The Complexity of the Routing Problem in POR

Wei Hu, Jin Xie, and Zhenghao Zhang
Computer Science Department
Florida State University Tallahassee, FL 32306, USA

***Abstract*—We have designed and implemented a new protocol for wireless mesh networks called Practical Opportunistic Routing (POR). In this report, we document our finding about the complexity of the routing problem in POR. We prove that the routing problem is NP-hard by reducing a slightly modified version of the MAX-2-SAT problem to the optimal routing problem in POR.**

I. PRELIMINARIES

We have proposed a new packet forwarding and routing protocol for wireless mesh networks called Practical Opportunistic Routing (POR) [2]. As we have mainly focused on the practical design and implementation of POR in [2], in this report, we document our findings about the theoretical routing problem in POR. We begin with a brief review of the routing metric in POR; the details can be found in [2].

Suppose the path is $(v_1, v_2, ..., v_L)$. Any POR path must satisfy the *feedback constraint*; that is, any node on the path must be able to receive from its next hop node to ensure the correct reception of possible feedbacks from its downstream nodes. POR considers both the *forward cost* for sending data and *backward cost* for sending feedback. The path cost calculation is carried out iteratively, starting from the node closest to the destination. Therefore, when calculating the cost of the path, the forward and backward costs of path $(v_i, v_{i+1}, ..., v_L)$ are known for $2 \leq i \leq L$, denoted as C_{v_i} and B_{v_i}, respectively.

A. Path Cost in a Given State

Suppose the links involving v_1 are in a certain set of states denoted as τ.

1) Forward Cost: The *forward cost* of the path is denoted as $C^{\tau}_{v_1}$, which is defined as the consumed air time in data transmission in order to deliver a unit size block to the destination when the links are in state τ. We denote the BRR of link $v_1 \rightarrow v_i$ at rate ρ_j as $\mu_i^{\rho_j,\tau}$. We use $C^{\tau}_{v_1,\rho_j}$ to denote the air time consumed to deliver a unit size block to v_L following path $(v_1, v_2, ..., v_L)$, under the condition that v_1 transmits at rate ρ_j. We have

$$C^{\tau}_{v_1,\rho_j} = \frac{\frac{1}{\rho_j} + \sum_{i=2}^{L} \mu_i^{\rho_j} \prod_{t=i+1}^{L}(1-\mu_t^{\rho_j}) C_{v_i}}{1 - \prod_{i=2}^{L}(1-\mu_i^{\rho_j})}. \quad (1)$$

$C^{\tau}_{v_1}$ is $C^{\tau}_{v_1,\rho^*}$ if ρ^* has the minimum cost among all rates.

2) Backward Cost: The *backward cost at* v_1 is denoted as $B^{\tau}_{v_1}$ and defined as the consumed air time in feedback transmission in order to deliver a packet to the destination. We let

$$\gamma_i = \frac{\mu_i^{\rho^*,\tau} \prod_{t=i+1}^{L}(1-\mu_t^{\rho^*,\tau})}{1 - \prod_{i=2}^{L}(1-\mu_i^{\rho^*,\tau})}.$$

The backward cost is calculated according to

$$B^{\tau}_{v_1} = \gamma_2 B_{v_2} + \sum_{i=3}^{L} \gamma_i [B_{v_i} + (1-\mu_2^{\rho^*,\tau})\eta_2 + \sum_{t=3}^{i} \eta_t] \quad (2)$$

where η_t denotes air time v_t uses to send one feedback, which can be calculated according to the rate v_t uses to send feedback and the simplifying assumption that each feedback frame contains exactly 8 feedbacks.

B. Path Cost in Multiple States

We find in our experiments that the links can be in multiple states. As a response, the rate selection algorithm can converge to different data rates. The cost of a path is simply the weighted average of the path costs in each individual set of states, where the weight of a set is the probability that the links are in this particular set of states.

II. NP-HARDNESS PROOF

Our proof is based on a modified version of the MAX-2-SAT problem [1]. In the original version of MAX-2-SAT problem, N binary variables and S clauses are given where each clause contains exactly two literals in the *or* operation, where a literal is either a binary variable or the negation of the variable. The goal is to find an assignment of the binary values of the variables such that maximum number of clauses are satisfied. We define a similar MAX-2-SAT problem in which the operation inside the clause is *and* as the AMAX-2-SAT problem.

