

An Interference Avoidance Routing Protocol for Wireless Networks

Bai Du, Hongyan Li

State Key Lab of ISN, Xidian University
Xi'an, China
du198614@163.com, hyl@xidian.edu.cn

Yuguang Fang

Department of Electrical and Computer Engineering,
University of Florida
FL 32611, USA
fang@ece.gufl.edu

Abstract—In wireless networks, interference between nodes is an important factor which affects the performance of communications. In this paper, we propose an interference avoidance routing protocol (IARP) that chooses a route with fewer collisions and show that the new routing algorithm is stable. The idea is to collect the busyness information about nodes, and use this to determine the next hop based on the backpressure routing algorithm. From the simulation results, we observe that IARP can achieve better performance in terms of network delay than the pure backpressure routing protocol.

Keywords—Wireless networks; Backpressure routing; Interference avoidance; Network stability.

I. INTRODUCTION

In a wireless network, if nodes share the same wireless channel, they may interfere with each other, which will cause more retransmissions and packet loss. If nodes are busy or the node density is high, it may take a long time for a node to compete for the channel successfully when the contention-based medium access control (MAC) protocols such as CSMA/CA is used. How to develop efficient routing algorithms in this network environment is challenging and has been studied intensively.

Zhang et al [1] proposed a routing algorithm in which they first compute the average link-interference between the source and destination, and then choose a path with the minimum interference, assuming that nodes know some routes to the destination before computing the interference and also know the information about the interference along all the routes. Lin et al [2] improved the throughput for wireless ad hoc networks by modeling the interference by means of interference graph and transmitting busy tone with encoded communication information to prevent the potentially interfering nodes from initiating new transmissions. The idea that the interference can be avoided based on the received signal of common packet during the preceding hops was originally proposed in [3], in which decode-and-cancel protocol and amplify-and-cancel protocol are proposed to estimate and cancel the interference. Liu et al [6] designed a decoding method which makes use of the existing correlation between the interference and the previously received data to mitigate the interference and

achieve the improved performance. Most strategies in the aforementioned papers use some parameters to choose a route with less interference or mitigate the interference in a chosen route. In [4] and [5], algorithms were developed to find the route with less interference directly. Liu et al [4] proposed a new metric quantifying the degree of interference of the entire network, and then obtaining the interference-minimizing network topology while preserving the connectivity of ad hoc networks. Mohanoor et al [5] developed an algorithm to find multiple paths which may have interference among them, but still achieve the maximum possible throughput. This indicates that the pattern of interference is more important than the number of interferers.

Notice that all the aforementioned algorithms usually need global information in the sense that a large amount of information exchange across the network. However, it is hard to implement these algorithms in practical wireless networks where nodes can move freely. Especially in mobile ad hoc networks, nodes move frequently and the information such as topology may be outdated quickly. So it will be better to design a distributed routing protocol which just needs local information to mitigate the interference.

In this paper, we propose an interference avoidance routing protocol (IARP) to mitigate the path interference by choosing routes with fewer collisions. For the purpose of analysis, we assume that transmission range and carrier sensing range are the same, which means a node is only interfered by its neighbors. IARP is to check the frequency on which nodes send packets and determine the busyness of every neighboring node. Using this information, nodes can choose a path leading to less loaded parts of the network. However, if the route to the destination is fixed, the busyness information will become useless. Considering this, we design IARP based on the backpressure routing algorithm.

Backpressure routing algorithm [11] is a simple routing algorithm which only utilizes the backlog in the neighbor nodes and chooses the next hop. The routes from source to destination are not fixed and can make use of all the links in the network. Every node determines the next hop by itself. So it is easy for us to incorporate our idea into the backpressure routing algorithm. Neely [11] has proved that the backpressure routing algorithm can maximize the network throughput while

maintaining stability. Here we also want our IARP to retain the network stability while improving the path throughput.

The rest of the paper is organized as follows. Section II describes the algorithm and the corresponding mathematical model. In Section III, we show the network stability by using Lyapunov drift theory. Final section gives the performance analysis.

