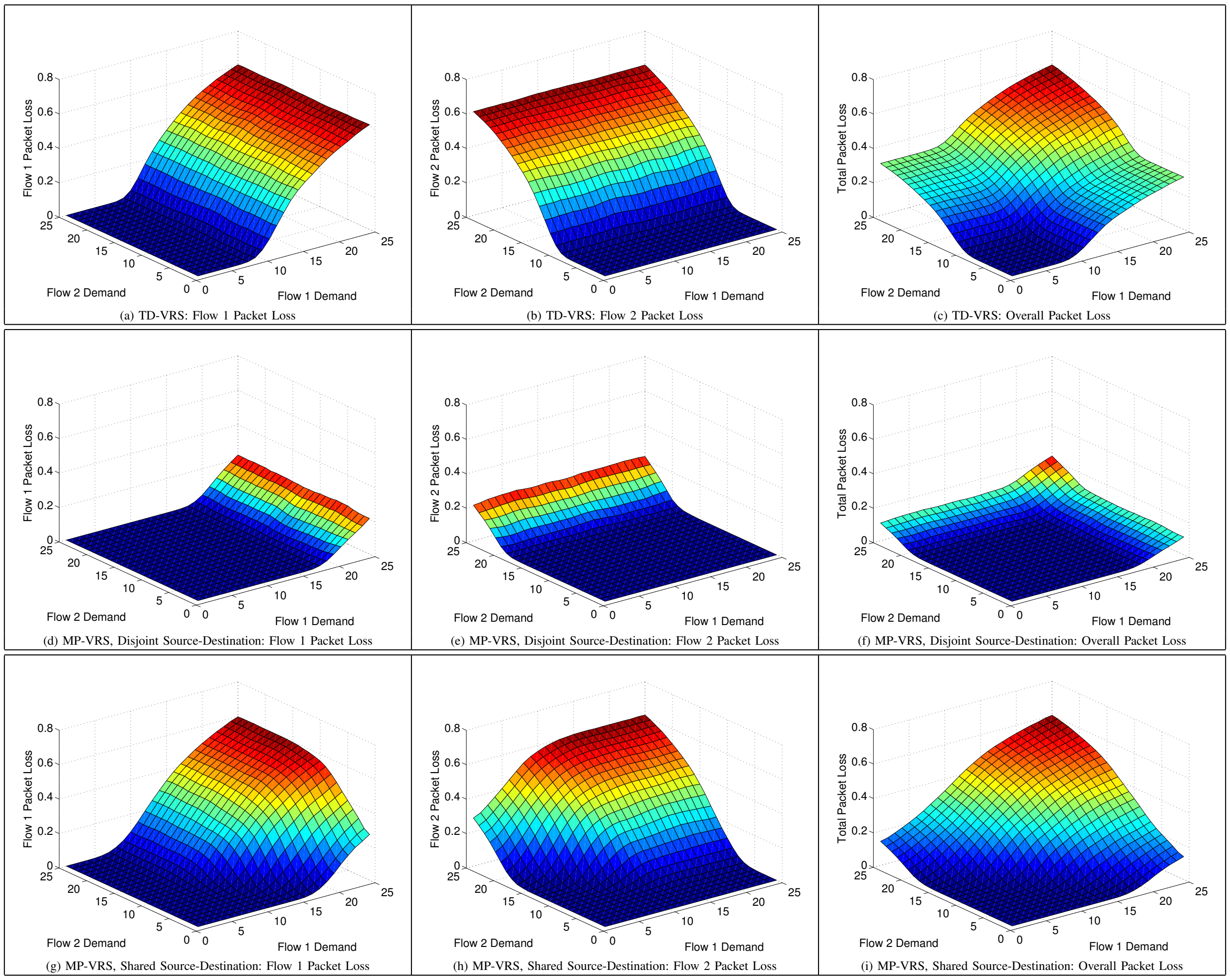
if queue sizes are chosen large enough (in our example the maximum queue size is 1000 packets) the stability of the system does not change with increasing the queue sizes. When the flow demand is beyond the capacity of the system, no matter how large the queues are, if the system runs for a sufficiently large time, the queues will start to overflow.

