SKVR: Scalable Knowledge-based Routing Architecture for Public Transport Networks

Shabbir Ahmed, Salil S. Kanere
School of Computer Science and Engineering
The University of New South Wales
Sydney, Australia
shabbira@cse.unsw.edu.au, salilk@cse.unsw.edu.au

ABSTRACT

Vehicular AdHoc Networks (VANET) can be treated as special kinds of Delay-tolerant Networks (DTN) where end-to-end path might never be possible. As a result, mobile ad-hoc (MANET) routing protocols perform poorly on DTNs. Moreover, traditional routing architecture is not scalable for public transport networks with large numbers of nodes (public transports). In this paper, we introduce a hierarchical knowledge-based DTN routing scheme for public transport networks that is not only scalable but also communication efficient. We evaluate our design using simulation with real bus traces.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: [Wireless communication]; C.2.2 [**Network Protocol**]: Routing protocols

General Terms

Performance, Reliability

Keywords

DTN, Delay Tolerant Networks, VANET, Vehicular AdHoc Networks, Routing

1. INTRODUCTION

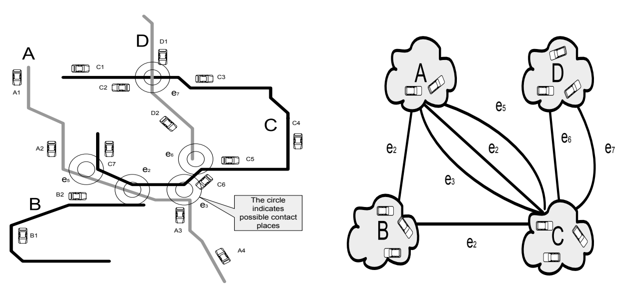
An analysis of trace files of actual vehicle movements used in [1] reveals that public transport networks are highly disconnected that share the properties of DTN. In public transport networks, the node mobility pattern is quite predictable. For example, bus time-tables give us an idea when a bus will arrive or depart from a bus stoppage, though the schedule is not precise. A little amount of additional knowledge can greatly aid routing protocols [5, 3, 2]. Instead of considering the entire knowledge about mobility as dynamic, we partition the knowledge in two categories: a) Static knowledge: that does not change often over time. For example, bus routes are fixed and usually do not change often. b) Dynamic knowledge: that change quite often. As an example, we can consider bus time tables. Though the time table is fixed, it is not guaranteed that a bus will be able to follow that time table exactly.

VANET'06, September 29, 2006, Los Angeles, California, USA.
ACM 1-59593-540-1/06/0009.

This classification simplifies knowledge based routing. In our proposed routing architecture, we use these two classes of knowledge at different hierarchical level. The hierarchy consists of inter-domain (among bus-routes) routing scheme at the top level and intra-domain (inside a bus route) routing scheme at the bottom level. The top level hierarchy use static knowledge and low level use both static and dynamic knowledge to make routing decision. The routing problem among the nodes of the top level hierarchy turns into merely a problem of finding shortest path between two nodes.

2. SKVR ARCHITECTURE

Fig. 1a shows an example of hierarchical structure of public transport networks. Each vehicle has a node id and a bus-route id. The circles indicate possible locations where contacts (neighboring vehicles) from other bus routes can be found. However, finding contacts at these locations are time dependent and probabilistic.



(a) (b)

Figure 1: Example hierarchical structure of public transport networks

Fig. 1b shows an equivalent graph of the top level hierarchy of fig. 1a. The nodes represent bus-routes and links between them are time varying and probabilistic. At the top level hierarchy of our proposed routing architecture, the bus-routes can be considered as nodes of a graph $G(V, E)$. The graph represents connection instance [4] of bus-routes where each vertex v_i in V denotes a bus-route and each edge e_{ij} in E represents a path from v_i to v_j (probably at road crossings or at areas of near proximity between two bus-routes) (Fig. 1). At the bottom level of the hierarchy, the routing mechanism follows the DTN routing paradigm *store, carry and forward* instead of the traditional *store and forward* method.

2.1 Intra-domain Routing

It is intuitive that depending on the destinations position in the bus route, the packet should be forwarded to another vehicle moving in either forward or reverse direction. If the destination vehicle is in front (may be several hops away) of the forwarding node, then the packet should be forwarded to the vehicle in front that is moving towards the destination and vice versa. A simple way to determine destination vehicles position is as follows: a vehicle keeps track of previous contacts (of same bus-route) it encounters during its journey from one end of the route to another end (and resets this list after reaching any end of the route). Each vehicle also announces its heading direction (a binary flag) in beacons. Whenever, it has a packet to be forwarded, it consults its previous contacts list and if the destination is in the previous contacts list then the destination must be located in reverse direction of the forwarding vehicle. The forwarding node then marks the direction flag of the packet.

