Virtual Back-Pressure Routing in Mobile Ad-Hoc Networks for Disaster Scenarios

Daeho Kang
School of Electrical Engineering
and Computer Science
Seoul National University
Email: dhkang@netlab.snu.ac.kr

Changehee Joo and Sunjung Kang
Electrical and Computer Engineering
Ulsan National Institute of
Science and Technology
Email: {cjoo, sky12382}@unist.ac.kr

Saewoong Bahk
School of Electrical Engineering
and Computer Science
Seoul National University
Email: sbahk@netlab.snu.ac.kr

***Abstract*—In disaster situations, cellular infrastructure can be severely damaged and may not function as a way of communication for emergency service. Mobile handsets can play a pivotal role by composing mobile ad-hoc networks and relaying urgent messages to distant live base stations. In deploying such emergency networks, low-complexity routing schemes that are scalable and reliable are the key to success. Previous routing schemes suffer from low performance due to lack of scalability, lengthy queueing delay, and failure to support high mobility. In this paper, we develop a novel routing protocol that exploits the well-known back-pressure technique. By introducing 'virtual back-pressure', our proposed scheme reduces end-to-end delay and quickly responds to topology changes, while preserving the advantages of the original back-pressure scheme; provable efficiency and capability of dynamic routing. Through simulations, we demonstrate that the proposed scheme significantly outperforms other competitive schemes in packet delivery ratio, end-to-end delay, and routing overhead.**

***Index Terms*—Routing, mobile ad-hoc networks, disaster recovery.**

I. Introduction

In modern cellular communication systems, a mobile device is directly connected to and communicates with a base station to provide voice and data service. However, in disaster scenarios or battle fields, communication infrastructure such as base stations and power stations could be severely damaged, and mobile devices may not find a live base station within its communication range. Such a lost of connectivity can be critical in emergency situation or military mission.

Mobile Ad-Hoc Network (MANET) consisting of mobile devices can be a solution to such situations and delivers packets via hop-by-hop communications to a live base station [1]. Recent advances in device-to-device communications in cellular networks also offer great potential to this approach [2]. One of the most challenging issues in MANET-based disaster recovery networks is development of a practical routing protocol for reliable path selection [3], [4]. In disaster scenarios, wireless nodes can be highly mobile, densely deployed in a small area, and can generate heavy traffic load from video streaming applications for remote treatment of patients or remote controls. Thus, the routing protocol should be able to deal with frequent changes of topology and traffic load in a robust and scalable manner.

Many routing protocols for wireless MANETs have been developed in the literature to provide efficient and reliable end-to-end packet delivery [5]–[9]. They can be grouped into two categories: Proactive routing protocols, such as DSDV [6] and OLSR [9], maintain a routing table in advance for possible data delivery, and reactive routing protocols, such as DSR [5], TORA [8], and AODV [7], try to find a route whenever a node has a data packet to deliver. Although these routing schemes achieve considerable performance in MANET [10], [11] in terms of packet delivery ratio, end-to-end delay, and routing overhead, their performance is limited due to the nature of single-path routing and load-unawareness, and they often fail to support high mobility and dynamic routing.

It has been shown in [12] that multi-path routing protocols can provide more reliable packet delivery and have the potential of dynamic routing to traffic load changes. However, creating and updating multiple routing path information in highly mobile environments is a challenging issue and can significantly complicate the operation of routing protocols.

An optimal solution, named the back-pressure scheme, that can exploit multi-path diversity has been developed [13], [14]. It considers routing and scheduling jointly, and has been shown to achieve optimal throughput. However, it often takes long to deliver data packets to the destination, in particular under low traffic load, since packets are likely to be forwarded at a random direction when most queues are empty. It also suffers from the last packet problem, where the last packet of a flow waits for a long time since there is no subsequent packet injection in the flow [15]. The poor delay performance makes the back-pressure scheme less attractive for disaster scenarios, where urgent messages should be delivered immediately.

In [16], the authors improve the delay performance of the back-pressure scheme, by introducing the shadow (or virtual) queue that evolves as data packets arrive and depart. They decouple scheduling and routing in the back-pressure scheme and design a probabilistic routing table to remove inefficient routing. Along with the introduction of the virtual queue, it has been also shown that Last-In-First-Out (LIFO) queue is helpful to improve the delay performance of the back-pressure scheme [17]. An effort to reduce the delay by combining the back-pressure routing and the shortest-path routing without relying on the virtual queue has been also put [18]. The authors of [18] consider the hop-count constraints in the optimization problem, resulting in a routing solution with low end-to-end delay. However, the delay performance of the above schemes is still not acceptable in practice. They rely on arrivals and departures of data packets for evolving the queues, which takes long to build the desirable back-pressure and causes a critical delay before establishing a valid path; especially when a user sends an emergency message without subsequent traffic, or when the network topology changes due to mobility.

In this paper, we develop a light-weight routing protocol for disaster scenarios. We aim at designing reliable routing schemes for large-scale mobile ad-hoc networks that can support high node mobility while achieving low delay and high throughput performance. Our main contributions are as follows.

- We develop a novel proactive routing scheme that quickly adapts to topology change and achieves good delay performance. Using a virtual-queue-based routing measure, we separate the back-pressure operations from the data traffic, leading to a fast response to route failure.
- We show that the proposed routing scheme is provably efficient and capable of dynamic routing. It successfully avoids congested links and achieves high packet delivery ratio and low end-to-end delay under heavy traffic load.
- We design a light-weight implementation of the proposed routing protocol and improve its performance in practice by taking into account important implementation issues.
- We verify the operation of the proposed scheme in comparison with the original back-pressure scheme, and evaluate its performance through NS-3 simulations along with other MANET routing protocols of DSDV, AODV and OLSR.

The rest of the paper is organized as follows. The system model is described in Section II. We develop the virtual back-pressure routing scheme, and explain its operation and routing strategy in Section III. We improve the proposed scheme by taking into account practical implementation issues in Section IV. We verify the operation of our proposed scheme and compare its performance with those of other MANET routing schemes through simulations in Section V. Finally, we conclude our paper in Section VI.

II. System Model

We consider a network graph $G(V,E)$ with the set V of mobile nodes and some fixed base stations that formally compose cellular systems (or Wi-Fi networks), and the set E of wireless links, where base stations are attached to backhaul networks. Due to the limited power of wireless signal transmission, most nodes can be connected to the base stations via multi-hop relays.

We consider a time-slotted system. The routing protocol starts its operation at time 0, when a disaster occurs. We assume that the traffic flows only between mobile nodes and a base station, which is typical in disaster scenarios [4], and focus on the upward traffic toward the base station in this work. For the downward traffic to mobile nodes, the base station can use source routing that exploits the routing information collected from received packets.