II. ROUTING ALGORITHM AND ANALYTICAL MODEL

A. Backpressure routing algorithm and its problems

We assume the network operates in a slotted fashion. The data in the network is associated with a particular commodity, which defines the destination of the data. Let $A_i^c(t)$ represent the amount of new commodity c data that exogenously arrives to source node i during slot t . Let $\mu_{ab}^c(t)$ denote the service rate offered to commodity c data over the (a,b) link during slot t . Let $U_i^c(t)$ be the current backlog of commodity c data stored at node i .

In a simple backpressure routing algorithm [11], every node knows the backlog $U_i^c(t)$ in its neighbor nodes. Hence, we can get the differential backlog $W_{ij} = U_i^c(t) - U_j^c(t)$ and define the maximum difference $W_i^* = \max_j (U_i^c(t) - U_j^c(t))$, where j is an arbitrary neighbor of node i . If $W_i^* > 0$, node i will send the data c to the corresponding j that will achieve such W_i^* and transmit at the maximum rate the link can support. When a node finds out one of its neighbors is the destination, its next hop is always the destination. This backpressure routing algorithm does not need the information of the network topology, so it is easy to implement in the distributed fashion. Now consider an example scenario shown in Fig. 2.1

In this scenario, the node S sends packets to node D. Node S will compute W_{s1} and W_{s2} to determine the next hop. If $W_{s1} > W_{s2}$, the next hop is node 1. But if the node 1 is very busy, it will always refuse the packets from node S, and in the

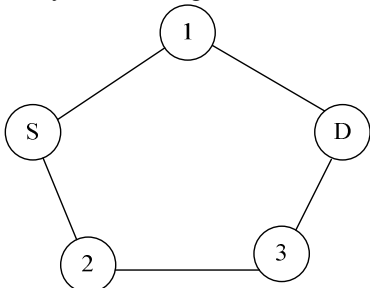


Fig 2.1 An example of interference

next slot node S also want to send packet to node 1. When data in node 2 transmits to node 3, W_{s2} will become small and then $W_{s1} < W_{s2}$. Now node S will choose node 2 to be its next hop. This process may last a long time and the network delay may be large. So we need to collect the information about the busyness of neighboring nodes. In this example, if node S knows that node 1 is busy, it can send data to node 2 first and do not need to wait for a long time to choose the best route. Fig 2.2 gives another example.

In Fig 2.2, if node 1 and node 2 are not busy, but node 4 is busy, the transmission from S to 1 will be interfered, causing the same effect as the busyness at node 1. So the same thing will happen as in Fig 2.2. Observing this, we also need to know the busyness state of node 4, the neighbor's neighbors. By using the information of the busyness of the neighboring nodes and their neighbors, i.e., the two-hop neighbor information, we can design a new efficient routing algorithm called IARP.

B. Description of IARP

We first need a metric to characterize the busyness of nodes. Our purpose is to choose the next hop to send the packets across the less loaded parts of the network. For node i , we define a pair of parameters (m_i, n_i) , where m_i represents the busyness at node i , and n_i is the busyness at the neighbor nodes of node i .

The value of m_i is calculated as follows:

- 1 $m_i = 0$ when the network is initialized.
- 2 we observed for a certain number M of consecutive time slots, if the number of busy slots is k , $m_i = k/M$, i.e., the busyness is the percentage that the slot is busy.

We notice that $m_i < 1$. So m_i is too small to put in the differential backlog in order to be effective. Because of this, we need to scale it up by multiplying a certain number, say, 10. Thus, m_i has upper limit 10 and lower limit 0. We notice that we can choose other scaling factor for the optimal design, which will be studied in the future.

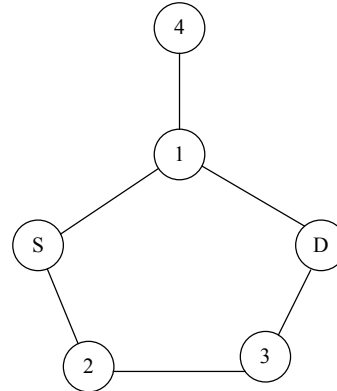


Fig 2.2 An example of interference from neighbor nodes

The value of n_i can be calculated as follows: $n_i = \sum_j m_j$

where j 's are the neighboring nodes of node i . For example, if node a has two neighbors: node b and node c , we have $n_a = m_b + m_c$. Notice that if node density is very high, which means a node has many neighbor nodes. So n_i at the node i will large, and node i will tend to deter further packets to it.

