Opportunistic Routing With Congestion Diversity in Wireless Ad Hoc Networks

S Farhana Banu & G.S Udaya Kiran Babu
M.Tech Student, Department of CSE, Bheema institute of technology and science,
Adoni , India
Associate Professor, Dept. of CSE, Bheema institute of technology and science, Adoni , India

Abstract

Considering the problem across a multi-hop network in routing a packet that is it consists of numerous sources of traffic and multiple links at the same time giving bounded expected delay. A random list which is subset of all nodes of receiver nodes is assigned for each packet transmission among which the next relay node is choose opportunistically. The problematic part in the construction of minimum-delay routing techniques is maintaining the balance between routing the packets along the shortest paths to the destination and distributing the traffic as per the maximum backpressure .Using important element of backpressure routing and shortest path, this project provides a planned development of D-ORCD. Distributed-opportunistic routing with congestion diversity (D-ORCD) make use of criteria of draining time to opportunistically select and route the packets along the paths to destination with a possible minimum overall congestion. Distributed-ORCD with single goal is proved to give a bounded expected delay for all networks and under any reasonable traffic.

Index Terms: Congestion measure, implementation, Lyapunov analysis, opportunistic routing, queuing stability, wireless ad hoc networks.

INTRODUCTION

OPPORTUNISTIC routing for multi-hop wireless ad hoc networks has long been proposed to overcome the deficiencies of conventional routing . Opportunistic routing mitigates the impact of poor wireless links by exploiting the broadcast nature of wireless transmissions and the path diversity. More precisely, the opportunistic routing decisions are made in an online manner by choosing the next relay based on the actual transmission outcomes as well as a rank ordering of neighboring nodes.. In particular, it is shown that for any packet, the optimal routing decision, in the sense of minimum cost or hop-count, is to select the next relay node based on an index. This index is equal to the expected cost or hop-count of relaying the packet along the least costly or the shortest feasible path to the destination.

When multiple streams of packets are to traverse the net- work, however, it might be desirable to route some packets along longer or more costly paths, if these paths eventually lead to links that are less congested. the opportunistic routing schemes in can potentially cause severe congestion and unbounded delay In contrast, it is known that an opportunistic variant of backpressure diversity backpressure routing (DIVBAR) ensures bounded expected total backlog for all stabilizable arrival rates. To ensure throughput optimality (bounded expected total backlog for all stabilizable arrival rates), backpressure-based algorithms do something very different from rather than using any metric of closeness (or cost) to the destination, they choose the receiver with the largest positive differential backlog (routing responsibility is retained by the transmitter if no such receiver exists). This very property of ignoring the cost to the destination, however, becomes the bane of this approach, leading to poor delay performance in low to moderate Recognizing the shortcomings of the two approaches, re-searchers have begun to propose solutions which combine elements of shortest path and backpressure computations E-DIVBAR is proposed: when choosing the next relay among the set of potential forwarders, E-DIVBAR considers the sum of the differential backlog and the expected hop-count to the destination (also known as ETX). However, as shown in E-DIVBAR does not necessarily result in a better delay performance than DIVBAR. The main contribution of this paper is to provide a distributed opportunistic routing policy with congestion diversity (D-ORCD) under which, instead of a simple addition used in E-DIVBAR, the congestion information is integrated with the distributed shortest path computations of

- we prove that D-ORCD is throughput optimal when there is a single destination (single commodity) and the network operates in stationary regime. While characterizing delay performance is often not analytically tractable, many variants of backpressure algorithm are known to achieve throughput optimality. We show that a similar analytic guarantee can be obtained regarding the throughput optimality of D-ORCD. In particular, we prove the throughput optimality of D-ORCD by looking at the convergence of D-ORCD to a centralized version of the algorithm. The optimality of the centralized solution is established via a class of Lyapunov functions proposed in