Let $A_i(t)$ denote packet arrivals at node i, which is upper bounded by A^{max} and has a mean $\lambda_i := \mathbb{E}[A_i(t)]$. We consider that all data packets, regardless of their originating nodes, go to a single base station node d. It is straightforward to extend our results to multiple destinations (i.e., routing to one of the destinations; anycast). Let $Q_i(t)$ denote the queue length at node $i \in V$. Clearly, $Q_d(t) = 0$ at the base station d. For ease of exposition, we assume a unit link capacity for all the links. It has been well-known that the solution to the following joint problem of routing and scheduling, named as the back-pressure scheme, can achieve the maximum throughput [13].

$$I^*(t) = \underset{\{I_{ij}\} \in F}{\operatorname{argmax}} \sum_{(i,j)\in E} \left(Q_i(t) - Q_j(t)\right) I_{ij}, \qquad (1)$$

where $I_{ij} \in \{0,1\}^{|E|}$ denotes the system decision of link (i,j), where $I_{ij} = 1$ if that link (i,j) is active, and $I_{ij} = 0$ otherwise, and F denotes the set of all feasible schedules that satisfy underlying interference constraints. The back-pressure scheme serves packets with $I^*(t)$ at each time t, which incorporate routing functionality by forwarding packets into the node with the least queue length in the transmitter's neighborhood.

Unlike the back-pressure scheme, most of practical routing schemes have been developed based on the layered architecture, separated from scheduling functionality. In general, routing protocol starts with broadcasting a routing message and detecting neighboring nodes (or links). Once a node recognizes a neighbor from received routing messages, they exchange routing information by local broadcasting, global flooding, or hop-limited flooding. Each node may have the entire network topology information or direction information to the destination (i.e., the base station). OLSR [9] is an example of the former case, and DSDV [6] belongs to the latter case. Also, routing information can be established in an on-demand fashion when the source has data packets (e.g., DSR [5] and AODV [7]), in which case the source floods a route request message throughout the network. When the request is received by the destination or an intermediate node who already has the route information to the destination, it sends a route reply message back to the source, and completes a route establishment between the source and the destination. The performance of the routing protocol can be affected by the frequency of routing information exchanges and information propagation range. It can be measured by successful data delivery ratio, end-to-end delay, routing overhead, and computational complexity.

In this work, we focus only on the routing problem. Motivated by the back-pressure scheme, we develop practical dynamic routing schemes that are scalable, can be rapidly deployed in disaster areas, and can achieve low delay and high throughput even in highly mobile environments.

III. Virtual Back-Pressure Routing

It has been known that the back-pressure scheme has poor delay performance in light traffic load and suffers from the last packet problem [15]. For example, when there is only one data packet in the network, due to lack of sufficient pressure, the packet can be forwarded to a random direction until it accidentally arrives at the destination. We remedy this weakness by building artificial pressure using virtual packets that are irrelevant with actual data packets, and use this virtual back-pressure information to route data packets. We start with explanation of our proposed scheme, named Virtual Back-Pressure Routing (VBPR), and then show how the virtual queue information can be used to forward data packets.

A. Creating virtual back-pressure

We consider a time-slotted system, where a time slot is a period in which a node conducts regular operation to detect its neighbor, i.e., broadcasting a Hello message. We assume that all nodes are synchronized, which will be relaxed later, and do not take into account scheduling.

When a disaster occurs, a few base stations that retain connection to backhaul networks, may survive and switch to the disaster ad-hoc mode. We assume that all data packets are destined for the backhaul networks via one of the live base stations. At time 0, the live base station begins to broadcast the ad-hoc Hello message. Each node who receives the Hello message changes its mode to disaster ad-hoc mode and creates routing table entry toward the transmitter of the Hello message. Then it forwards the Hello message to its neighbors by local broadcasting. Repeating the procedure, all the nodes located within multi-hop communication range from a base station can find a path to the base station. Each node also broadcasts the Hello message *periodically* at a rate of one per time slot.

Once a node switches to the disaster ad-hoc mode, it starts generating virtual packets and building pressure as follows. Let N_i denote the set of neighboring nodes of node i. At each time slot t, each node i, except the base station, generates virtual packets following a Poisson distribution with mean λ_i^v and adds them to its virtual queue $Q_i^v(t)$. Virtual packets do not require physical storage and the virtual queue is simply a counter. Let $A_i^v(t)$ denote the number of virtual packets generated at node i at time slot t. Since all the data packets go to the base station, each node can build a tree rooted at the base station by setting a routing parent node $p_i^v(t) \in N_i$. In particular, VBPR sets the routing parent node $p_i^v(t)$ as the node that has the least virtual queue length among node i's neighbors, i.e.,

$$p_i^v(t) = \underset{j \in N_i}{\operatorname{argmax}} \left(Q_i^v(t) - Q_j^v(t) \right). \tag{2}$$

If none of N_i has smaller virtual queue than i, $p_i^v(t)$ is set to Null. Unless $p_i^v(t)$ is Null, node i transmits $S_{ip_i^v}^v(t) := \min\{Q_i^v(t), D_{ip_i^v}^v(t)\}$ virtual packets to node $p_i^v(t)$ during time slot t, where $D_{ip_i^v}^v(t)$ denotes the maximum number of virtual packets that can be served in time slot t, which will be determined later. After transmitting virtual packets, node i decreases its virtual queue $Q_i^v(t)$ by the amount of $S_{ip_i^v}^v(t)$. The transmission of virtual packets is done by piggybacking onto the Hello message, which now includes the following additional fields; the address of the receiver $p_i^v(t)$ (of virtual packets), the number of virtual packets $S_{ip_i^v}^v(t)$, node i's leftover virtual queue length $Q_i^v(t)$, and node i's data queue length $Q_i(t)$. (See line 11 of Algorithm 1 for details.)

Although the Hello message specifies the receiver under VBPR, it is locally broadcasted[1] for virtual queue update such that all the neighboring nodes can update the virtual and data queue length information about node i. If node j finds that it is the receiver of the virtual packets specified in the Hello message from node i, it increases its virtual queue length by $S_{ij}^v(t)$. Hence, the virtual queue evolves as

$$Q_i^v(t+1) \le \left(Q_i^v(t) + A_i^v(t) - D_{ip_i^v}^v(t) \right)^+ + \sum_{k \in N_i} D_{ki}^v(t), \tag{3}$$

where $(\cdot)^+ := \max\{0, \cdot\}$, and $D_{ki}^v(t) = 0$ if $p_k^v(t) \neq i$. The last term $\sum_{k \in N_i} D_{ki}^v(t)$ indicates the contribution of node i's neighboring nodes to the increase in the node i's virtual queue length. The base station discards all the received virtual packets and sets $Q_d^v(t) = 0$ for all t.

The virtual queue length information can be used to forward data packets. For instance, if the virtual queue lengths are monotonically decreasing toward the base station, a simple greedy forwarding strategy will deliver data packets to the base station. To this end, we aim at shaping the virtual queue lengths decreasing toward the base station *linearly* in hop-distance, which will direct the data packets to the base station via the shortest path.

Now, we explain how to determine the service rate of the virtual queues. Note that from (3), we can keep the virtual queue of a node finite by setting its service rate greater than its aggregated arrival rate. On the other hand, in order for the virtual queues to linearly decrease toward the base station, we have to carefully control their service rates such that they remain at a certain level in a consistent manner.