Next, we use the Q-learning algorithm [12] to obtain a new pair of parameters (M_i, N_i) , in order to get a better estimate for the busyness state of a node. If in slot $t-1$ we have the variables $m_i^{t-1}, n_i^{t-1}, M_i^{t-1}, N_i^{t-1}$, then the values of

M_i^t and N_i^t can be calculated as follows:

$$M_i^t = (1 - \alpha)m_i^{t-1} + \alpha(\max_{t-k < x < t} M_i^x - M_i^{x-1}), 0 < \alpha < 1$$

$$N_i^t = (1 - \beta)n_i^{t-1} + \beta(\max_{t-k < x < t} N_i^x - N_i^{x-1}), 0 < \beta < 1$$

where k is the window size for the learning.

In our algorithm, we incorporate this estimated (M_i, N_i) into the calculation of refined backlog W . If node a sends data to node b , (M_a, N_a) is the parameter at node a and U_a is the backlog at node a . Then, the corresponding parameters of b are $(M_b, N_b), U_b$. Then, $W_{ab} = U_a - U_b + M_a - M_b + N_a - N_b$. From this formula, we find if node a is busy and node b is idle, the value of W_{ab} will add a significant part from $U_a - U_b$. Node a will choose node b as the next hop, hence the data will be delivered to the less loaded part of the network.

We note that when node a sends a packet to node b , the busyness at node b will increase. When this value (the busyness) becomes large enough, node a will change the next hop. However, m_b is caused by node a . So when the transmission from a to b is successful at the current slot, in the next slot, we let m_b be zero for node a .

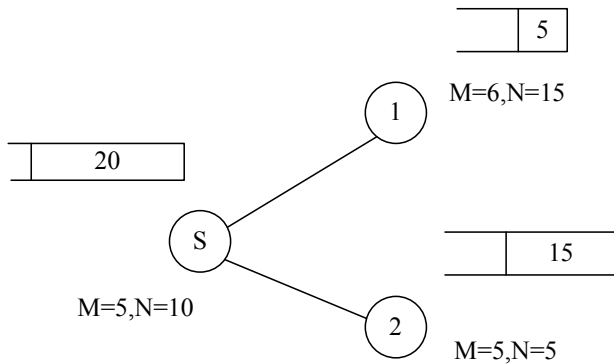


Fig 2.3 An example of next hop selection

Here is a simple example of next hop selection in Fig 2.3. Node S wants to send packets to node 1 or node 2. Parameters are shown in Fig 2.3. So we can get

$$W_{s1} = 20 + 5 + 10 - 5 - 6 - 15 = 9,$$

$$W_{s2} = 20 + 5 + 10 - 15 - 5 - 5 = 10.$$

Note that $W_{s2} > W_{s1}$, $W_s^* = W_{s2}$. The next hop is node 2.

III. THE PROOF OF NETWORK STABILITY

A Definition of Stability and Description of Lyapunov Drift

Here we want to prove the stability of the new algorithm. A queue $U(t)$ is said to be stable if

$$\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E\{U(\tau)\} < \infty$$

We say that a network is stable if all queues in the network are stable.

It is hard to prove network stability directly. Usually, the theory of Lyapunov drift can be used to study the stability of queueing networks and for developing stabilizing control algorithms [9,11]. The idea is to define a Lyapunov function, a non-negative function to measure the aggregate congestion of all queues in the network. We use this function to evaluate how the queue change from one slot to the next.

Specifically, we define a function $L(U) = \sum_{i=1}^L U_i^2$,

where U_i is the backlog in the node i . It is easy to see that L is non-negative and any queue in the network becoming infinite will make L become infinite. We will use the following result to prove the stability.