Theorem 1: The AMAX-2-SAT problem is NP-hard.

Proof: Given any instance of the original MAX-2-SAT problem, we create an instance of the AMAX-2-SAT problem as follows. For any variable x_i, create two types of shadow variables denoted as x_i^1 to x_i^3 and y_i^1 to y_i^G, where $G > 100S$. For notational simplicity, we also use x_i^0 to denote x_i. We refer to x_i^0 to x_i^3 the *set-1 variables* and y_i^1 to y_i^G the *set-2 variables* for x_i. For each pair of set-2 variables y_i^u and y_i^v, create two clauses: $(\neg y_i^u \wedge \neg y_i^v)$ and $(y_i^u \wedge y_i^v)$. For each x_i^t and y_i^u where $0 \leq t \leq 3$ and $1 \leq u \leq g$, create two clauses $(\neg x_i^t \wedge \neg y_i^u)$ and $(x_i^t \wedge y_i^u)$, where $10S < g < \frac{G}{10}$. Group all clauses in the MAX-2-SAT instance according to the variables in the clauses; clauses involving the same two variables belong to the same group. Clearly, each group contains at most 4 clauses. Give indices to the clauses in the same group; for example, for the clauses involving x_i and x_j, $(\neg x_i \vee \neg x_j)$, $(\neg x_i \vee x_j)$, $(x_i \vee \neg x_j)$, $(x_i \vee x_j)$ are indexed as 0, 1, 2, 3 and denoted as $C_0^{i,j}$ to $C_3^{i,j}$, respectively. Convert each *or* clause in the MAX-2-SAT instance to 3 clauses in the *and* form such that one of the *and* clauses will be true if the original clause is true. Then, replace the variables in the converted *and* clauses with the shadow variables, e.g., x_i in $C_t^{i,j}$ is replaced with x_i^t.

We first claim than set-2 variables for x_i will take the same values. This is because if there are certain variables taking different values from the majority, by converting them to take the same value as the majority, there will be an increase of at least $G-1$ satisfied clauses among the clauses involving only the set-2 variables. By so doing, there will be a decrease of at most $4g$ satisfied clauses among the clauses involving one of the set-2 variables and one of the set-1 variables. Therefore, the set-2 variables must take the same value because $G > 10g$ in our construction. We next claim that the set-1 variables also take the same value as the set-2 variables. This is because if one set-1 variable takes a different value, by changing its value, there will be an increase of g satisfied clauses among the clauses involving one of the set-2 variables and one of the set-1 variables. By so doing, there will be a decrease of at most $3S$ satisfied clauses among the clauses involving only the set-1 variables. Therefore, the set-1 variables must take the same value because $g > 10S$ in our construction.

Given any optimal solution in the constructed AMAX-2-SAT instance, if there are N satisfied clauses among the clauses involving only the set-1 variables, we can use the same assignment and obtain N satisfied clauses in the MAX-2-SAT instance, and vice versa. ■

We next introduce a new problem called the *Distance-Weighted MAX-2-SAT* (DWM-2-SAT) based on the AMAX-2-SAT problem. We define the *distance weight* of a clause involving x_i and x_j as $|i-j|$. The DWM-2-SAT problem is defined as: Given N binary variables indexed from 1 to N and S *and* clauses with two literals, find an assignment such that the sum of the weights of the satisfied clauses is maximized.

Theorem 2: The DWM-2-SAT problem is NP-hard.

Proof: Given any instance of the AMAX-2-SAT problem, we construct an instance of the DWM-2-SAT problem as follows. First, for any variable x_i, create two shadow variables denoted as x_{i+D1} and x_{i+D2}, where $D1$ and $D2$ are large constants, i.e., $D1 > SN$ and $D2 > S(D1+N)+D1$. Then, for any clauses in the AMAX-2-SAT instance involving x_i and x_j, if $i < j$, replace x_j with x_{j+D1}. Also, introduce 4 shadow clauses for each variable: $(\neg x_i \wedge \neg x_{i+D2})$, $(x_i \wedge x_{i+D2})$, $(\neg x_{i+D1} \wedge \neg x_{i+D2})$, and $(x_{i+D1} \wedge x_{i+D2})$. Then, create dummy variables to fill in the gaps of the indices; the dummy variables are not in any clauses.