Suppose that the system is in steady state and the routing decision is static with $p_i^v(t) = j$ so that node i forwards virtual packets to node j. We drop subscript t if there is no confusion. Then we have

$$\begin{aligned} \mathbb{E}[Q_i^v] - \mathbb{E}[Q_j^v] &= \mathbb{E}[Q_i^v - Q_j^v] \\ &\le \mathbb{E}[Q_i^v - Q_j^v \; | Q_i^v \ge Q_j^v] \cdot P(Q_i^v \ge Q_j^v), \end{aligned} \tag{4}$$

since $\mathbb{E}[Q_i^v - Q_j^v \; | Q_i^v < Q_j^v] < 0$. Suppose that there exists stationary probability $\tilde{P}_i := P(Q_i^v \ge Q_j^v)$. We approximate the virtual queue difference between node i and node j as an M/M/1 queue with arrival rate $\tilde{\lambda}_i$ and service rate $\tilde{\mu}_i \cdot \tilde{P}_i$ as

[1] Due to the intrinsic nature of wireless transmission, some virtual packets can be lost in reality. Fortunately, the simulation results show that unless most virtual packets are lost, the loss of some virtual packets does not make a significant impact on the performance.

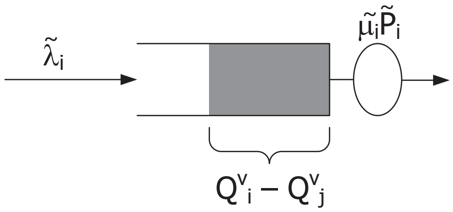


Fig. 1. Approximation of virtual queue difference to an M/M/1 queue.

shown in Fig. 1, and estimate the queue difference as

$$\mathbb{E}[Q_i^v - Q_j^v \,|Q_i^v \geq Q_j^v] \approx \frac{\frac{\tilde{\lambda}_i}{\tilde{\mu}_i \tilde{P}_i}}{1 - \frac{\tilde{\lambda}_i}{\tilde{\mu}_i \tilde{P}_i}}, \tag{5}$$

where $\tilde{\lambda}_i$ and $\tilde{P}_i$ can be measured and $\tilde{\mu}_i$ needs to be determined. Let Δ denote the target queue difference between two consecutive nodes over the route. By setting $\Delta = \mathbb{E}[Q_i^v] - \mathbb{E}[Q_j^v]$ and combining (4) and (5), we can obtain

$$\tilde{\mu}_i \leq \tilde{\lambda}_i \left(\frac{1}{\Delta} + \frac{1}{\tilde{P}_i} \right). \tag{6}$$

For linearly decreasing virtual back-pressure with the difference Δ, we set the virtual packet service rate to $\tilde{\mu}_i = \tilde{\lambda}_i \left(\frac{1}{\Delta} + \frac{1}{\tilde{P}_i} \right)$, which will also stabilize Q_i^v since $\tilde{\mu}_i \tilde{P}_i > \tilde{\lambda}_i$. Thus, at each time t, we choose the number of forwarding virtual packets $D_{ij}^v(t)$ following a Poisson distribution with mean $\tilde{\mu}_i$, where $\tilde{\lambda}_i$ and $\tilde{P}_i$, respectively, can be obtained by measuring incoming virtual packet arrivals, i.e., $\tilde{\lambda}_i = \lambda_i + \frac{1}{t} \sum_{\tau=0}^{t} \sum_{k \in N_i} S_{ki}^v(\tau)$, and by calculating the rate of no transmission as $\tilde{P}_i = 1 - \frac{1}{t} \sum_{\tau=0}^{t} \mathbb{1}_{\{p_i(\tau)=\text{Null}\}}$, where $\mathbb{1}_{\{\cdot\}}$ denotes the indicator function of having Null parent. We set the generation rate of virtual packets to be identical for all $i \in V$, i.e., $\lambda_i^v = \lambda$.

Remarks: Our setting is based on rough approximation, and the arrival and the departure processes may be a non-Poisson process in practice. However, under our routing scheme, it suffices to provide finite virtual queue lengths while maintaining their differences at a certain level. This is confirmed through simulations in Section V-A (See Fig. 2).

B. Data packet forwarding

Given the state of the virtual queues, the data packets can be forwarded to the node with the least virtual queue length in a greedy manner. Then, under light traffic load, each data packet can simply follow the maximum virtual queue difference and will quickly arrive at the base station through the shortest path. However, under heavy traffic load, data packets may need to avoid congested links and exploit path diversity for high throughput and low delay. To this end, we take into account both the virtual queues and the data queues to direct data packets.

Suppose that node i has a data packet and gets a chance to transmit it to the next hop. If the transmission occurs in time slot t[2], it finds the next hop $p_i(t)$ as[3]

$$p_i(t) = \underset{j \in N_i}{\operatorname{argmax}} \left((Q_i^v(t) + Q_i(t)) - (Q_j^v(t) + Q_j(t)) \right), \tag{7}$$

where $Q_i(t)$ denotes the data queue length at node i. Algorithm 1 describes the detailed operations of VBPR to build the virtual back-pressure and to forward the data packets.

Algorithm 1 Virtual Back-Pressure Routing

Each node i at the beginning of each time slot:

- Update $\tilde{P}_i$.
- Update $\tilde{\mu}_i$ according to (6) with new $\tilde{P}_i$ and measured $\tilde{\lambda}_i$.
- Get parent node j from (2).
- **if** parent node j is Null **then**
  - $D_{ij}^v = 0$.
- **else**
  - Pick D_{ij}^v randomly following a Poisson process with mean $\tilde{\mu}_i$.
- **end if**
- $S_{ij}^v = \min\{Q_i^v, D_{ij}^v\}$.
- Decrease Q_i^v by S_{ij}^v.
- Broadcast Hello message containing j, S_{ij}^v, Q_i^v, Q_i periodically.

Each node i on receiving a Hello message from node k:

- Update queue length information Q_k^v and Q_k.
- **if** node k forwards virtual packets to node i **then**
  - Increase Q_i^v by the number of the virtual packets S_{ki}^v.
- **end if**

Routing at each node i:

- **if** $Q_i > 0$ **then**
  - Get the next hop node from (7).
  - Transmit a data packet to the next hop.
- **end if**

We claim that our data forwarding (7) can exploit multipath diversity to the base station given the stabilized virtual queues. To elaborate on this, we show that our routing scheme can achieve optimal throughput when it is combined with the MaxWeight scheduling scheme, which is also used in [13], [14].

Proposition 1: The joint solution of our routing protocol (7) and MaxWeight is throughput-optimal provided that all the virtual queue lengths remain finite.

Note that since the generation of virtual packets and their forwarding are independent from the data packets, the stationary behavior of the virtual queues can be achieved regardless of the data queues. Once the virtual queue lengths are shaped

[2]We abuse the notation here. The time t is indeed a scheduling time, and the scheduling decisions are usually made with a finer time granularity (some micro seconds) than the routing decisions (a few seconds). In this case, we consider the virtual queue length $Q_i^v(t)$ as the virtual queue length at scheduling time slot t.

[3]We can also extend the policy such that the decision is based on the difference of the weighted sum of the two queues (e.g., $Q_i^v(t) + \beta Q_i(t)$). Then along with Δ, the weight β can be used to adjust VBPR's dynamic routing behaviors.

well, we can think that the data packets build back-pressure *with offset of the virtual queues*, flowing to the lowest pressure. Hence, when a link is congested with data packets, its data queue will build up, and the increased pressure will cause subsequent data packets to detour around congested areas.