- **TD-VRS vs. MP-VRS.** In any single flow network TD-VRS and MP-VRS behave identically. Thus, we compare the packet loss ratio of TD-VRS and MP-VRS in a network with two flows. The demand of each flow ranges from one to twenty five packets per time slot. Parts (a), (b), and (c) of Figure 10 illustrate the amount of packet loss for flow one, the packet loss for flow two, and the total amount of packet loss in a network running TD-VRS. We can see that once the demand of flow one or flow two goes beyond nine packets per time slot, the system starts loosing a fraction of packets. However, as we can see in parts (e), (f), and (g) of Figure 10, if we run MP-VRS on the same network, the system can tolerate up to eighteen packets for each of the two flows.
The fact that the system remains stable under MP-VRS when both flows have a demand of eighteen packets per time slot, might seem unexpected at first glance. Since TD-VRS (which assigns all links to each of the two flows every other time slot) can tolerate up to nine packets per flow, we expect the system remain stable when one of the two flow demands goes up to eighteen and the other remains zero. In this case, MP-VRS will assign all links to the flow with demand eighteen, but when both flows have a demand of eighteen, we expect the system to become unstable, since the total flow is thirty six in this case. However, a closer look at our network shows that the constraining factor for instability of TD-VRS is the average number of links going out of the source. In this experiment, the average degree of each nodes is about eighteen. Since TD-VRS assigns the links to each flow every other time slot, the source node can inject up to eighteen packets in every two time slots or nine packets per time slot. However, if the source and destination of the two flows are distinct nodes, the bottle-neck is not the average degree of the source or destination anymore, and MP-VRS can route up to eighteen packet for each flow. Parts (g), (h), and (i) of Figure 10, illustrate the packet loss of a network with two flows sharing the same source node. Here, like the TD-VRS case, the bottleneck is the number of links going out of the source node, and when the demand of both flows is increased the system becomes unstable.
- **Communication Range.** For a fixed number of nodes in the network, expanding the communication range of each node increases the average degree of each node, which in turn, enhances the connectivity of the network and reduces the packet loss ratio. Figure 11 (a) depicts the packet loss of a single-flow network with a demand of fifteen packets per time slot. In Figure 11 (b) packet loss is represented as a function of the flow demand for various values of communication range.
- **Number of Nodes.** If the communication range of each nodes is fixed, increasing the number of nodes in the network reduces the packet loss[3] (Figure 12 (a)).
- **Transfer Threshold $\Delta$.** Figure 12 (b) depicts the packet loss of a network for various values of $\Delta$. In this network, the maximum queue size is 1000 packets. We observed that when the amount of $\Delta$ is relatively small compared to the maximum queue size, the packet loss ration of the system does not depend on $\Delta$. However, when $\Delta$ becomes a large fraction of the maximum size, it can significantly increase the packet loss ratio.
- **Mobility Process.** In Figure 9 we noticed that choosing between random walk or waypoint mobility process does not affect the stability of the system. Here, we measure the impact of changing the velocity on the packet loss ratio of the system. We can see in Figure 13 (a) that reducing the velocity to values near zero, can lead to subtle instabilities in the system. The reason is that in this case topology changes are very slow, therefore, if some node happens to have a very small number of adjacent neighbors, it might degrade the routing process. We notice that for larger velocities, the packet loss ratio of the system does not depend on the node movement speed (Figure 13 (b)).

**Queue Size Distribution.** The average queue size for any given flow is an important metric for evaluating the performance of the routing scheme, since it is directly proportional to the average delay observed by packets. To see this, we consider an $N$-node network carrying a flow with demand $D$. We denote the average number of packets residing in the network at any time slot with $P_{avg}$, and the average delay observed by packets with $W_{avg}$. Then,

$$P_{avg} = D \times W_{avg}, \tag{9}$$

[3]We assume that a higher number of nodes does not increase the interference.



(a) TD-VRS: Flow 1 Packet Loss
(b) TD-VRS: Flow 2 Packet Loss
(c) TD-VRS: Overall Packet Loss
(d) MP-VRS, Disjoint Source-Destination: Flow 1 Packet Loss
(e) MP-VRS, Disjoint Source-Destination: Flow 2 Packet Loss
(f) MP-VRS, Disjoint Source-Destination: Overall Packet Loss
(g) MP-VRS, Shared Source-Destination: Flow 1 Packet Loss
(h) MP-VRS, Shared Source-Destination: Flow 2 Packet Loss
(i) MP-VRS, Shared Source-Destination: Overall Packet Loss

Fig. 10. Packet loss ratio for TD-VRS vs. MP-VRS.