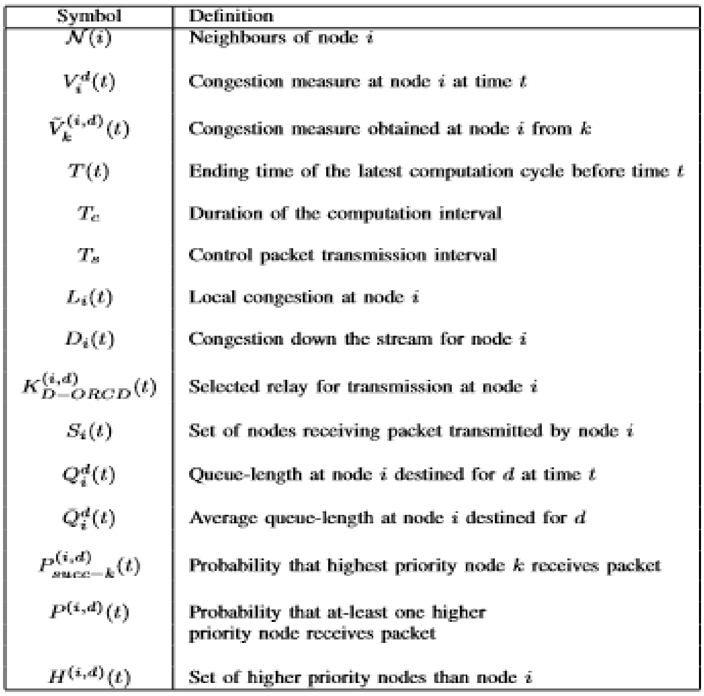
II. OPPORTUNISTIC ROUTING WITH CONGESTION DIVERSITY

The goal of this paper is to design a routing policy with im-proved delay performance over existing opportunistic routing policies. In this section, we describe the guiding principle behind the design of Distributed Opportunistic Routing with Congestion Diversity (D-ORCD). We propose a time-varying distance vector, which enables the network to route packets through a neighbor with the least estimated delivery time.

D-ORCD opportunistically routes a packet using three stages of: (a) transmission, (b) acknowledgment, and (c) relaying. During the transmission stage, a node transmits a packet. During the acknowledgment stage, each node that has success-fully received the transmitted packet, sends an acknowledgment (ACK) to the transmitter node. D-ORCD then takes routing decisions based on a congestion-aware distance vector metric, referred to as the congestion measure. More specifically, during the relaying stage, the relaying responsibility of the packet is shifted to a node with the least congestion measure among the ones that have received the packet.

The congestion measure of a node associated with a given destination provides an estimate of the best possible draining time of a packet arriving at that node until it reaches destination. Each node is responsible to update its congestion measure and transmit this information to its neighbors. Next, we detail D-ORCD design and the computations performed at each node to update the congestion measure.

TABLE I
NOTATIONS USED IN THE DESCRIPTION OF THE ALGORITHM



Symbol	Definition
$N(i)$	Neighbours of node i
$V_i^d(t)$	Congestion measure at node i at time t
$\tilde{V}_k^{(i,d)}(t)$	Congestion measure obtained at node i from k
$T(t)$	Ending time of the latest computation cycle before time t
T_c	Duration of the computation interval
T_s	Control packet transmission interval
$L_i(t)$	Local congestion at node i
$D_i(t)$	Congestion down the stream for node i
$K_{D-ORCD}^{(i,d)}(t)$	Selected relay for transmission at node i
$S_i(t)$	Set of nodes receiving packet transmitted by node i
$Q_i^d(t)$	Queue-length at node i destined for d at time t
$Q_i^d(t)$	Average queue-length at node i destined for d
$P_{succ-k}^{(i,d)}(t)$	Probability that highest priority node k receives packet
$P^{(i,d)}(t)$	Probability that at-least one higher priority node receives packet
$H^{(i,d)}(t)$	Set of higher priority nodes than node i