Proof: We prove the proposition using the Lyapunov technique. Under mild assumptions, we show that for any feasible arrival rate strictly within the capacity region, we can find a Lyapunov function that has a negative drift when data queue lengths are sufficiently large and the virtual queue lengths remain finite. This implies the stability of the network system.

We consider a time-slotted system and assume that each time slot is divided into two exclusive time intervals: one for routing message exchanges and the other for data packet transmissions. We assume that all routing control messages including the Hello messages are exchanged during the first interval in an interference-free manner, and the data packets are transmitted during the second interval under interference constraints. The assumption can be justified by the fact that transmissions of routing control messages rarely collide due to small message size. Further, for ease of exposition, we assume that each link can transmit one data packet during the second interval, which can be easily generalized.

For data packet scheduling, we employ the well-known MaxWeight scheduling scheme [13] that activates the feasible set of links $\{I_{ij}(t)\} \in F$ that maximizes the weighted rate sum

$$\sum_{(i,j)\in E} w_{ij}(t) I_{ij}(t), \tag{8}$$

where $w_i(t) := \max_{j\in N_i} \big((Q_i^v(t)+Q_i(t)) - (Q_j^v(t)+Q_j(t))\big)$, and each data queue $Q_i(t)$ evolves satisfying that

$$Q_i(t+1) \le \left(Q_i(t) + A_i(t) - \sum_{k\in N_i} I_{ik}(t)\right)^+ + \sum_{k\in N_i} I_{ki}(t). \tag{9}$$

We define the Lyapunov function

$$\mathcal{V}(t) := \sum_{i\in V} \left(Q_i^v(t) + \frac{Q_i(t)}{2}\right)\frac{Q_i(t)}{2},$$

and its drift

$$\partial\mathcal{V}(t) := \mathbb{E}[\mathcal{V}(t+1) - \mathcal{V}(t) | Q(t)],$$

where $Q(t)$ denotes the vector of data queues. We show that the drift is negative for sufficiently large queue lengths. Note that if $Q_i(t) = 0$, we have $\left(Q_i^v(t+1) + \frac{Q_i(t+1)}{2}\right)\frac{Q_i(t+1)}{2} - \left(Q_i^v(t) + \frac{Q_i(t)}{2}\right)\frac{Q_i(t)}{2} \le \left(Q_i^v(t+1) + \frac{1}{2}(A^{max}+1)\right) \cdot \frac{1}{2}(A^{max}+1)$, which will be bounded if the virtual queue remains finite. Hence, we consider only the queues with $Q_i(t) > 0$. We drop subscript t if there is no confusion.

Letting $Z_i := \frac{1}{2}(A_i - \sum_{k\in N_i} I_{ik} + \sum_{k\in N_i} I_{ki})$, we have

$$\begin{aligned}
&\mathcal{V}(t+1) - \mathcal{V}(t) \\
&\quad \le \textstyle\sum_i (Q_i^v + \frac{Q_i}{2}) \cdot \frac{Q_i}{2} + \sum_i (Q_i^v + \frac{Q_i}{2}) \cdot Z_i \\
&\qquad + \textstyle\sum_i (A_i^v - \sum_{k\in N_i} D_{ik}^v + Z_i) \cdot \frac{Q_i}{2} \\
&\qquad + \textstyle\sum_i (A_i^v - \sum_{k\in N_i} D_{ik}^v + Z_i) \cdot Z_i \\
&\qquad - \textstyle\sum_i (Q_i^v + \frac{Q_i}{2}) \cdot \frac{Q_i}{2} \\
&\quad = \textstyle\sum_i (Q_i^v + Q_i) \cdot Z_i + \sum_i \frac{Q_i}{2}(A_i^v - \sum_{k\in N_i} D_{ik}^v) \\
&\qquad + \textstyle\sum_i (A_i^v - \sum_{k\in N_i} D_{ik}^v + Z_i) \cdot Z_i.
\end{aligned} \tag{10}$$

We consider the conditional expectation of each term of (10). Note that if the arrival vector is feasible, there must exist a stationary solution $\{\tilde{I}_{ij}\}, \{\tilde{A}_i^v\}, \{\tilde{D}_{ij}^v\}$ such that

$$\begin{aligned}
&\mathbb{E}[A_i - \textstyle\sum_{k\in N_i} \tilde{I}_{ik} + \sum_{k\in N_i} \tilde{I}_{ki}] < 0, \\
&\mathbb{E}[\tilde{A}_i^v - \textstyle\sum_{k\in N_i} \tilde{D}_{ik}^v] < 0.
\end{aligned} \tag{11}$$

Let $\tilde{Z}_i := \frac{1}{2}(A_i - \sum_{k\in N_i} \tilde{I}_{ik} + \sum_{k\in N_i} \tilde{I}_{ki})$. Taking the conditional expectation on the first term of (10), we can obtain

$$\begin{aligned}
&\mathbb{E}[\sum_i (Q_i^v + Q_i) \cdot Z_i \ |Q] \\
&= \mathbb{E}[\sum_i (Q_i^v + Q_i) \cdot \tilde{Z}_i \ |Q] - \mathbb{E}[\sum_i (Q_i^v + Q_i) \cdot (\tilde{Z}_i - Z_i) \ |Q] \\
&= \mathbb{E}[\sum_i (Q_i^v + Q_i) \cdot \tilde{Z}_i \ |Q] \\
&\quad + \frac{1}{2}\mathbb{E}\left[\sum_i (Q_i^v + Q_i) \cdot (\sum_{k\in N_i} \tilde{I}_{ik} - \sum_{k\in N_i} \tilde{I}_{ki}) \ |Q\right] \\
&\quad - \frac{1}{2}\mathbb{E}\left[\sum_i (Q_i^v + Q_i) \cdot (\sum_{k\in N_i} I_{ik} - \sum_{k\in N_i} I_{ki}) \ |Q\right] \\
&= \mathbb{E}[\sum_i (Q_i^v + Q_i) \cdot \tilde{Z}_i \ |Q] \\
&\quad + \frac{1}{2}\mathbb{E}\left[\sum_i \big((Q_i^v + Q_i) - (Q_j^v + Q_j)\big)\, \tilde{I}_{ij} \ |Q\right] \\
&\quad - \frac{1}{2}\mathbb{E}\left[\sum_i \big((Q_i^v + Q_i) - (Q_j^v + Q_j)\big)\, I_{ij} \ |Q\right].
\end{aligned}$$

From (11) and the state independence of packet arrivals and scheduling of the stationary solution, we have that for some small $\epsilon > 0$, $\mathbb{E}[\sum_i (Q_i^v + Q_i) \cdot \tilde{Z}_i \ |Q] < -\epsilon\mathbb{E}[\sum_i (Q_i^v + Q_i) \ |Q]$. Also, the operation of MaxWeight scheme (8) implies that the sum of the second term and the third term is non-positive. Hence, we can conclude that

$$\mathbb{E}[\sum_i (Q_i^v + Q_i) \cdot Z_i \ |Q] < -\epsilon\mathbb{E}[\sum_i (Q_i^v + Q_i) \ |Q]. \tag{12}$$

For the conditional expectation of the second term of (10), since the virtual packet arrivals and departures are independent of the data queues and all the virtual queues remain stable, we should have

$$\mathbb{E}[A_i^v - \textstyle\sum_{k\in N_i} D_{ik}^v \ |Q] = \mathbb{E}[A_i^v - \sum_{k\in N_i} D_{ik}^v] \le 0. \tag{13}$$

Similarly, since $A_i \leq A^{max}$ and $\sum_{j\in N_i} I_{ij} \leq 1,\ \forall i \in V$, we can also obtain from the third term of (10) that

$$\mathbb{E}[\textstyle\sum_i (A_i^v - \sum_{k\in N_i} D_{ik}^v + Z_i)\cdot Z_i \ |Q] \leq B_1, \tag{14}$$

for some constant B_1.

