Reliable Routing in Mobile Ad-Hoc Networks*

Spyros Tragoudas Khadija Stewart
Department of Computer and Electrical Engineering
Southern Illinois University, Carbondale IL 62901, USA

Abstract: - We propose that a new parameter, the set of link stability functions, should be taken into consideration when selecting a route for reliable data transmission in an Ad Hoc Network. Due to this parameter, any proposed routing algorithm must explicitly take into consideration the data transmission method. Polynomial time routing algorithms for both the store-forward and cut-through methods are presented, and their comparison in terms of transmission time and link stability is given on benchmarks.


1 Introduction

In routing, the goal is to select the shortest route between a source s and a target t in the network. This formulation implies that the data are transmitted using the store and forward (SF) method for all the nodes (hops) along the selected route. That way, there is no need to explicitly consider either the amount of data that needs to be transmitted or the link capacities on the route. Faster data transmission can be obtained using the cut-through (CT) method but in that case the amount of data σ and the link capacities c(u,v) must be considered [1-5].

Such methods may not be very reliable for Mobile Ad-Hoc Networks (MANETs) where each link may be disconnected for some time intervals. We assume that at time t=0, the time of route selection, each link (u,v) is associated with a set of time intervals LS(u,v) during which the link has high probability of being functional. During all other intervals we assume that the link will be most likely nonfunctional. This set LS(u,v) is called the link stability function of the link (u,v). The LS(u,v) set can be obtained with techniques as in [6], among others. We also assume that the data transmission will start at time t=0.

A reliable route must avoid routing data through a link (u,v) on time intervals not in LS(u,v). We should also attempt to route through intervals in any LS(u,v) that are as close to t=0 as possible; (u,v) may also turn out to be nonfunctional during intervals in LS(u,v) that are far from t=0. In addition, it is known that the route’s reliability decreases exponentially in MANETs as the data transmission time (TT) along the route increases.

From the above, we conclude that a highly reliable route should minimize TT while satisfying all LS(u,v) as much as possible. New route selection algorithms for the SF and CT transmission methods are required and are outlined in Section 2 of this paper. An experimental comparison between the two algorithms is given in Section 3. We consider reactive routing [7,8]. The algorithms can easily be integrated to scalable architecture as in [9-12]. They resemble or built upon

* Research supported in part by NSF grant 0096199
Dijkstra’s shortest path algorithm and therefore can be easily modified to apply to a distributed environment [13].

2 Routing algorithms

When using the SF transmission method, the TT along a selected route can be approximated using Eq. (1) below. Quantities \( l(u,v) \) and \( c(u,v) \) denote the lead delay and capacity functions of link \((u,v)\) on the route. \( \varepsilon(u,v) \) is the minimum required delay in transmission at node \( u \) so that link \((u,v)\) becomes functional.

\[
\sum_{(u,v)} (\sigma/c(u,v)(1 + l(u,v)) + \varepsilon(u,v)).
\]

The routing algorithm for the SF transmission method is a modification of Dijkstra’s shortest path algorithm. At Step (2), the algorithm selects a node \( u \) for which the quantity \( \sigma/c(s,u)(1 + l(s,u)) + \varepsilon(s,u) \) is minimized. This selection guarantees that the TT is minimized when using Eq. (1). New nodes are selected as in Step (3). Such node selections resemble Dijkstra’s method, and we have shown that the TT is minimized according to Eq. (1). Let \( V \) be the set of all nodes.

Algorithm SF :

1. \( TT[s]=0; \ S=\{s\}. \ TT[u] = \infty, u \leftrightarrow s. \ //S \ is \ the \ permanent \ set./\
2. for \( i := 2 \) to \( n \) do
   \( TT[u] = +\sigma/c(s,u)(1 + l(s,u)) + \varepsilon(s,u) \)
3. for \( i := 1 \) to \( n-1 \) do {
   a. Select \( u \) in \( V-S \) such that \( u \) with minimum \( TT[u] \); \( S = S \cup \{u\} \).
   b. for each \( v \) in \( V-S \) and \( u \) as (a) do
      \( TT[v]=\min(TT[v], TT[u] + \sigma/c(u,v)(1 + l(u,v)) + \varepsilon(u,v)) \)