A. D-ORCD Design

We consider a network of D nodes labeled by $\Omega -$ $\{1\_D\}$. We characterize the behavior of the wireless channel using a probabilistic transmission model. Node j is said to be neighbor of node i, if there is a positive probability P_{ij} that a transmission at node i is received at node j. The set of all nodes in the network which are reachable by node is referred to as neighborhood of node and is denoted by

D-ORCD relies on a routing table at each node to determine the next best hop. The routing table at node i consists of a list of neighbors and a structure consisting of estimated congestion measure for all neighbors in associated with different destinations. The routing table acts as a storage and decision component at the routing layer. The routing table is up-dated using a "virtual routing table" at the end of every "computation cycle": an interval of T units of time. To update virtual routing table, during the progression of the computation cycle the nodes exchange and compute the temporary congestion measures. The temporary congestion measures are computed in a fashion similar to a distributed stochastic routing computation of [4] using the backlog information at the beginning of the computation cycle (generalizing the computations of distributed Bellman-Ford). We conceptualize this in terms of the virtual routing table updating and maintaining these temporary congestion measures. We assume that each node has access to a common global time to ensure that the nodes update the routing table roughly at the same time.

at node i (denoted by $L_i(t)$) and the draining time from its next Hop to the destination (denoted by $D_i(t)$), i.e.,

$$V_i^d(t) = L_i^d(t) + D_i^d(t). \tag{1}$$

In order to compute $L_i(t)$ and $D_i(t)$, node i relies on the following quantities:

- p_{ij} : Probability that a packet transmitted by node i is received by node j.

•$S_i(t)$: Set of nodes that received packet transmitted by node i at time t.
•$Q_i(t)$: Average number of packets at node i destined for d
During the last computation cycle, i.e., $Q_i(t)$ is updated as $\bar{Q}_i^d(t) = \frac{T_s}{T_c} \sum_{l=0}^{\frac{T_c}{T_s}-1} Q_i^d(T(t)-l)$.

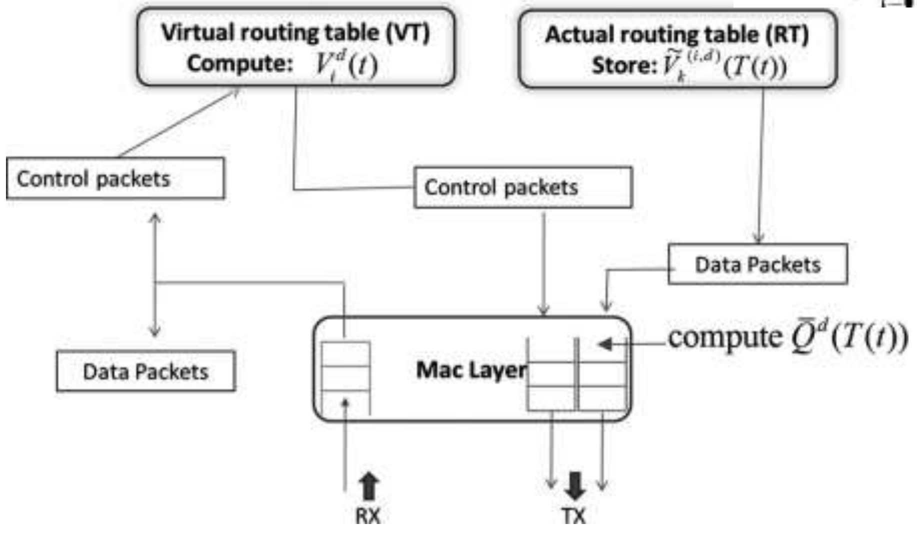

Virtual routing table (VT)
Compute: $V_i^d(t)$
Actual routing table (RT)
Store: $\widetilde{V}_k^{(i,d)}(T(t))$
Control packets
Control packets
Data Packets
Data Packets
Mac Layer
compute $\bar{Q}^d(T(t))$
RX
TX

Fig.1. Operation of D-ORCD.