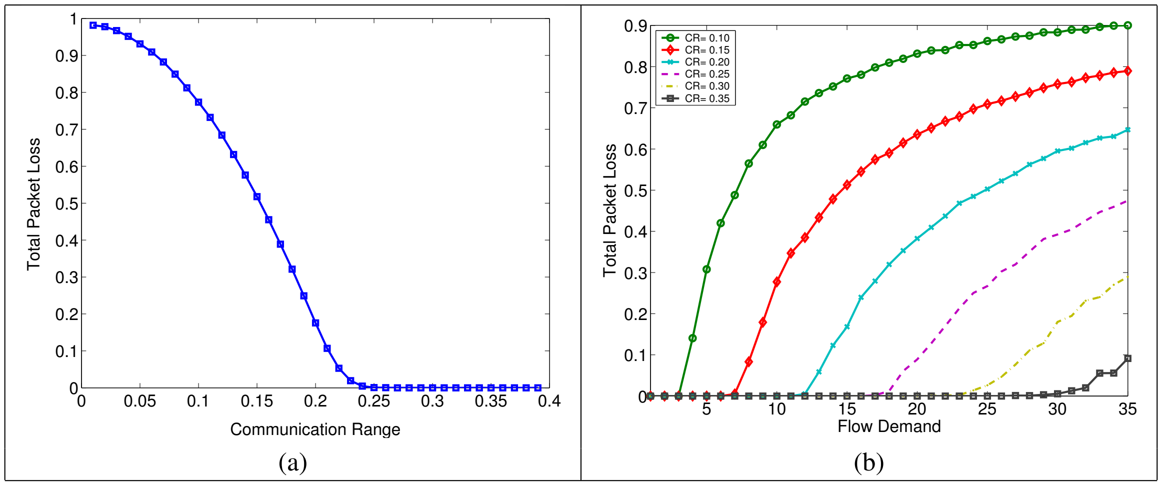


Fig. 11. Packet loss vs. communication range.

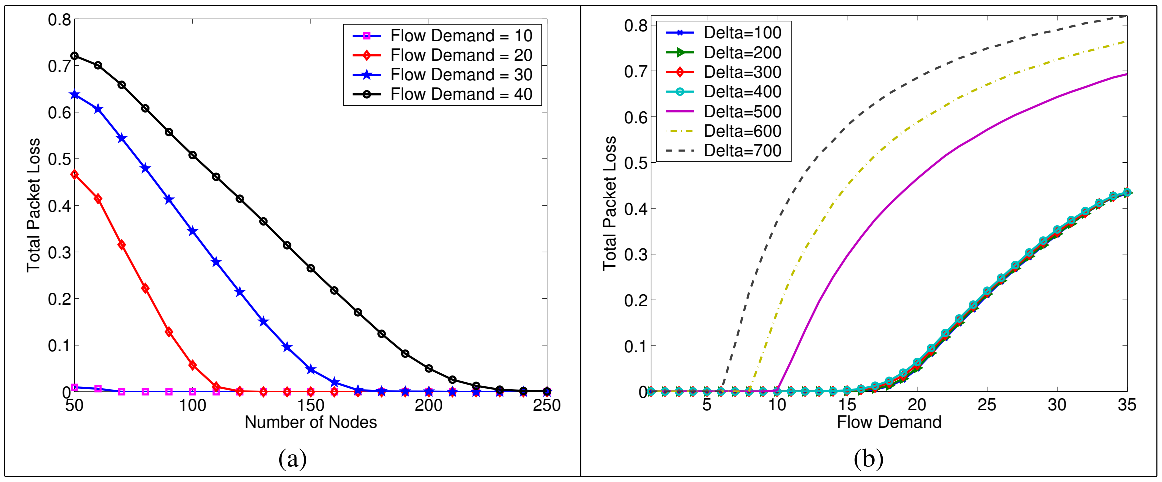


Fig. 12. (a) Packet loss vs. the number of nodes in the network. (b) Packet loss vs. transfer threshold $\Delta$.