We first claim that x_i and x_{i+D1} must take the same value in the optimal solution for the constructed DWM-2-SAT instance. Suppose this claim is not true and there is an optimal solution in which x_i and x_{i+D1} take different values. As a result, exactly one of the shadow clauses for x_i is satisfied. On the other hand, if we change x_i and x_{i+D1} to be the same as x_{i+D2}, between the shadow clauses, the weighted sum is increased by at least $D2-D1$. Among the original clauses, due to the change of x_i or x_{i+D1}, the weighted sum is decreased by at most $S(D1+N)$, which is still less than $D2-D1$. Therefore, this contradicts the fact that the solution is optimal.

We next claim that the optimal solution for the DWM-2-SAT instance actually maximizes the number of satisfied original clauses. To see this, consider two assignments denoted as A1 and A2 which satisfy $N1$ and $N2$ original clauses, respectively, where $N1 > N2$. The weighted sum of A1 among the original clauses is greater than $N1D1$, while the weighted sum of A2 among the original clauses is less than $N2(D1+N)$, which is less than $N1D1$ because $D1 > SN$ in our construction. ■

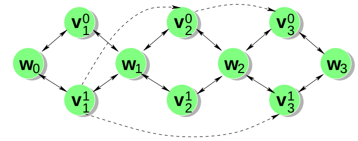

Fig. 1. Constructed OPOR instance for a DWM-2-SAT instance with three variables and three clauses: $(x_1 \wedge \neg x_2)$, $(x_1 \wedge x_3)$, and $(\neg x_2 \wedge \neg x_3)$.

Theorem 3: The Optimal Practical Opportunistic Routing (OPOR) problem is NP-hard.

Proof: Given any instance of the DWM-2-SAT problem, we construct an OPOR instance as follows. Assume there is no partial packet. First, introduce a source node denoted as w_0. For any variable x_i, create three nodes denoted as v_i^0, v_i^1, and w_i. Create links from v_i^0 to w_i and from v_i^1 to w_i. Also create links from w_{i-1} to v_i^0 and from w_{i-1} to v_i^1. Such links are referred to as *main links* and can only operate at a specific rate with PRR of 1. The reverse direction of the main links operate at a much higher data rate with PRR of 1. There are *overhearing links* from a v-node to another v-node depending on the DWM-2-SAT instance. Suppose $i < j$, there is an overhearing link from v_i^0 to v_j^0, from v_i^0 to v_j^1, from v_i^1 to v_j^0, and from v_i^1 to v_j^1 if clauses $(\neg x_i \wedge \neg x_j)$, $(\neg x_i \wedge x_j)$, $(x_i \wedge \neg x_j)$, and $(x_i \wedge x_j)$ are in the DWM-2-SAT instance, respectively. The PRR of the overhearing link is denoted as β which is a positive but very small number, e.g., $\beta < N^{-100}$. The reverse direction of an overhearing link has PRR of 0 over all rates. The goal is to find an optimal path from w_0 to w_N. Fig. 1 shows an example of the construnction.

Given this construction, we make the following claims:

1) If w_i is on the path, at least one between v_i^0 and v_i^1 must appear immediately before w_i, due to the feedback constraint.
2) If v_i^0 (or v_i^1) is on the path, w_{i-1} must also be on the path and appear immediately before it due to the feedback constraint.
3) Every w-node is on the path. Clearly, w_N must be on the path. Suppose w_i is not on the path and is the one with the largest index. Therefore, neither v_{i+1}^0 nor v_{i+1}^1 is on the path. However, this contradicts that w_{i+1} is on the path.
4) w-nodes appear on the path in order. If not, there must exist a w_i and w_{i+1} where w_{i+1} appears earlier than w_i on the path. Suppose v_{i+1}^0 appears before w_{i+1}.

However, this contradicts that w_i must appear before v^0_{i+1}.

5) In any path, between w_{i-1} and w_i, there is exactly one node, either v^0_i or v^1_i. Suppose there are more than one node. Without loss of generality, suppose v^0_i is between w_{i-1} and w_i. If v^0_i appears immediately before w_i, it cannot send feedback to the node between itself and w_{i-1} because it can only send feedback to w_{i-1}. If v^0_i appears immediately after w_{i-1}, w_i can send feedback to its previous hop only when its previous hop is v^1_i; however, v^1_i cannot send feedback to its previous hop because it can only send to w_{i-1} while v^0_i is between them.
6) A path will always be in the form of $w_0, v^{x_1}_1, w_1, v^{x_2}_2, w_2, ...w_N$ where x_i is a binary number.