Lemma[11]: If there exist constants $B > 0, \epsilon > 0$ such that for all timeslots t we have:

$$E\{L(U(t+1)) - L(U(t)) | U(t)\} \leq B - \epsilon \sum_{i=1}^L U_i(t), \quad (1)$$

Then the network is strongly stable, and furthermore:

$$\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{i=1}^L E\{U_i(\tau)\} \leq \frac{B}{\epsilon}, \quad (2)$$

Observing this result, we only need to show inequality (1). From inequality (2), we can then get an upper bound for the backlog. For convenience, the left hand side of (1) can be denoted as $\Delta(U(t))$.

B Stability of IARP

Georgiadis et al [11] has proved the stability of backpressure algorithm. Here, we will show the stability of IARP based on their work. Firstly, we let $L(U) = \sum_{ic} (U_i^{(c)})^2$, where i denotes the node and c denotes the data or commodity. According to [11]:

$$L(U(t+1)) - L(U(t)) \leq 2BN - 2 \sum_{ic} U_i^{(c)}(t) \left[\sum_b \mu_{ib}^c(t) - A_i^{(c)}(t) - \sum_a \mu_{ai}^{(c)}(t) \right], \quad (3)$$

$$\text{Here } B = \frac{1}{2N} \sum_{i \in N} [(\mu_{\max,i}^{\text{out}})^2 + (A_i^{\max} + \mu_{\max,i}^{\text{in}})^2]$$

Taking expectation on both sides of (3)

$$\Delta(U(t)) \leq 2BN + 2 \sum_{ic} U_i^{(c)}(t) E\{A_i^{(c)}(t) | U(t)\} - 2E\left\{ \sum_{ic} U_i^{(c)}(t) \left[\sum_b \mu_{ib}^c(t) - \sum_a \mu_{ai}^{(c)}(t) \right] | U(t) \right\}, \quad (4)$$

which is similar to (1). We now show that we can obtain the same bound needed in the lemma.

From [11], we know if there is a route algorithm leading to a stable network, then there must exist a constant ε

$$E\left\{ \sum_b \tilde{\mu}_{ib}^{(c)}(t) - \sum_a \tilde{\mu}_{ai}^{(c)}(t) | U(t) \right\} = \varepsilon + \lambda_i^{(c)}, \quad (5)$$

$$\text{and } \sum_{ic} U_i^{(c)}(t) \left[\sum_b \tilde{\mu}_{ib}^{(c)}(t) - \sum_a \tilde{\mu}_{ai}^{(c)}(t) \right] \leq, \quad (6)$$

$$\sum_{ic} U_i^{(c)}(t) \left[\sum_b \mu_{ib}^c(t) - \sum_a \mu_{ai}^{(c)}(t) \right]$$

where the left hand side is for an arbitrary algorithm, while the right hand side is for backpressure routing. According to (4), (5) and (6), and thus we can obtain

$$\Delta(U(t)) \leq 2BN - 2\varepsilon \sum_{ic} U_i^{(c)}(t), \quad (7)$$

This is just the form of (1), so we have proved the stability of backpressure routing algorithm.

Now we examine the difference between the backpressure routing algorithm and our new algorithm IARP. In the backpressure algorithm, $W_{ab} = U_a - U_b$, and in IARP, $W_{ab} = U_a - U_b + M_a + N_a - M_b - N_b$. Since M_a, M_b, N_a, N_b are finite, the difference between these two weights has an upper bound denoted as l . Also, both algorithms need to find W^* . At the current slot, if the maximum W^* is W_{old} in the backpressure algorithm, then in IARP the corresponding maximum weight W_{new} must satisfy the following inequality:

$$W_{new} \geq W_{old} - l, \quad (8)$$

According to [11]:

$$\sum_{ic} U_i^{(c)}(t) \left[\sum_b \mu_{ib}^c(t)_{new} - \sum_a \mu_{ai}^{(c)}(t)_{new} \right] = \sum_{ab} \sum_c \mu_{ab}^c(t)_{new} [U_a^{(c)}(t) - U_b^{(c)}(t)], \quad (9)$$

Note that $U_a^{(c)}(t) - U_b^{(c)}(t)$ is just the W_{old} , so we can get

$$\sum_{ic} U_i^{(c)}(t) \left[\sum_b \mu_{ib}^c(t)_{new} - \sum_a \mu_{ai}^{(c)}(t)_{new} \right] \geq \sum_{ab} \sum_c \mu_{ab}^c(t) [U_a^{(c)}(t) - U_b^{(c)}(t)] - \sum_{ab} \sum_c \mu_{ab}^c(t) l, \quad (10)$$

where the left hand side of (10) is for the IARP algorithm and the right hand side is for the backpressure algorithm.