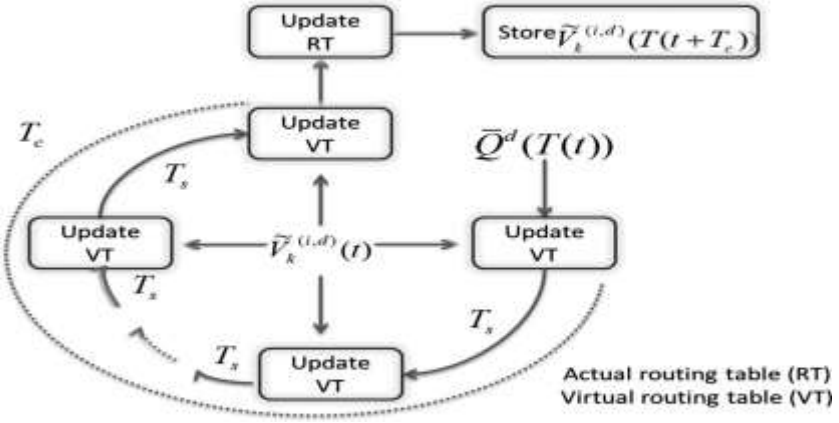

Update RT
Store $\widetilde{V}_k^{(i,d)}(T(t+T_c))$
T_c
Update VT
T_s
$\bar{Q}^d(T(t))$
Update VT
$\widetilde{V}_k^{(i,d)}(t)$
Update VT
T_s
T_s
T_s
Update VT
Actual routing table (RT)
Virtual routing table (VT)

Fig. 2. Actual routing table is updated every T_a units of time while virtual routing table is updated after receiving any control packet.

- $H_k^{(i,d)}(t)$: Neighbors of node i whose perceived congestion measures are smaller than that of node k, i.e., $H_k^{(i,d)}(t) = \left\{ j \in \mathcal{N}(i) : \tilde{V}_j^{(i,d)}(t) < \tilde{V}_k^{(i,d)}(t) \right\}$

- $H^{(i,d)}(t)$: Neighbors of node i which are used by node i to route packets destined for node d, i.e.,

$$H^{(i,d)}(t) = \begin{cases} H_i^{(i,d)}(t) & \text{if } H_i^{(i,d)}(t) \neq \emptyset \\ \underset{j \in \mathcal{N}(i) \quad \{i\}}{\arg\min} \ \tilde{V}_j^{(i,d)}(t) & \text{if } H_i^{(i,d)}(t) = \emptyset \end{cases} \quad (2)$$

We denote the temporary congestion measure associated with node at time and destination as Each node computes based on congestion measures obtained via periodic communication with its neighbors $k \in \mathcal{N}(i)$ and the queue backlog at the start of the computation cycle. D-ORCD stores these temporary congestion measures congestion measure and subsequently advertises it to its neighbors using control packets at intervals of $T_v < T_a$ seconds. Finally the actual routing table is updated using the entries in the virtual routing table after every T_a second. The sequence of operations performed by D-ORCD is shown in Figs. 1 and 2.

Meanwhile, for routing decisions, node i uses the entries in the actual routing table (updated at the end of the last computing)
$\tilde{V}_k^{(i,d)}(T(t))$.

Next, we describe the distributed computations performed during each computation cycle.

B. Congestion Measure Computations

The congestion measure associated with node i for a destination d at time t is the aggregate sum of the local draining time

- $P_k^{(i,d)}(t)$: Probability that node k is selected as the next
  Relay for a packet destined for node d and transmitted by node i, i.e..

$$P^{(i,d)}_{\mathrm{succ},k}(t) - p_{ik} \prod_{j \in H_k^{(i,d)}(t)} (1 \quad p_{ij}). \quad (3)$$

  Note that, among the multiple receivers of the packet node k is selected as the next hop, if and only if none of the other
  higher priority nodes $H_k^{(i,d)}(t)$ received the packet.