A forwarding node should choose the oldest node in the contact list heading in opposite direction but the oldest contact in the list should not be too old that it goes out of communication range within short period. This situation can be avoided by checking the timestamp of the oldest node with current timestamp against a threshold. In other words, the contact should be chosen for which the following inequality holds:

$$T_c - max(t_c - t_o) > t_{th} \quad (1)$$

where: t_c = current timestamp, t_o = first appearance time of the oldest contact, t_{th} = a threshold value safe enough to transfer the packet, T_c = contact duration (depends on the velocity of the two nodes). For example, if the communication range is 250m then $t_{th} = \alpha \times 2 \times 250/(v_1 + v_2)\,[where: 0 < \alpha < 1]$.

2.2 Inter-domain Routing

The basic idea for inter-domain routing is to forward the message to a contact that either belongs to the destination bus-route or comes across destination bus-route in some time later. If we can push the message to any vehicle of destination route, it is guaranteed to be delivered because delivery is ensured once the message has been forwarded to a node which belongs to the same bus-route as the destination node (using Intra-domain rules). At this stage, the forwarding node can also purge the message from its buffer to prevent unnecessary circulation of the message.

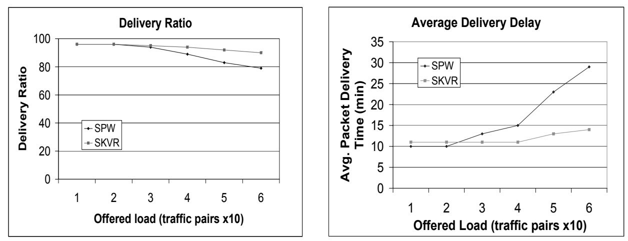
If the contact list does not contain any node which belongs to the destination bus-route, a straight forward approach is to send the copy of the message to m distinct contacts from a total of n contacts ($n > m$). However, a more efficient forwarding rule can be constructed from the global knowledge. Contact list can be ranked by using the knowledge of destination bus-route and then copying the message to the top ranking 2 or 3 contacts increase the probability of delivery. The stochastic knowledge about the quality of the links can be used to make path selection when there are multiple paths between two nodes.

3. PERFORMANCE EVALUATION

We use ns-2 to simulate our proposed routing model and to compare it with *Spray and Wait*[6]. Simulation parameters are given in table 1. Fig. 2 compares both protocol's packet delivery ratio and delay. The offered load is varied at the x-axis. Due to space constraints, results from other experiments are omitted. It can be concluded that unnecessary traffic generation, collision and buffer utilization can negatively impact the performance of a routing protocol. So even though the *Spray and Wait* works quite well at low traffic, its performance degrades seriously at high load. On the other hand our protocol seems to scale well with increasing load.

Table 1: Simulation Parameters

Parameter	Value
Simulation Area	30000m×65000m
Simulation Time	1 hour
Beacon Interval	5 sec
No. of vehicles	178
No. distinct bus routes	84
No. of packet senders	60
Communication Range	250m
MAC protocol	802.11
Antenna Type	Omni-directional
Radio Propagation Model	Two-Ray Ground



(a) (b)

Figure 2: Comparison of delivery ratio (a) and packet delay (b)

4. CONCLUSIONS

In this paper, we have presented a scalable and disruption tolerant routing architecture for public transport networks that utilizes both static and dynamic knowledge about network and node mobility. We empirically evaluate our proposed method using simulation with real bus traces with a large number of buses and compare it with other routing protocols. The preliminary simulation results show the efficacy of our proposed method.

5. REFERENCES

[1] J. G. J. et al. Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture. In *Proceedings of 5th IEEE Workshop on Mobile Communications.*

[2] D. G. Jason LeBrun, Chen-Nee Chuah. Knowledge-based opportunistic forwarding in vechicular wireless ad hoc networks. In *Proceedings of IEEE VTC, Vol. 4*, pages 2289–2293, Spring 2005.

[3] E. P. C. Jones and P. A. S. Ward. Routing strategies for delay-tolerant networks. In *Proceedings of ACM SIGCOMM*, 2004.

[4] Q. Z. Kun Tan and W. Zhu. Shortest path routing in partially connected ad hoc networks. In *IEEE Global Telecommunications Conference GLOBECOM, Vol. 2*, pages 1038–1042, April 2003.

[5] R. P. Sushant Jain, Kevin Fall. Routing in delay tolerant network. In *Proceedings of ACM SIGCOMM*, 2004.

[6] K. P. Thrasyvoulos Spyropoulos and C. S. Raghavendra. An efficient routing scheme for intermittently connected mobile networks. In *Proceedings of ACM SIGCOMM*, 2005.