We note that the optimal path need only optimize the forward cost because the feedback cost is small. We claim that the optimal path is the one that maximizes the total number of bypassed main links by the overhearing edges. We denote the number of overhearing links originated at v^x_i as R_i. Among the links, suppose the b_{th} shortest overhearing link bypasses s^i_b w-nodes. Let $S_i = \sum_{b=1}^{R_i} s^i_b$. We define $d_i = C_{v^x_i} - [2(N-i)+1]$. It is clear that $d_i \leq 0$ and $C_{w_i} = C_{v^x_{i+1}} + 1 = 2(N-i) + d_{i+1}$. Normalizing the data rate to 1, we have

$$\begin{aligned} C_{v^x_i} &= 1 + (1-\beta)^{R_i} C_{w_i} \\ &\quad + \sum_{b=1}^{R_i} \beta(1-\beta)^{R_i - b}[2(N-i) + 1 - 2s^i_b + d_{i+s^i_b}] \end{aligned}$$

Therefore,

$$\begin{aligned} C_{v^x_i} &< 1 + (1-\beta)^{R_i} 2(N-i) + (1-\beta)^N d_{i+1} \\ &\quad + \sum_{b=1}^{R_i} \beta(1-\beta)^{R_i - b}[2(N-i) + 1 - 2s^i_b]. \end{aligned}$$

As a result

$$d_i < (1-\beta)^N d_{i+1} + [1 - (1-\beta)^N] - \beta(1-\beta)^N 2S_i$$

As $d_N = 0$, for $i < N$, based on simple induction, we have

$$d_i < (N-i)[1-(1-\beta)^N] - \beta \sum_{j=i}^{N} (1-\beta)^{(j-i+1)N} 2S_j$$

and therefore

$$d_1 < N[1-(1-\beta)^N] - \beta(1-\beta)^{N^2} \sum_{j=1}^{N} 2S_j \quad (3)$$

On the other hand,

$$\begin{aligned} C_{v^x_i} &> (1-\beta)^{R_i}[2(N-i)+1] + d_{i+1} \\ &\quad + \sum_{b=1}^{R_i} \beta(1-\beta)^{R_i - b}[2(N-i) + 1 - 2s^i_b + d_{i+s^i_b}], \end{aligned}$$

therefore,

$$d_i > d_{i+1} - \beta 2S_i + \beta \sum_{b=1}^{R_i} d_{i+s^i_b}$$

We prove that

$$d_i > -\beta \sum_{j=i}^{N} 2S_j - \beta^2 (N-i) N^3,$$

which is clearly true when $i = N$. Suppose it is true till d_{i+1}. For d_i, we note that

$$\begin{aligned} d_i &> -\beta \sum_{j=i+1}^{N} 2S_j - \beta^2 (N-i-1) N^3 - \beta 2S_i \\ &\quad -\beta \sum_{j=i+1}^{N} [\beta \sum_{t=j}^{N} 2S_t] - \beta \sum_{j=i+1}^{N} \beta^2 (N-j-1) N^3 \\ &> -\beta \sum_{j=i}^{N} 2S_j - \beta^2 (N-i) N^3 \end{aligned}$$

Therefore,

$$d_1 > -\beta \sum_{j=1}^{N} 2S_j - \beta^2 N^4 \quad (4)$$

Considering Eq. 3 and Eq. 4, given any two paths, the one with a larger $\sum_{j=1}^{N} 2S_j$ will have a lower cost.

It is then clear that an optimal solution for the constructed OPOR instance leads to an optimal solution for the original DWM-2-SAT instance because each overhearing link exploited in the optimal path is a satisfied clause. ■

REFERENCES

[1] M. R. Garey and D. S. Johnson, "Computers and Intractability, a Guide to the Theory of NP-Completeness," *W. H. Freeman and Company*, 1979.

[2] W. Hu, J., Xie, and Z. zhang, Practical opportunistic routing in high-speed multi-rate wireless mesh networks, in *Proc. of ACM Mobihoc*, Bangalore, India, August 2013.