- $P_{\mathrm{succ}}^{(i,d)}(t)$: Probability that a packet transmitted by node i and destined for node d progresses towards the destination,

$$P_{\mathrm{succ}}^{(i,d)}(t) = \sum_{k \in H^{(i,d)}(t)} P^{(i,d)}_{\mathrm{succ},k}(t). \quad (4)$$

With these parameters and assuming a FIFO discipline at layer-2, we proceed with the relay selection rule. In particular, Next we provide the computation details of L_i^d and D_i^d. The local draining time, L_i^d, relies on the fact that, when a packet arrives at a node i, its waiting time is equal to the time spent in draining the packets that have arrived earlier plus its own transmission time. Noting that the expected transmission time.

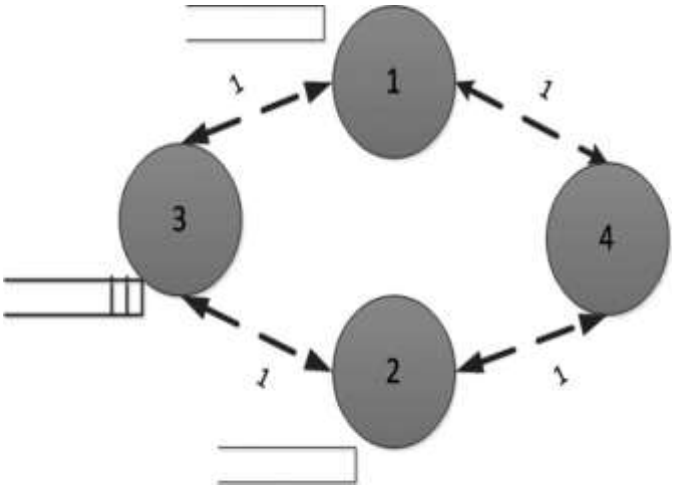


Fig. 3. Topology for the illustrative example.

Node 4 is the destination and node 3 is the source node.

At node i for the packet can then be approximated by the local draining time for node i to destination d at time t is

$$L_i^d(t) = \frac{1}{P_{\mathrm{succ}}^{(i,d)}(t)} + \sum_{d' \in \Omega} \frac{Q_i^{d'}(T(t))}{P_{\mathrm{succ}}^{(i,d')}(t)}. \qquad (5)$$

D-ORCD computes the expected congestion measure "down The stream", $D_i^d(t)$ for each node $i \in \Omega$ using the latest congestion measures $\tilde{V}_k^{(i,d)}(t)$ received from nodes $k \in \Omega$ with lower congestion measure. More specifically, the expected congestion "Down the stream" $D_i^d(t)$ can be given as

$$D_i^d(t) = \frac{1}{P_{\mathrm{succ}}^{(i,d)}(t)} \sum_{k \in H^{(i,d)}(t)} P_{\mathrm{succ}\ k}^{(i,d)}(t) \tilde{V}_k^{(i,d)}(t). \qquad (6)$$

Remark 1: In each computation cycle, assuming T_c is large D-ORCD computations converge to the Bellman equation associated with the minimum cost ("shortest path") route in a net-work, where the nodel costs are given in terms of the queue Length $Q_i(t)$.

C. Illustrative Example

We describe an example to illustrate the detailed design of D-ORCD. Fig. 3 shows a 4-node topology where packets from source node 3 are destined for node 4. At any time instant, node 3 chooses either node 1 or 2 as the next hop, based on the queue lengths at nodes 1 and 2. D-ORCD updates the congestion measures every $T_c=1$ unit, while the routing table is updated after every $T=3$ units. We assume that at time 0, $Q_1(0)=0, Q_2(0)=1, Q_3(0)=8$, while the congestion measures in the actual routing table are initialized as $\tilde{V}_{3,4}(0)=0, \tilde{V}_{3,4}(0)=1$. Fig. 4 shows the various parameters (i) actual congestion measure for node 1, (ii) actual congestion mea-sure for node 2, (iii) virtual congestion measure for node 1,

(iv)Virtual congestion measure for node 2, (v) queue lengths for various nodes, and (vi) next hop at node 3, as time progresses. The table also shows that the effect of the delayed congestion information causes lags in the routing decisions. Note that D-ORCD routing decisions lag by T , as actual routing table update is delayed.