Combining (12), (13), and (14), we have

$$\begin{aligned}\partial\mathcal{V}(t) = \mathbb{E}\Big[&\textstyle\sum_i (Q_i^v + Q_i)\cdot Z_i + \sum_i \frac{Q_i}{2}(A_i^v - \sum_{k\in N_i} D_{ik}^v) \\ &+ \textstyle\sum_i (A_i^v - \sum_{k\in N_i} D_{ik}^v + Z_i)\cdot Z_i \ \Big|Q\Big] \\ &\leq -\epsilon\mathbb{E}\Big[\textstyle\sum_i (Q_i^v + Q_i) \ \Big|Q\Big] + B_1.\end{aligned} \tag{15}$$

Therefore, for sufficiently large data queues, the Lyapunov function has a negative drift, which implies the system stability by Foster's Lemma [23]. ■

By creating the virtual pressure, VBPR can greatly improve the delay performance (as shown in Section V). Also, it inherits the advantages of the original back-pressure scheme: provable efficiency, capability of dynamic routing, and low complexity for the next-hop calculation.

So far, we have developed VBPR based on steady-state system with time synchronization. In practice, however, the time synchronization may not be available, and the routing protocol should quickly react to system dynamics. We revise our proposed VBPR for more practical settings and address important practical issues.

IV. Implementation of VBPR

We consider a practical network system, where the network functionalities are implemented over the layered architecture. We assume that each node has one data queue at the MAC layer for all the data packets, and has no data queue at the network layer. Under VBPR, node i uses its queue length at the MAC layer as $Q_i(t)$ and has a counter for $Q_i^v(t)$. Each node broadcasts Hello packets asynchronously, but at the same rate of one per unit time. It generates and forwards virtual packets, and also direct data packets as described in Algorithm 1. In addition to the core operations of VBPR, the following operations provide reliable end-to-end connectivity even under high mobility and loss of routing messages, and improve response time and robustness of VBPR to system dynamics.

A. Initial set-up of virtual back-pressure

Since each node generates virtual packets at a constant rate λ, it may take long for a leaf node to build virtual back-pressure at the beginning, in particular, when the network size is large. For example, if a leaf node is 10-hop distant from the base station, it would take at least $10\Delta/\lambda$ time.

Note that once a base station enters the disaster ad-hoc mode, it triggers mode change of the nodes in the network by broadcasting Hello messages. We reduce the set-up delay of virtual back-pressure by changing the behavior of the node who receives for the first time an ad-hoc Hello message with virtual queue length x as follows.

1) The node changes to the disaster ad-hoc mode,
2) adds in its routing table the transmitter of the Hello message,
3) initializes its virtual queue with $x + \Delta$, and
4) broadcasts Hello message with $(x + \Delta)$.

This procedure sets the initial virtual queue lengths of the nodes such that they quickly build up and well-shaped, i.e., linearly increasing with difference Δ per hop distance to the base station. It is also possible for a node to wait for a random amount of time after receiving the first Hello message before it sets its initial parent node to select a node of the least virtual queue length more appropriately.

B. Purging routing information

In practice, especially in mobile environments such as disaster scenarios, wireless links become unavailable when a node moves out of the transmission range. If the routing table entry of an unavailable link is not removed quickly, data packets can be transmitted to the unreachable next hop and suffer from unnecessary long delay or can be even discarded at the MAC layer. This feature is not new and has already been used in many routing schemes, e.g., DSDV and OLSR.

Under VBPR, if a node receives no Hello message from a neighboring node for a certain period, it purges that node from its neighbor list (or from the routing entry list) to quickly remove the stale routing information. Throughout simulations (whose results are omitted due to lack of space), we have observed improvement in packet delivery ratio up to 50% in high-mobility scenario compared to VBPR without purging.

C. Recovering route loss

In certain cases, a node could have the smallest virtual queue length among its neighborhood, due to topology changes or irregular virtual packet arrivals. Then the node has Null parent and does not have valid route information to the base station. The problem will be resolved as time goes on. The node with no route will generate or receive virtual packets, and soon get higher virtual pressure than one of its neighbors, establishing new route. In the meantime, it would result in loss of some data packets since all data packets arrive at the node with no route will be dropped.[4] The problem would be worsened when scheduling and packet transmission operate at much smaller time granularity than the routing protocol, which is conventional. For example, the default interval of Hello message broadcast in OLSR is 2 sec [9], while the transmission of 1000-byte data packet takes no more than 0.2 msec over a wireless link of 54 Mbps.

We alleviate the route loss problem as follows.

1) A node does not select a node with Null parent as its next hop. That is, in (7), N_i only contains neighboring nodes with valid parents. This will let data packets detour around the dead end.

[4] If the network layer has a queue, data packets can be stored in it, which, however, still cannot prevent long delay until the new route information is established.

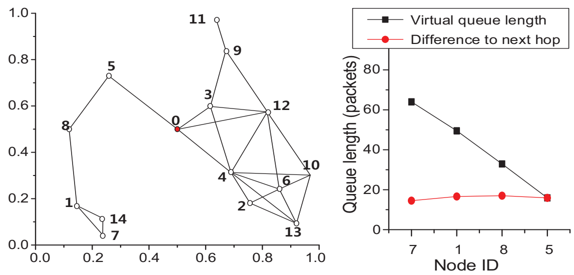


Fig. 2. Left: network topology with 15 nodes on 1×1 space, where any two nodes are connected by an edge if their distance is no greater than 0.35. The base station (node 0) is located at the center. Right: average virtual queue lengths of nodes $5, 8, 1,$ and 7 that form a linear subgraph.

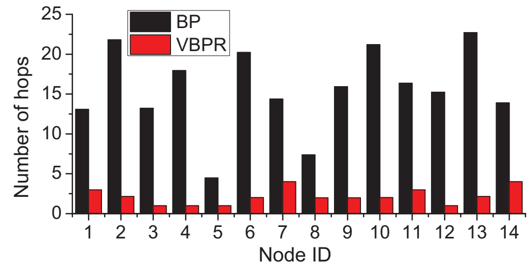
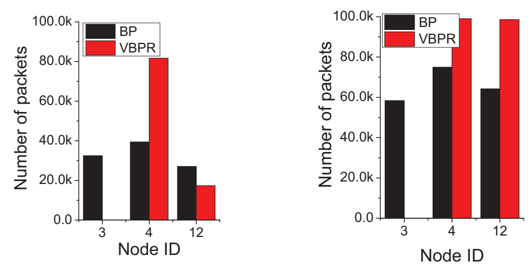


Fig. 3. Average number of hops that a data packet traverses from different source nodes. Thanks to the virtual back-pressure, most packets take the shortest path under VBPR.