We have shown that the time complexity of algorithm SF is linear to the number of links, up to poly-logarithmic factors. However, for the CT transmission method we have shown that no polynomial time routing algorithm that guaranteed to find a route that does not violate the \( LS(u,v) \) functions on the links can be designed. The problem is NP-hard, and we rely on heuristics that try to satisfy the \( LS(u,v) \) functions as much as possible. The TT is calculated by considering a linear number of sub networks \( N_c \). Each \( N_c \) consists of all the links with capacity higher than \( c \). (For calculation purposes we can then assume that all links have capacity \( c \).) For each sub network, the TT is given by Eq. (2) below, and the goal is to select the route \( P \) that results into minimum TT.

\[
\sigma/c+\Sigma_{(u,v)}on\ P[l(u,v)].
\]

At the end, the best TT route among all sub networks \( N_c \) is considered. If we were able to find the route that minimizes Eq. (2) in some sub network \( N_c \) we would have had an optimal polynomial time algorithm, but it is NP-hard to calculate this quantity even on one \( N_c \). We thus present a linear time complexity heuristic (up to poly-logarithmic factors which is based on a modification of Dijkstra’s algorithm and is described below. We call it Algorithm CT.

The idea is that before we put any link \((u,v)\) in the permanent set \( S^1 \), we check to see if the link is predicted to be functional at the time that the transmission is predicted to take place according to Eq. (2). If not, we ignore node \( v \); if yes, we put \((u,v)\) in the permanent set. If that way if we cannot reach destination \( t \), we go to our temporary list and we choose the link \((u,v)\) that has the smallest positive \( \varepsilon(u,v) \) value such that \( u \) is in the permanent set \( S \) and \( v \) is not in \( S \). We then try to find a path \( P \) from the source \( s \) to node \( u \) whose links are all functional according to the provided \( LS(u,v) \) functions.

The above process is repeated recursively until we eventually reach the destination node \( t \). That way, the route uses functional edges. However, a route may not be found using this approach, in which case a fast transmission route that may require some time intervals which are not in the \( LS(u,v) \) functions is selected. For the experimental results listed in the next section, routes that require time intervals which are not in functions \( LS(u,v) \) are not taken into consideration. In that case we report that no route exists.

---

1 The permanent set is the set of links to which the algorithm has found the shortest route.
2 The temporary list contains the links that haven’t been put in the permanent set.
3 Experimental results

We experimented on a Sun Blade 1000 workstation and on the ISCAS 85 benchmarks for VLSI CAD due to the lack of standardized benchmarks for MANETs. The circuits were transformed to bi-directional networks. The CPU performance was always in the order of seconds for both algorithms. The number of links is approximately double of the number in each benchmark’s description. The $c(u,v)$, $l(u,v)$ and $LS(u,v)$ we assigned randomly and the derived benchmarks are given in www.engr.siu.edu/grad1/jirari.

We experimented by considering 1 or 2 randomly generated intervals where a link is functional in each $LS(u,v)$. The number of $s,t$ pairs is also randomly chosen and is uniformly distributed from 10 to 40. The $%LS$ column for algorithm CT indicates the average percentage of routes that used only functional links. This quantity is often below 70%. Since the $LS$ functions imply that the transmission will be incomplete unless we consider the functional intervals [1], we conclude that the SF routing algorithm results in more reliable routing despite the comparatively high transmission time $TT$ along the selected routes. Future work should focus on improving the quality of the CT heuristic or on relaxing the constraint of minimizing the $TT$ in the route selected by the CT heuristic so that all used intervals are in the $LS(u,v)$ functions.

<table>
<thead>
<tr>
<th>Bench</th>
<th>1 interval</th>
<th>2 intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alg SF</td>
<td>Alg CT</td>
</tr>
<tr>
<td></td>
<td>$TT$</td>
<td>$TT$</td>
</tr>
<tr>
<td>c17</td>
<td>536</td>
<td>10</td>
</tr>
<tr>
<td>c432</td>
<td>916</td>
<td>17</td>
</tr>
<tr>
<td>c499</td>
<td>830</td>
<td>194</td>
</tr>
<tr>
<td>c880</td>
<td>1382</td>
<td>59</td>
</tr>
<tr>
<td>c1908</td>
<td>1664</td>
<td>414</td>
</tr>
<tr>
<td>c2760</td>
<td>2252</td>
<td>465</td>
</tr>
<tr>
<td>c5315</td>
<td>9470</td>
<td>351</td>
</tr>
</tbody>
</table>

Table 1. Average performance of algorithms SF and CT.

References