In the next section, we discuss the practical issues associated with computation of the time-varying congestion measures $V_i^d(t), i \in \Omega$.

PROTOCOL IMPLEMENTATION DETAILS

In this section, we discuss the implementation issues of D-ORCD, and in particular, distributed and asynchronous iterative computations of $V_i(t)$'s. We provide a brief discussion of the basic challenges of D-ORCD including the three-way handshake procedure employed at the MAC layer, link quality estimation, avoidance of loops while routing, and overhead reduction issues.

A. Compatible Implementation

1) Three-Way Handshake: The implementation of D-ORCD, analogous to any opportunistic routing scheme, involves the se-lection of a relay node among the candidate set of nodes that have received and acknowledged a packet successfully. One of the major challenges in the implementation of an opportunistic routing algorithm, in general, and D-ORCD in particular, is the design of an 802.11 compatible acknowledgement mechanism at the MAC layer. Below we propose a practical and simple way to implement acknowledgement architecture.

The transmission at any node i is done according to 802.11 CSMA/CA mechanisms. Specially, before any transmission, transmitter i performs channel sensing and starts transmission after the back off counter is decremented to zero. For each

neighbor node $j \in N(i)$, the transmitter node i then reserves virtual time slot of duration $T_{ACK}+T_{SIFS}$, where T_{ACK} is the duration of the acknowledgement packet and T_{SIFS} is the duration of Short Inter Frame Space (SIFS) . Transmitter i then piggy-backs a priority ordering of nodes $N(i)$ with each data packet transmitted. The priority ordering determines the virtual time slot in which the candidate nodes transmit their acknowledgement. Nodes in the set S_i that have successfully received the packet then transmit acknowledgement packets sequentially in the order determined by the transmitter node.

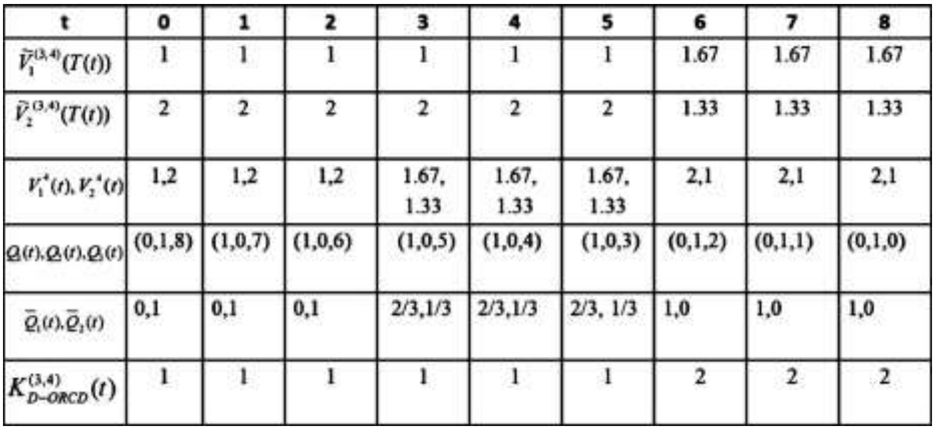
After a waiting time of $T_{fill}=|N(i)|(T_{ACK}+T_{SIFS})$ during which each node in the set S_i has had a chance to send an ACK, node i transmits a forwarding control packet (FO). The FO packets contain the identity of the next forwarder, which may be node i itself (i.e., node i retains the packet) or any node $j \in S_i$.