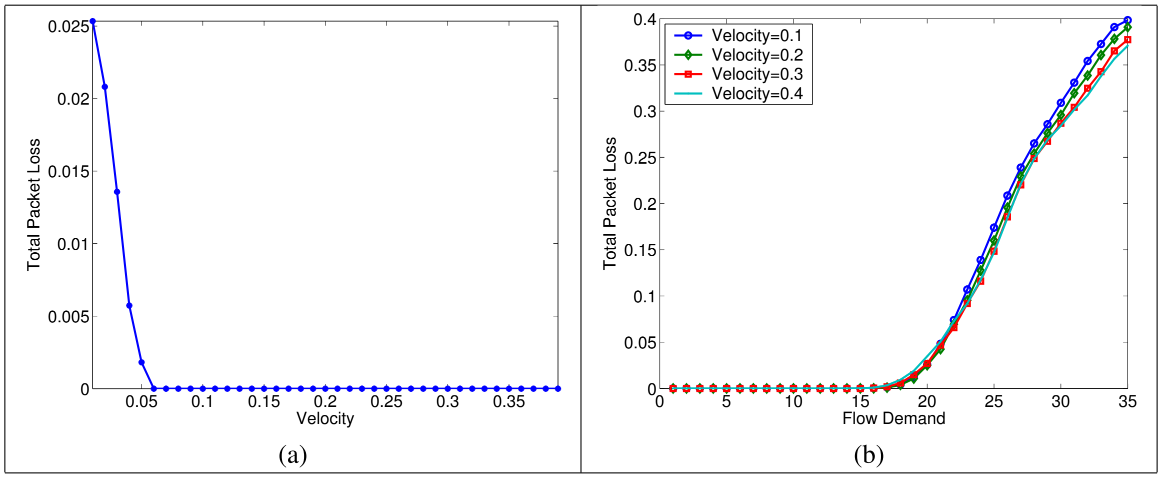


Fig. 13. The packet loss ratio of the network vs. the node movement velocity.

and,

$$P_{avg} = Q_{avg} \times N. \tag{10}$$

Therefore,

$$W_{avg} = \frac{Q_{avg} \times N}{D}. \tag{11}$$

Equation 11 shows the relationship between the average delay observed by packets and the average queue size. Next, we will study the impact of changing system parameters on queue size distribution.

- **Flow Demand.** Figure 14 (a) depicts the average queue size, maximum queue size and the standard deviation of queue sizes in a network as a function of the flow demand. We can see that when the network is lightly loaded, the average queue size and the maximum queue size are very close to each other. Only when the network becomes heavily loaded, the maximum queue size grows severely.
  Figure 14 (b) shows the queue size distribution for different amount of flow demand. We can see that 99% of the time the queue sizes are less than ten, even though the maximum queue size is about 100. We notice that the theoretical bound on the number of packets residing in the network (Theorem 2), which is 199 packets per nodes, is far from the number of packets in practice. In other words, in practice we expect the packets to have a relatively short delay under VRS.
- **TD-VRS vs. MP-VRS.** Figure 15 parts (a), (b) and (c) respectively present the average queue size in a two-flow network under TD-VRS, MP-VRS with disjoint source-destination nodes, and MP-VRS with a node which is a source of two flows. Comparing these graphs with those in Figure 10 we can see that as unless the network is very highly loaded, the average queue size remains small under both TD-VRS, and MP-VRS.
- **Communication Range, Number of Nodes, and the Mobility Process.** Our simulations show that when the network is not highly loaded, the average and maximum queue sizes do not change with the communication range, the number of nodes, and the mobility process. However, when the network becomes highly loaded as a result