Note that $\sum_{ab} \sum_c \mu_{ab}^c(t) l$ is finite, so we can have

$\sum_{ab} \sum_c \mu_{ab}^c(t) l \leq l_1$, where l_1 is a constant. Then the inequality (7) can be rewritten as follows:

$$\Delta(U(t)) \leq 2BN - 2\varepsilon \sum_{ic} U_i^{(c)}(t) + l_1, \quad (11)$$

The difference here from (1) is just a constant l_1 , which does not affect the stability results. So we have shown IARP still retains the stability of the network.

IV. SIMULATIONS

We now present simulation results for the performance of IARP and backpressure routing.

We consider a network as shown in Fig 4.1. The source node 8 desires to send data to destination node 20. Every node in the network is only interfered by its neighbors. We let node 2 and node 18 be busy every slot. This will increase the interference around them. The new packets arrive at the source every slot according to Poisson process with rate λ . In our simulation, each node can broadcast at most one packet per slot to its neighbors. Simulation was run for 10000 timeslots. Here we let $\text{Max}=5$, $\alpha = \beta = 0.8$. The results are given from Fig 4.2 to Fig 4.3, the unit of delay is slot.

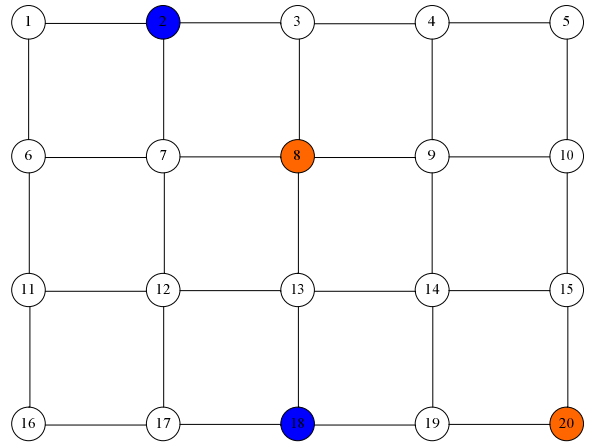


Fig 4.1 the network topology

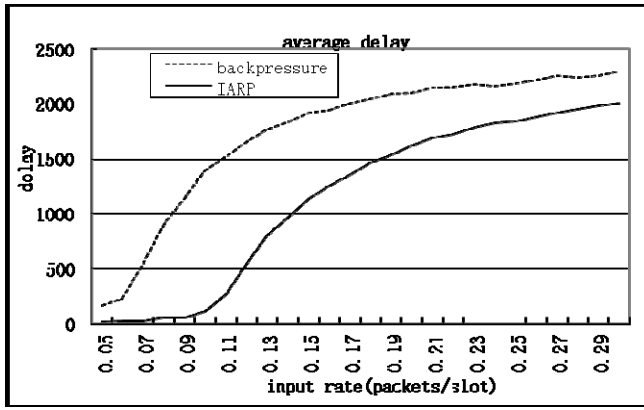


Fig 4.2 average delay

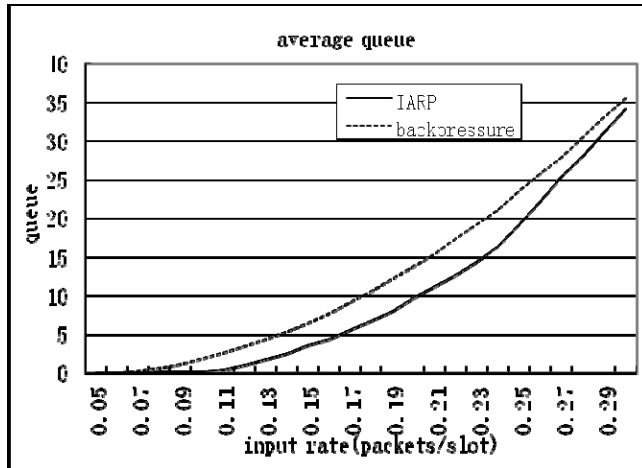


Fig 4.3 average queue

It can be seen that IARP has better performance in terms of network delay and average queue. We can observe that in Fig 4.2, when input rate is small, the difference from the two algorithms is significant. However, when input rate increases, the delay of the two algorithms become close to each other. This is because when network load is light, IARP can find a route with lower interference than that for backpressure routing. For example, route 8-9-10-15-20. When the whole network is busy, the parameters M_a, N_a, M_b, N_b in $W = U_a - U_b + M_a - M_b + N_a - N_b$ are all close to the upper bounds and W becomes close to $U_a - U_b$ in backpressure routing. This implies that the route 8-9-10-15-20 is also busy in this situation.

Finally, we have already theoretically shown that IARP is stable and we can also validate this in Fig 4.3. Note that when the input rate is large, the queue in the two algorithms are close, this means that the stability performance of the two algorithms are close.

V. CONCLUSION

In this paper, based on the busyness state of nodes, we propose a new routing protocol IARP that uses the only local information to find a route with less interference. The simulation results show that IARP can reduce the network delay and maintains shorter queues in nodes. In contrast to most related works, IARP only needs local information.