(a) Rate 0.5 pkt/slot (b) Rate 1 pkt/slot

Fig. 4. Number of data packets that traverse the nodes of interest (i.e., IDs 3, 4, 12 in x-axis) immediately before arriving at the base station. Two source nodes 2 and 13 generate data packets at the same rate of either 0.5 or 1.0 packet per slot. Under VBPR, most packets are routed to the shortest path (i.e., via node 4) even when the arrival rate is very close to the capacity of the path. When the arrival rate far exceeds the capacity, VBPR actively detours data packets to the second shortest path (i.e., via node 12) and evenly balances the traffic load.

2) When a node has Null parent, it purges all its routing information and immediately transmits a special message, named 'Hello Message Trigger'. The node who receives this trigger message responses with 'Trigger Response' message, after a random amount of time. The Trigger Response message is of the same format as Hello message except that there is no virtual packet receiver. Receiving the Trigger Response message, the node with no route can initialize its virtual queue length as in Section IV-A and thus can quickly recover the route loss.

We have evaluated the advantage of the modifications through simulations (whose results are omitted due to lack of space), and observed that they improve the response time of VBPR to a route loss as well as increase the packet delivery ratio up to 25% compared to VBPR without them.

V. Numerical Results

In this section, we evaluate the performance of VBPR through simulations. We first simulate VBPR in a synchronous time-slotted system with a static setting, which clarifies the advantage of VBPR over the original back-pressure (BP) scheme. We also consider more practical scenarios with mobility, and implement our VBPR in NS-3 simulator incorporating the modifications provided in Section IV, and compare its performance with those of other well-known MANET routing protocols including DSDV, AODV and OLSR. Under different mobility scenarios, we measure packet delivery ratio, average delay, and control overhead.

A. Performance comparison in static environments

We verify the basic operations of VBPR (i.e., Algorithm 1) in a static setting and compare its performance with that of the original back-pressure (BP) scheme under different traffic loads.

We consider a network topology as shown in Fig. 2, where 15 nodes are distributed over 1×1 space. We assume that two nodes within distance 0.35 can communicate directly. The base station (node 0; the destination) is located at the center $(0.5, 0.5)$. Time is slotted, and there is no wireless interference for Hello messages and data packet transmissions. We assume *equal time-scale control* for routing and scheduling: during a time slot, each node can transmit one Hello message and one data packet. The assumption of the same time granularity for routing and scheduling decision may not be practical, since multiple data packets are often transmitted during the time interval of Hello messages. However, it shows the impact of routing scheme on throughput performance under heavy traffic load, and can be easily extended to the case of different time granularity. We will remove this assumption later when we evaluate VBPR in more practical settings. For VBPR, we set $\Delta = 10$ (packets) and $\lambda = 1$ (packet per time slot).

First, we ensure whether the virtual back-pressure is built up as we intended. We run Algorithm 1 for 10^5 time slots with no data packet and measure average virtual queue lengths for nodes $5, 8, 1,$ and 7 that compose a linear topology as shown in Fig. 2. The right one in Fig. 2 shows that the virtual queue lengths are linearly decreasing toward the base station, and well shaped with consistent difference to the next hop. The difference is a bit greater than our setting $\Delta = 10$ due to approximation and measurement errors. However, it is sufficient for VBPR to maintain a reasonable difference of virtual queue lengths between two neighboring nodes.

Next, we investigate the efficiency of VBPR under light traffic load. At every 100 time slots, we randomly select a node and generate a data packet at the chosen node. Then we measure the number of hops that the packet traverses before it arrives at the base station. To avoid any bias in the routing decision when more than two next-hop nodes have the same weight, we break the tie at random. Fig. 3 shows the average number of hops (y-axis) that a data packet traverses from its source node (x-axis). The results clearly show that under VBPR, most data packets arrive at the base station following the shortest path, while it takes much longer under BP.

In the next simulation, we examine our theoretical result on the dynamic routing of VBPR under heavy traffic load. We generate data packets at nodes 2 and 13 of the topology shown in Fig. 2, at an identical generation rate of A packets per slot. We measure how many data packets go through one of the three nodes $3, 4,$ and 12 that are directly connected to the base station. Since there is no interference and one data packet can be forwarded at each time slot, a path via node 4 that is the shortest path for both nodes 2 and 13 can handle up to 1 packet per time slot. This implies that if $A > 0.5$, some data packets must detour around node 4 to avoid buffer overflow and to exploit the path diversity. Fig. 4 shows the results of VBPR and BP with $A = 0.5$ and 1. The results demonstrate that VBPR not only successfully detours excessive traffic load to the second shortest path (via node 12), but also balances the traffic load *only when the shortest path is congested*.

B. Performance comparison in dynamic settings

In this section, we evaluate VBPR in more practical settings and compare its performance with those of DSDV, AODV, and OLSR that are provided in NS-3 [21], [22]. Since BP is a joint scheduling and routing solution and cannot directly work in an asynchronous system, it is not included in this set of simulations. We implement our VBPR in NS-3 simulator, taking into accounting the practical modifications shown in Section IV.

We consider a network topology where 30 nodes are randomly distributed within a rectangular area of 1500m $\times$ 500m as shown in Fig. 5. For the Physical and MAC layers, we use ad-hoc mode WiFi protocol stacks with Distributed Coordinate Function (DCF) from the IEEE 802.11a standard. We use the Friis propagation loss model implemented in NS-3, and set the transmission power to 3 dBm. In our settings and topology, any two nodes are within 7-hop distance.

Under all the routing protocols, a node checks its neighbors by periodically exchanging Hello messages. We set the Hello message interval to 3 seconds for all the routing protocols, and use the hop count as the routing metric, except VBPR, which uses both virtual and data queue lengths. For VBPR, we set[5] $\Delta = 50$ packets, $\lambda = 0.33$ packet per second, and the purge time as 3.5 seconds. For other parameters of DSDV, AODV, and OLSR, we use their default values defined in NS-3.

We run simulations for 1000 seconds and measure the average virtual queue length of each node to verify its operations in practical settings. Fig. 5 show the results, where the number beside each node indicates the average virtual queue length. We can observe that the virtual queue lengths of the nodes proportionally increase with the distance to the base station. This implies that VBPR successfully controls virtual packets providing the intended pressure.

[5] Since multiple data packets can be transmitted within 3 seconds that equal one time slot for VBPR, we set a higher value of Δ and a lower value of λ than in Section V-A. A study on the impact of these parameter settings is beyond the scope of the paper and remains as future work.

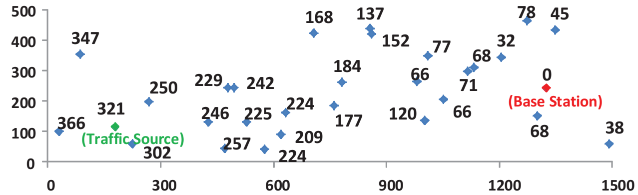


Fig. 5. Network topology with 30 nodes. One base station is located on the right side. The number beside each node denotes its average virtual queue length after 1000 seconds. As intended, each node's virtual queue length increases with its distance from the base station.