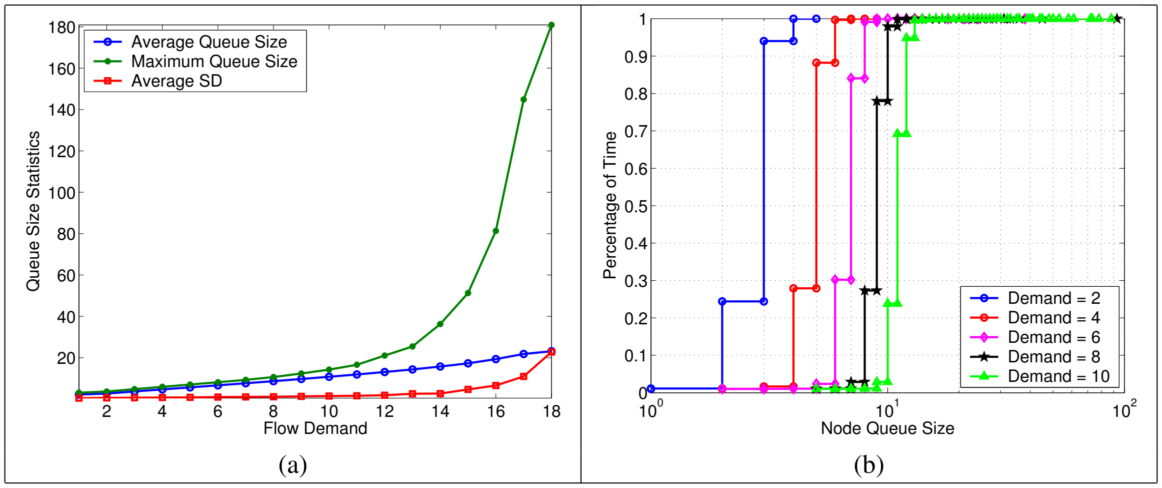


Fig. 14. Queue size distribution vs. flow demand.

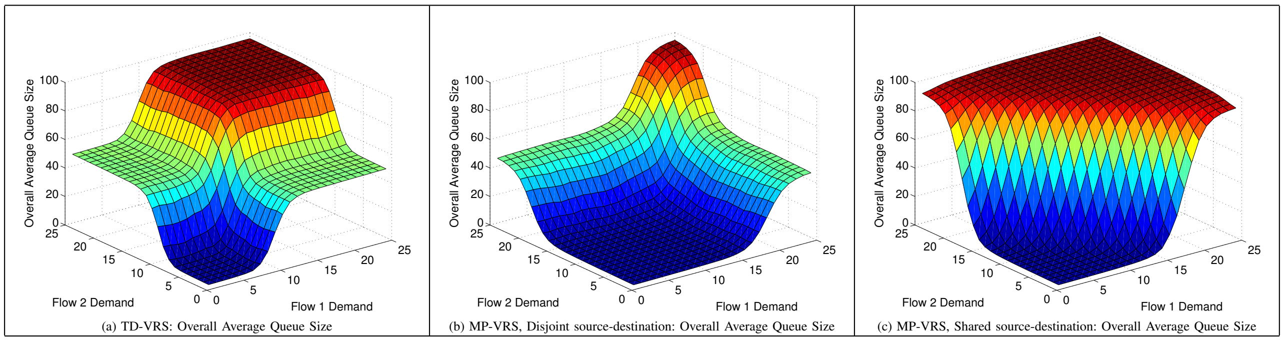


Fig. 15. Average queue size under TD-VRS and MP-VRS in a two flow network.

of reducing the communication range or decreasing the number of nodes, the average and maximum queue sizes increase (similar to Figure 14).

- **Transfer Threshold $\Delta$.** Figure 16 (a) shows the average and maximum queue size in the network as a function of the transfer threshold $\Delta$. As expected, the average and maximum queue sizes grow when $\Delta$ is increased. We note that unless $\Delta$ is set to very large amounts (relative to the maximum available queue size), the average and maximum queue sizes remain small.

**Routing Path Length.** Finally, we evaluate the performance of VRS based on the routing path length of the packets. Note that the routing path length and the delay are not necessarily correlated, since a packet might wait for a long time in a queue even though it is routed on the shortest path. Based on our analysis in Section V, the only parameter that directly impacts the path length is the transfer threshold $\Delta$. Figure 16 (b) depicts the ratio between the routing path length under VRS and the shortest path in a lightly loaded network with a fixed topology. Note that when $\Delta = 0$, nodes move randomly in the network, but as soon as $\Delta = 1$ paths start to take the shortest path.

## VII. Conclusions

While mobile ad-hoc networks are new, and few systems have been deployed, it is not yet clear how dynamic they will be. Perhaps, over time, experience will show them to be relatively static, with only slow changes in topology. However, it is reasonable to expect that mobility will be hard to describe or constrain; and applications might emerge which are highly dynamic, with rapidly moving sensors. In this case, existing routing algorithms will be of little use.

In rapidly changing networks in which nodes are periodically (although perhaps infrequently) generating new packets, then Volcano Routing seems simple (it has no discovery phase), reliable (packets will eventually reach their destination), and stable (the number of packets in the network is bounded). We don't know of any other routing algorithm with these properties.

## References

[1] Dimitri P. Bertsekas and Robert Gallager, *Data Networks*, Prentice Hall, second edition, 1992.

[2] Joseph M. Kahn, Randy H. Katz, and Kristofer S. J. Pister, "Next century challenges: Mobile networking for "smart dust"," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. 1999, pp. 271–278, ACM Press.