If T_{fill} expires and no FO packet is received (FO packet reception is unsuccessful), then the corresponding candidate nodes drop the received data packet. If transmitter i does not receive any acknowledgement, it retransmits the packet. The backoff window is doubled after every retransmission. Furthermore, the packet is dropped if the retry limit (set to 7) is reached.

B. Reliability of Control Packets

The implementation and design of D-ORCD depend on a re-liable, frequent, and timely delivery of the control packets. As documented in, the loss of routing layer control packets may destabilize the algorithm operation and cause significant performance degradation for many well-known routing algorithms. In our implementation, we have taken advantage of the priority-based queuing; D-ORCD prioritizes the control packets by assigning them the highest strict priority, reducing the probability that the packets are dropped at the MAC layer and also ensuring a timely delivery of the control packets. Moreover, D-ORCD scheduler assigns a sufficiently lower PHY rate for the control packets.

The reliability of MAC layer FO packets is another important factor affecting the performance of D-ORCD; as a result, FO packets are transmitted at the lower rate of 1 Mbps.



t	0	1	2	3	4	5	6	7	8
$\tilde{V}_1^{(3,4)}(T(t))$	1	1	1	1	1	1	1.67	1.67	1.67
$\tilde{V}_2^{(3,4)}(T(t))$	2	2	2	2	2	2	1.33	1.33	1.33
$V_1^*(t), V_2^*(t)$	1,2	1,2	1,2	1.67, 1.33	1.67, 1.33	1.67, 1.33	2,1	2,1	2,1
$Q_1(t), Q_2(t), Q_3(t)$	(0,1,8)	(1,0,7)	(1,0,6)	(1,0,5)	(1,0,4)	(1,0,3)	(0,1,2)	(0,1,1)	(0,1,0)
$\bar{Q}_1(t), \bar{Q}_2(t)$	0,1	0,1	0,1	2/3,1/3	2/3,1/3	2/3, 1/3	1,0	1,0	1,0
$K_{D-ORCD}^{(3,4)}(t)$	1	1	1	1	1	1	2	2	2

Fig. 4. Table indicating (i) actual congestion measure for node 1, (ii) actual congestion measure for node 2, (iii) virtual congestion measure for node 1, (iv) virtual congestion measure for node 2, (v) queue lengths for various nodes, and (vi) next hop at node 3, as the time is varied. $T_{2\to1}/T_{2\to3}$.

C. Link Quality Estimation Protocol

D-ORCD computations given by (1) utilize link success probabilities $p_{i,j}$ for each pair of nodes i,j. We now describe a method to determine the probability of successfully receiving a data packet for each pair of nodes i,j∈Ω. Our method consists of two components: active probing and passive probing. In the active probing, dedicated probe packets are broadcasted periodically to estimate link success probabilities. In passive probing, the overhearing capability of the wireless medium are utilized. The nodes are configured to promiscuous mode, hence enabling them to hear the packets from neighbors. In passive probing, the MAC layer keeps track of the number of packets received from the neighbors including the retransmissions. Finally, a weighted average is used to combine the active and passive estimates to determine the link success probabilities. Passive probing does not introduce any additional overhead cost but can be slow, while active probing rate is set independently of the data rate but introduces costly overhead.

D. Loop Avoidance Heuristic

D-ORCD approximates the solution to the fixed point equation via a distributed distance vector approach. The classical problem of counting to infinity in distance vector routing can affect D-ORCD performance due to the time varying nature of the congestion metric. The problem is most acute when there is a sudden burst of traffic1 and can cause severe transient effects due to slow updates of the control packets. The looping results in large delays, increased interference, and loss of packets.