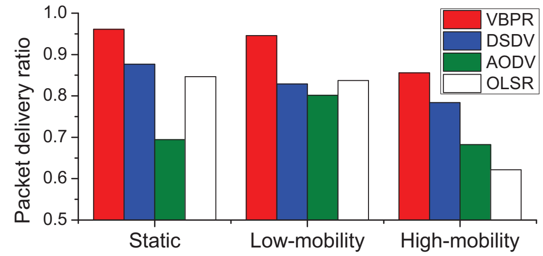


Fig. 6. Packet delivery ratio.

We compare the performance of each routing protocol under low traffic load in terms of packet deliver ratio, average delay, and routing message overhead. We choose 5 nodes at random and each of which generates data traffic at a constant rate of 1 packet per second for 300 seconds. All the data packets are destined for the base station and the packet size is set to 512 bytes.

We simulate three different mobility scenarios: static, low-mobility, and high-mobility. In static scenario, all the nodes stay in their initial positions, and in mobility scenarios, each node moves following a random waypoint model with zero-pause time. The speed or velocity is randomly chosen in range of $[1, 5]$ m/s for low-mobility (walking speed) and $[1, 20]$ m/s for high-mobility (vehicle speed), respectively. We ran 10 independent simulations and averaged the results.

Fig. 6 illustrates the packet delivery ratio under each routing protocol. VBPR outperforms all the other routing schemes and achieves the best delivery ratio. Throughout our simulations, we have observed that AODV sometimes fails to find a path to the base station when the base station is located far from the source. We conjecture that the message flooding method of AODV is unreliable especially when the flooding source and the destination are far from each other. On the other hand, DSDV successfully finds a valid route as VBPR. However, when there is a link failure or signal weakening due to

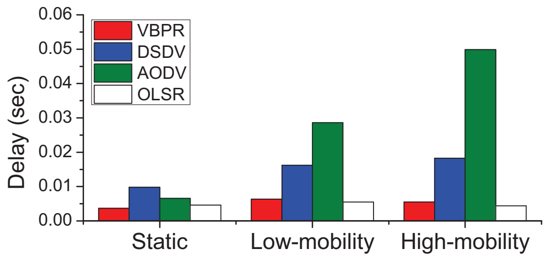
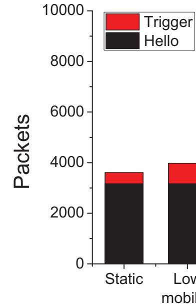
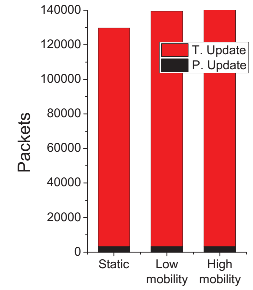


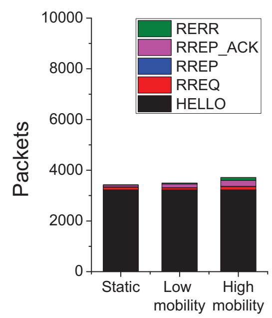
Fig. 7. Average end-to-end delay.



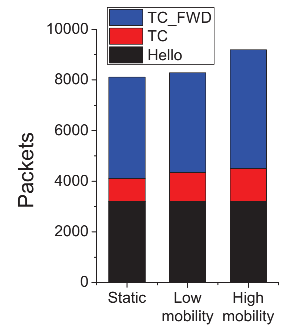
(a) VBPR



(b) DSDV



(c) AODV



(d) OLSR

Fig. 8. Routing control overhead of routing protocols.

mobility, it often takes long to find an alternative path. OLSR performs better than AODV and similarly to DSDV in static or low-mobility scenario, but it achieves the lowest performance in high-mobility scenario. When nodes are highly mobile, frequent topology changes makes OLSR vulnerable to stale topology information.

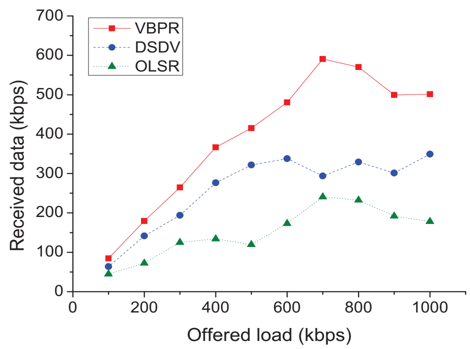
Fig. 7 shows the average end-to-end delay of delivered packets. The delay is reduced when data packets follow the shortest path to the base station. VBPR achieves better delay performance than the other routing protocols except OLSR. Note that the average delay is calculated over the delivered packets. OLSR has a smaller number of delivered packets than VBPR. Since OLSR calculates the shortest path based on the entire topology information, data packets arrives at the base station quickly if the path exists, or they will be discarded if the path disappears due to node mobility. Interestingly, VBPR achieves good delay performance comparable to OLSR.

Under AODV, a route is likely to be determined as the one that the routing request message follows while it travels toward the base station or an intermediate node with cached route information. Unfortunately, it commonly occurs that the first-arrived request message does not follow the shortest path in mobility scenarios, which causes a large end-to-end delay under AODV. It would be possible to improve the performance by adopting additional route learning period before sending the response, which, however, also will contribute to the end-to-end delay. Under DSDV, we have observed that a large number of packets experience long delay waiting for ARP (Address Resolution Protocol) completion, which seems to be due to its excessive number of control messages (which will be explained in the next experiment).

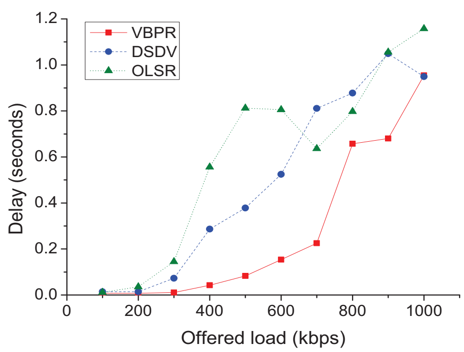
We now compare the routing overhead. Although we called all routing messages as Hello message so far, there are different types and names of routing messages depending on the routing protocols, e.g., topology control (TC) message in OLSR, route request (RREQ) in AODV, and update message in DSDV. They also have different sizes. For example, in our simulation settings, AODV has a smaller message size of 10-30 bytes, DSDV has message sizes of about 360 bytes (periodic update) or 28 bytes (triggered update), and OLSR has message sizes of about 100 bytes (Hello) or 300 bytes (TC). For VBPR, we use 325-byte[6] Hello message (or Hello Message Trigger). On the other hand, we use the IEEE 802.11a for the physical and MAC layers. Since it supports up to 54 Mbps, the 100-byte message can be transmitted in 15 usec. Considering the fact that the slot time of the IEEE 802.11a is 9 usec, the routing control overhead would depend more on the MAC contention time for control messages rather than their transmission times. Therefore, we demonstrate the number of routing messages to compare the routing overhead.