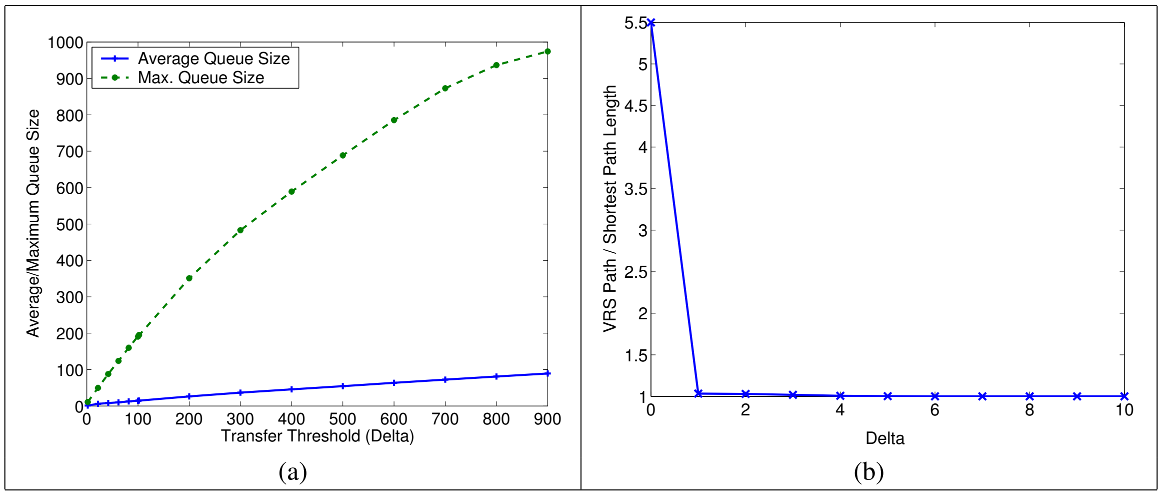


Fig. 16. (a) Average and maximum queue sizes vs. the transfer threshold $\Delta$. (b) Path length ratio vs. $\Delta$.

[3] Henri Dubois-Ferriere, Matthias Grossglauser, and Martin Vetterli, "Age matters: Efficient route discovery in mobile ad hoc networks using encounter ages," in *Proceedings of ACM MobiHoc'03*, Maryland, USA, June 2003.

[4] Charles E. Parkins, Elizabeth M. Royer, and Samir R. Das, "Ad-hoc on-demand distance vector routing," in *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, February 1999, pp. 90–100.

[5] Vincent D. Park and M. Scott Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proceedings of the IEEE INFOCOM'97*, Kobe, Japan, April 1997.

[6] Atsushi Iwata, Ching-Chuan Chiang, Guangyu Pei, Mario Gerla, and Tsu-Wei Chen, "Scalable routing strategies for ad hoc wireless networks," *IEEE Journal on Selected Areas in Communication*, vol. 17, no. 8, pp. 1369–1379, August 1999.

[7] Charles E. Parkins, Elizabeth M. Royer, Samir R. Das, and Mahesh K. Marina, "Performance comparison of two on-demand routing protocols for ad hoc networks," *IEEE Personal Communications*, vol. 8, no. 1, pp. 16–28, February 2001.

[8] Tomasz Imielinski and Julio C. Navas, "GPS-based geographic addressing, routing, and resource discovery," *Communications of the ACM*, vol. 42, no. 4, pp. 86–92, April 1999.

[9] Sung-Ju Lee, *Routing and Multicasting Strategies in Wireless Mobile Ad Hoc Networks*, Ph.D. thesis, Computer Science Department, UCLA, 2000.

[10] Charles E. Perkins and Pravin Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the conference on communications architectures, protocols and applications*. 1994, pp. 234–244, ACM Press.

[11] George Aggelou and Rahim Tafazolli, "RDMAR: a bandwidth-efficient routing protocol for mobile ad hoc networks," in *Proceedings of the 2nd ACM international workshop on Wireless mobile multimedia*. 1999, pp. 26–33, ACM Press.

[12] Sung Ju Lee, William Su, and Mario Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," *Mobile Networks and Applications*, vol. 7, no. 6, pp. 441–453, 2002.

[13] David B. Johnson, "Scalable and robust internetwork routing for mobile hosts," in *Proceedings of the 14th International Conference on Distributed Computing Systems*, Poznan, Poland, June 1994, IEEE Computer Society, pp. 2–11.