To address this issue, in our experiments we utilize an ex-tension of the Split-horizon with poison reverse solution to avoid loops. In Split-horizon with poison reverse, a node advertises routes as unreachable to the node through which they were learned. Intuitively, this method penalizes the routes with loops and removes them from the set of available options. We have extended the rule to D-ORCD by advertising the routes as unreachable to higher ranked nodes. Even though the proposed solution is not provably loop-free in eliminating the routes with loops, it significantly reduces the packets stuck in routes with loops.

CONCLUSION:

In this paper, we provided a distributed opportunistic routing policy with congestion diversity (D-ORCD) by combining the important aspects of shortest path routing with those of backpressure routing. Under this policy packets are routed according to a rank ordering of the nodes based on a congestion measure. Furthermore, we proposed a practical distributed and asynchronous 802.11 compatible implementation of D-ORCD, whose performance was investigated via a detailed set of QualNet simulations for practical and realistic networks.

In D-ORCD, we do not model the interference from the nodes in the network, but instead leave that issue to a classical MAC operation.. In future, we are interested in generalizing D-ORCD for joint routing and scheduling optimizations as well considering the system-level implications. Incorporating throughput optimal CSMA based MAC scheduler with congestion aware routing is also promising area of research. The design of D-ORCD requires knowledge of channel statistics. Designing congestion control routing algorithms to minimize expected delay without the topology and the channel statistics knowledge is an area of future research.

REFERENCES

[1] P. Larsson, “Selection diversity forwarding in a multihop packet radio network with fading channel and capture,” ACM SIGMOBILE Mobile Comput. Commun. Rev., vol. 5, no. 4, pp. 47–54, Oct. 2001.

[2] M. Zorzi and R. R. Rao, “Geographic random forwarding (GeRaF) for Ad Hoc and sensor networks: Multihop performance,” IEEE Trans. Mobile Comput., vol. 2, no. 4, 2003.

[3] S. Biswas and R. Morris, “ExOR: Opportunistic multi-hop routing for wireless networks,” ACM SIGCOMMComput. Commun. Rev., vol. 35, pp. 33–44, Oct. 2005.

[4] C. Lott and D. Teneketzis, “Stochastic routing in ad hoc networks,” IEEE Trans. Autom. Contr., vol. 51, pp. 52–72, Jan. 2006.

[5] S. Jain and S. R. Das, “Exploiting path diversity in the link layer in wireless ad hoc networks,” in Proc. WoWMoM, 2005, pp. 22–30.

[6] P. Gupta and T. Javidi, “Towards throughput and delay optimal routing for wireless ad hoc networks,” in Proc. Asilomar Conf., 2007, pp. 249–254.

[7] M. J. Neely and R. Urgaonkar, “Optimal backpressure routing for wireless networks with multi-receiver diversity,” Ad Hoc Networks, vol. 7, no. 5, pp. 862–881, Jul. 2009.

[8] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” IEEE Trans. Autom. Contr., vol. 37, no. 12, pp. 1936–1949, Aug. 1992.

[9] S. Sarkar and S. Ray, “Arbitrary throughput versus complexity tradeoffs in wireless networks using graph partitioning,” IEEE Trans. Autom. Contr., vol. 53, no. 10, pp. 2307–2323, Nov. 2008.

[10] Y. Xi and E. M. Yeh, “Throughput optimal distributed control of stochastic wireless networks,” in Proc. 4th Int. Symp. Modeling and

Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2006, pp. 1–10.

[11] B. Smith and B. Hassibi, "Wireless erasure networks with feedback," arXiv: 0804.4298v1, 2008.

[12] Y. Yi and S. Shakkottai, "Hop-by-hop congestion control over a wireless multi-hop network," IEEE/ACM Trans. Netw., vol. 15, no. 1, pp. 133–144, Feb. 2007.

[13] L. Huang, S. Moeller, M. J. Neely, and B. Krishnamachari, "LIFObackpressure achieves near optimal utility-delay tradeoff," in Proc. 9th Int. Symp.Modeling and Optimization inMobile, Ad Hoc, andWireless Networks (WiOpt), 2011, pp. 70–77.