ACKNOWLEDGEMENT

This work was partially supported by National S&T Major Project(2011ZX03005-003-03, 2008ZX03005-001, 2010ZX03005-003), National Science Fund for Distinguished Young Scholars(60725105), National Basic Research Program of China (No.2009CB320404), the 111 Project (B08038) and State Key Laboratory Foundation (ISN090305, ISN1002005) National Natural Science Foundation of China (60972047, 60832001, 60725105, 61003300,60972048), Program for Changjiang Scholars and Innovative Research Team in University (IRT0852).

References

- [1] X. Zhang, Q. Liu, D. Shi, Y. Liu and X. Yu, "An Average Link Interference-aware Routing Protocol for Mobile Ad hoc Networks", Wireless and Mobile Communications, March 2007.
- [2] W.Lin, C. Zhang, M. Li and M. Wu; "An Interference Graph Based MAC Protocol for Ad Hoc Wireless Networks", Computer and Information Technology, Sept. 2006
- [3] W. Chen, K.B. Letaief and Z. Cao, "Cooperative Interference Cancellation in Multi-hop Wireless Networks: A Cross Layer Approach", GLOBECOM '06. IEEE, pp.1-5, Dec 2006
- [4] M. Liu, L. Li, Z. Xu and P. Chen, "IMBP: An Interference Control Topology Algorithm Based on Interference Measure in Ad Hoc Networks", Wireless Communications, Networking and Mobile Computing, pp.1-4, Sept. 2006.
- [5] Mohanoor, A.B, Radhakrishnan and S, Sarangan, V, "Interference Aware Multi-path Routing in Wireless Networks", Mobile Ad Hoc and Sensor Systems, pp.516-518, 2008.
- [6] K. Liu, S. Bohacek, J. Garcia-Frias, "Interference Mitigating in Wireless Networks Using Prior Knowledge", Information Sciences and Systems, pp.655-660, March 2007.
- [7] R. Huang, S. Kim, C. Zhang and Y. Fang, "Exploiting the Capacity of Multichannel Multiradio Wireless Mesh Networks", IEEE Transactions on Vehicular Technology, vol. 58, no. 9, pp.5037-5047, November 2009.
- [8] X. Yin, X. Zhou, R. Huang, Y. Fang, and S. Li, "A Fairness-Aware Congestion Control Scheme in Wireless Sensor Networks", IEEE Transactions on Vehicular Technology, vol. 58, no. 9, pp.5225-5234, November 2009.
- [9] 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, JULY 2009.
- [10] P. Gupta, P.R. Kumar, "The Capacity of Wireless Networks", IEEE Transactions on Information Theory, vol.46, March 2000.
- [11] L. Georgiadis, M.J. Neely, L. Tassiulas, "Resource Allocation and Cross-Layer Control in Wireless Networks", pp.1-144, 2006.
- [12] R.S. Sutton and A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998.