[14] Zygmunt J. Haas and Marc R. Pearlman, "The performance of query control schemes for the zone routing protocol," *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 427–438, 2001.

[15] Julio C. Navas and Tomasz Imielinski, "Geocastgeographic addressing and routing," in *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*. 1997, pp. 66–76, ACM Press.

[16] Rahul Jain, Anuj Puri, and RAja Sengupta, "Geographical routing using partial information for wireless ad hoc networks," *IEEE Personal Communications*, vol. 8, no. 1, pp. 48–57, February 2001.

[17] Baruch Awerbuch, Petra Berenbrink, André Brinkmann, and Christian Scheidler, "Simple routing strategies for adversarial systems," in *FOCS'01*, Las Vegas, Nevada, October 2001, pp. 158–167.

[18] Baruch Awerbuch, Yishay Mansour, and Nir Shavit, "End-to-end communication with polynomial overhead," in *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science (FOCS)*, 30 October – 1 November 1989, pp. 358–363.

[19] Baruch Awerbuch and Tom Leighton, "Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks," in *Proceedings of the Twenty-Sixth Annual ACM Symposium on the Theory of Computing (STOC)*, Montréal, Québec, Canada, 23–25 May 1994, pp. 487–496.

[20] Anindya Basu, Alvin Lin, and Sharad Ramanathan, "Routing using potentials: A dynamic traffic-aware routing algorithm," in *Proceedings of ACM SIGCOMM'03*, Karlsruhe, Germany, August 2003, pp. 37–48.

## Appendix

### Stability of SF-VRS

We need to prove a number of preliminary results before proving the stability of SF-VRS (Theorem 1 and Theorem 2).

**Lemma 2:** At a given time $t$, if there is a subset of nodes $U$, with $|U| = k$ and $P_U(t) > B(k)$, we can find a subset of nodes $U'$ consisting of $k$ nodes such that for any node $v \in U'$ and any node $w \notin U'$ we have $P_v(t) \geq P_w(t)$.

*Proof:* If for any node $v \in U$ and any node $w \notin U$ we have $P_v(t) \geq P_w(t)$, then we let $U' = U$ and the lemma is proved. Otherwise, there must be a node $v \in U$, such that $P_v(t) < P_w(t)$ for some node $w \notin U$. We can swap the two nodes getting another set $U'$ (*i.e.* $U' = (U - v) \cup w$). The number of elements in $U'$ is still $k$, but the number of packets residing in $U'$ is more than that of $U$. We can repeat this process for only a limited number of times, which proves the lemma. $\square$

**Lemma 3:** At a given time $t$, if $U$ is a minimal subset of nodes with $|U| = k$ and $P_U(t) > B(k)$, for any node $v \in U$ we have

$$P_v(t) > \Delta \times (N - k) + 1. \tag{12}$$

*Proof:* Since $U$ is a minimal set with $P_U(t) > B(k)$, we have $P_{(U-\{v\})}(t) \leq B(k-1)$ for any node $v$ in $U$. We also know that $P_{(U-\{v\})}(t) = P_U(t) - P_v(t)$; therefore

$$P_v(t) > B(k) - B(k-1), \quad (13)$$

or

$$P_v(t) > \Delta \times (N-k) + 1. \quad (14)$$

□

Now we are ready to prove the main result of Section IV-A.

*Proof:* (**Theorem 2**). There are no packets residing in the network at time zero. Let us assume time slot $t$ is the first time that there is a minimal subset $U$ ($|U| = k$) with $P_U(t) > B(k)$. Let us also assume that $U$ is composed of the nodes which have at least as many packets buffered as any other node in the network (Lemma 2 shows that such a set always exists).

The source node resides in $U$ (otherwise we cannot have $P_U(t) > B(k)$), therefore $F$ new packets have entered $U$ during the time slot $t-1$. Since at time $t-1$ we have $P_U(t-1) \leq B(k)$ the number of packets which have left $U$, say $F'$ is less than $F$.