Fig. 8 shows the routing control overhead of routing protocols. All the routing protocols generate periodic Hello (or Update in DSDV) message for every 3 seconds, resulting in about 3200 messages during the simulation time. From Fig. 8(a) and Fig. 8(c), we can observe that VBPR and AODV generates moderate numbers of routing messages as the mobility increases in order to respond to topology changes. Under OLSR, a lot of TC messages (more than the number of periodic Hello messages) are transmitted and result in higher routing overhead as shown in Fig. 8(d). Fig. 8(b) shows that DSDV experiences the worst performance in terms of control overhead. It has generated a similar number of periodic update message as VBPR, and in addition, a huge number of triggered updates ($>$120,000 packets). This is because, once a node generates an update message at a topology change, neighboring nodes update their routing table entries by changing either

[6]We have used a larger size message by padding for fair comparison with other routing schemes and to avoid the links with weak signal strength [19].



(a) Packet delivery rate



(b) Average end-to-end delay

Fig. 9. Performance of routing schemes under different traffic loads.

the sequence number or the number of hop counts, which, in turn, causes them to generate an update message, leading to an enormous number of update message transmissions. The EnableRouteAggregation option is not much helpful and can reduce the number of update messages only by about 10%.

We compare the routing schemes under heavy traffic load, focusing on the capability of dynamic routing. To see this, we consider a scenario where the nodes are static and a source node generates the data traffic in an increasing manner. Fig. 5 shows the locations of the source node and the base station. There are multiple paths from the source node to the base station, which take 5 to 7 hops. The source generates constant-rate data traffic for 300 seconds after an initial period of 20 seconds. We measure packet delivery rate and average end-to-end delay.

Figs. 9(a) and 9(b) show the performance of the routing schemes except AODV, which fails to find a valid path. It is clear that VBPR outperforms both DSDV and OLSR in terms of packet delivery rate and average end-to-end delay. DSDV and OLSR find the shortest path based on the hop count, and keep using the path regardless of traffic loads. As the traffic load increases, some intermediate nodes build up their data queues, which is caused by both high-level interference between neighboring wireless links and excessive packet injection beyond the link capacity. Such large queue lengths result in frequent packet loss as inferred from Fig. 9(a) and high delay as shown in Fig. 9(b).

On the other hand, VBPR does direct data packets dynamically according to the queue length and successfully balance the traffic load over multiple paths. Note that when the traffic load is low, most of the data traffic goes through the shortest path according to the virtual back-pressure. As the traffic load increases (up to 700 kbps), VBPR tries to utilize alternative paths and avoids traversing congested intermediate nodes. Such efforts for dynamic routing are rewarded by higher packet delivery rate and lower average delay compared to DSDV and OLSR.

VI. Conclusion

In this paper, we propose the virtual back-pressure routing protocol (VBPR) for upward traffic service in disaster networks. It has been shown that VBPR exploits the benefits of the original back-pressure (BP) scheme such as provable performance optimality and dynamic routing capability, and it also overcomes the limitation of BP on the delay performance by generating the virtual back-pressure. We address the issues on VBPR implementation, aiming to support high mobility and quick deployment in disaster areas. Through extensive simulations, we evaluate VBPR and show that it achieves high packet delivery ratio, low end-to-end delay, and low routing overhead in a wide range of node mobility. The comparison with other MANET routing schemes also demonstrates that VBPR significantly outperforms AODV, DSDV, and OLSR.

Currently, VBPR works for a single destination or multiple destinations as anycast. The extension to all-destination-pair routing while maintaining scalability is an interesting open problem. Also, the impact of λ and Δ on dynamic routing behaviors and system stability needs further investigation and remains as future work.

References

[1] H. Luo, et al. "UCAN: A Unified Cellular and Ad-Hoc Network Architecture," in *Proc. of ACM MobiCom*, PP. 353–367, 2003.

[2] K. Doppler, et al. "Device-to-Device Communication as an Underlay to LTE-Advanced Networks," *IEEE Communications Magazine*, vol. 47, issue. 12, pp. 42–49, Dec. 2009.

[3] B.S Manoj and A. H. Baker, "Communication Challenges in Emergency Response," *Communications of the ACM*, vol. 50, issue. 3, 2007.

[4] R. Bruno, M. Conti, and A. Passarella, "Opportunistic Networking Overlays for ICT Services in Crisis Management," in *Proc. of the 5th International ISCRAM Conference*, 2008.

[5] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR:The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks," *Ad hoc networking, 5*, pp. 139–172, 2001.

[6] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers,"*ACM SIGCOMM Computer Communications Review*, vol. 24, no. 4, pp. 234-244, 1994.

[7] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100, 1999.

[8] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," in *Proc. of IEEE INFOCOM*, vol. 3, pp. 1405–1413, 1997.

[9] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626, IETF Network Working Group, October 2003.

[10] J. Broch, et al., "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," in *Proc. of ACM MobiCom*, 1998.

[11] C. E. Perkins, E. M. Royer,, S. R. Das, and M. K. Marina, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," *IEEE Personal Communications,* vol. 8, no. 1, 2001.

[12] M. Tarique, K. E. Tepe, S. Adibi, and S. erfani, "Survey of multipath routing protocols for mobile ad hoc networks," *Journal of Network and Computer Applications*, vol. 32, no. 6, pp.1125–1143, 2009.

[13] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks," *IEEE Transactions on Automatic Control*, vol. 37, no.12, pp. 1936–1948, 1992.

[14] X. Lin, N. B. Shroff, and R. Srikant, "A Tutorial on Cross-Layer Optimization Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1452–1463, 2006.

[15] B. Ji, C. Joo, and N. B. Shroff, "Delay-Based Back-Pressure Scheduling Multi-hop Wireless Networks", *IEEE/ACM Transactions on Networking*, to appear, 2013.

[16] E. Athanasopoulou, et al., "Backpressure-based Packet-by-Packet Adaptive Routing in Communication Networks," *IEEE/ACM Transactions on Networking*, vol. 21, no. 1, Feb. 2013.

[17] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing Without Routes: The Backpressure Collection Protocol," in *Proc. of ACM/IEEE IPSN*, pp. 279-290, ACM, 2010.

[18] L. Ying, S. Shakkottai, A. Reddy, and L. Liu, "On Combining Shortest-Path and Back-Pressure Routing Over Multihop Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, 2011.

[19] H. Lundgren, E. Nordstrom, and C. Tschundin, "The gray zone problem in IEEE 802.11 b based ad hoc networks," *ACM SIGMOBILE mobile computing and communications review*, vol. 6, no. 3, pp.104–105, 2002.

[20] D. Kang, C. Joo, S. Kang, and S. Bahk, "Virtual Back-Pressure Routing in Mobile Ad-Hoc Networks for Disaster Scenarios," UNIST, Tech. Rep., 2013, available at http://netlab.unist.ac.kr/ cjoo/infocom14tr.pdf

[21] H. Narra, et al. "Destination-sequenced distance vector (DSDV) routing protocol implementation in ns-3," in *Proc. of the 4th International ICST Conference on Simulation Tools and Techniques. ICST*, 2011.

[22] The ns-3 network simulator. **http://www.nsnam.org,** 2012.

[23] S. Asmussen, Applied Probability and Queues. New York: Wiley, 1987.