Let us consider the cut $C = (U, V-U)$ separating $S$ and $D$. Since the network is $F$-min-provisioned, the capacity of $C$ is at least $F$, and there are at least $F$ nodes outside $U$ which are adjacent to some node in $U$ at time slot $t-1$. Based on Lemma 3 and the fact that $U$ is minimal, for any node $v \in U$ we have $P_v(t-1) > \Delta \times (N-k) + 1$. Now, since only $F'$ packets have left $U$ at time $t-1$, at least $F - F'$ neighbors of $U$ must have packets more than $\Delta \times (N-k-1) + 1$ residing in them (otherwise the number of packets leaving $U$ would be more than $F'$, which is not possible). Let us denote this set of neighbors of node in $U$ with $U'$.

We consider the set of nodes $U \cup U'$ at time $t-1$. This set has $k + F - F'$ nodes in it and

$$P_{(U \cup U')}(t-1) >$$
$$B(k) - F + F' + \Delta \times (F - F')(N-k-1) + F - F'. \quad (15)$$

Simplifying this equation we get

$$P_{(U \cup U')}(t-1) > B(k + \mathcal{F} - \mathcal{F}'). \quad (16)$$

This is not possible (time $t$ is the first time for which the bound is broken), therefore the theorem holds. □

## Stability of TD-VRS

*Proof:* (**Theorem 3**) During any period of $K$ time slots, the total number of packets generated by the $i$-th flow ($1 \leq i \leq K$) is at most $KF$. Since the capacity of any cut separating $S_i$ and $D_i$ is at least $KF$, and since all the links are dedicated to flow $i$ at least once during these $K$ time slots, Theorem 2 shows that the system is strictly stable with respect to flow $i$.

□

## Routing Path Length Under VRS

In order to prove Theorem 4 we need the following lemma.

**Lemma 4:** In a single-flow network with a fixed topology, and flow demand $1-\epsilon$, for any node $v$ at distance $dist(v)$ of the destination node, we almost surely have

$$P_v(t) \leq \Delta \times dist(v). \quad (17)$$

*Proof:* We use mathematical induction on the distance of the nodes from the destination. The only node with $dist(v) = 0$ is the destination node $D$, for which we always have $P_D(t) = 0$. Let us assume that for any node at distance less than or equal to $m-1$ of the destination, the lemma holds. Now, consider a node $v$, with $dist(v) = m$. There is a node $w$, with $dist(w) = m-1$ which is connected to $v$. Based on induction hypothesis, we know that with probability one $P_w(t) \leq \Delta \times dist(w)$. Therefore, if $P_v > \Delta \times dist(v)$, the link $(v, w)$ will drain packets from $v$ with rate equal to one. However, we know that with probability one the arrival rate to node $v$ is less than or equal to $1-\epsilon$, and whenever the arrival is greater than $1-\epsilon$ it is limited to the number of adjacent nodes to $v$. Therefore, whenever there are more than one packets arriving at $v$, they queue is immediately drained, since the draining rate is one. Therefore, with probability one inequality 17 holds. □

*Proof:* (**Theorem 4**) Let us assume that the length of the shortest path, say $P$, is $L$. Let us also consider another path, $P'$, of length $L' > L$. packets take path $P'$ to reach the destination node, we must have

$$P_S - (\Delta - deg) \times (L' - 1) \geq \Delta. \quad (18)$$

where $deg$ is the maximum degree of any node in the network. To see this we consider a packet $X$ at the source node $S$, which follows the path $P'$ to reach the destination. Let us assume that the height of $X$ is the size of the queue in which it resides in. Then, the height of $X$ will be $P_S$ at the beginning. Now, whenever $X$ is transferred from a node $v$ to a node $w$, since $P_v - P_w \geq \Delta$ and the maximum number of packets arriving to $w$ is $deg$, the height of $X$ will be reduced by at least $\Delta - deg$. However, since $X$ reaches the destination, its height when it arrives to an adjacent node of the destination must be at least $\Delta$ (otherwise it will not be transferred to the destination). Thus, inequality 18 holds. We can rewrite this inequality as follows.

$$P_S \geq (\Delta - deg) \times (L' - 1) \geq \Delta. \quad (19)$$

But based on Lemma 4 we know that with probability one

$$P_S \leq \Delta \times L. \quad (20)$$

Combining inequalities 19, and 20 we get

$$L' \leq \frac{\Delta \times L}{\Delta - deg} - \frac{\Delta}{\Delta - deg} + 1. \quad (21)$$

We can see that as $\Delta \to \infty$, the length of any path $L'$ taken by packets must be very close to $L$, the length of the shortest path. Therefore, if we let $\Delta$ to be large enough, all the packets must follow the shortest path with probability one. □