Location Independent Compact Routing for Wireless Networks

Robert Gilbert, Kerby Johnson, Shaomei Wu, Ben Y. Zhao, Haitao Zheng
Department of Computer Science
University of California, Santa Barbara, CA 93106-5510
{rgilbert, kerby, shaomei, ravenben, htzheng}@cs.ucsb.edu

ABSTRACT

While reactive routing protocols such as AODV operate efficiently for small ad hoc wireless networks, their $O(N)$ per-flow control overhead limits deployment on larger-scale networks. Deployment of compact routing protocols such as geographic routing have met with challenges. In this paper, we present Table Attenuation Routing Protocol (TARP), a protocol that combines compact per-node routing state with scalability to large networks. Preliminary evaluation shows that TARP performs similar to AODV in smaller networks and better in larger networks.

Categories and Subject Descriptors

C.2.2 [**Computer-communication Networks**]: Network Protocols

Keywords

Wireless routing, Bloom filters

1. INTRODUCTION

Existing work on wireless routing protocols have focused on two primary approaches. Reactive routing protocols such as AODV [15] and DSR [8] are de facto standards, but require global broadcasts in route management, and maintains per-flow state. Route broadcasts grow exponentially with network size, increasing overhead and congestion on large networks. Geographic forwarding protocols such as GPSR [9] are proposed a "state-less" alternative, where nodes maintain no per-flow state, and only a constant ($O(1)$) amount of routing state. While lowering overhead, these protocols are limited by a dependence on geographic location information (*e.g.* GPS devices). In addition, recent work has shown that GPSR operates poorly in real world conditions, while other variants incur higher control overhead or per-node state [10, 11].

To achieve low per-node state and scalability to large networks in a simple protocol, we propose *Table Attenuation Routing Protocol* (TARP), a scalable and compact wireless routing protocol. TARP is a proactive routing protocol, where nodes proactively maintain connectivity and routing information with immediate neighbors. By managing routing state as "soft state," TARP is simple and avoids explicit failure recovery mechanisms. Unlike previous proactive protocols [7, 14], TARP uses lossy compression via Bloom Filters [1] to compress network-wide routing state to a fixed number of routing entries.

This short paper introduces the TARP protocol, describes a mechanism for extending TARP to large networks using self-organizing beacons, and presents Qualnet simulation results that evaluate TARP against AODV and Optimized Link State Routing (OLSR) [7].

MobiShare'06, September 25, 2006, Los Angeles, California, USA.

2. RELATED WORK

Studies of wireless routing protocols [2,5] have found proactive algorithms to incur high control message overhead, resulting in network congestion and lower packet delivery rates. Geographic routing protocols such as GPSR [9] were designed as a light-weight alternative to reactive protocols that required $O(1)$ per-node routing state independent of network flows. Followup work led to the development of more complex and stateful protocols such as CLDP [10] and GDSTR [11].

Several existing protocols share similarities with TARP. Randomly placed beacons are also used in Beacon vector routing [6], but nodes near the beacon are likely to pay a higher cost to forward traffic. The new VRR protocol [3] performs scalable wireless routing by adopting some ideas from the structured overlay community, essentially maintaining proactive routes to a node-specific set of "landmarks." One property that distinguishes TARP is that, in the absence of failure, it provides the shortest hop path between any two endpoints. Finally, TARP's backtracking to beacons is similar to the decentralized object location and routing (DOLR) [4] interface implemented by structured overlays like Tapestry [18].

3. TABLE ATTENUATION ROUTING

TARP is a proactive routing protocol that utilizes Bloom filters to compress routing state into compact form. The compressed routing state results in both smaller per-node state as well as lower routing maintenance overhead.

3.1 Background: Bloom Filters

A Bloom filter [1] is a compact data structure for fast membership decisions that queries whether a particular ob-

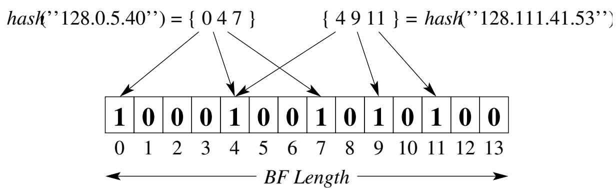


Figure 1: The computation of a Bloom Filter for a set containing two IP addresses.

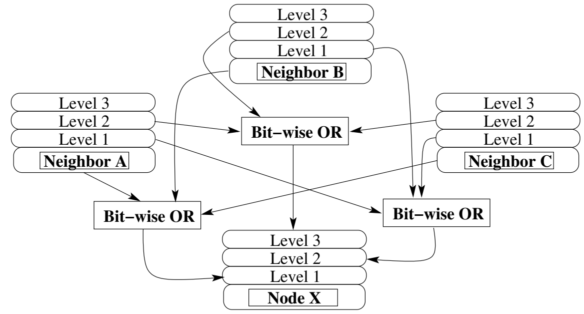


Figure 2: Building up a LayerTable from those of its neighbors.

ject is a member of a known group. Membership information is encoded as a sequence of bits. Specifically, a Bloom filter is a bit vector of size m used to represent a set $S = \{s_1, s_2, ..., s_n\}$ of n objects. The filter is populated by applying k independent hash functions, $h_1, h_2, ..., h_k$, to each of the elements in S, resulting in k bit positions in the bit vector. Each $s \in S$ generates a bit vector with k bits turned on, one for each bit position returned from $h_i(s)$ for $1 \leq i \leq k$. The bit-wise union (`OR`) of all bit vectors gives us the Bloom filter for object set S. An example of a Bloom filter encoding is shown in Figure 1.

Item x is an element of S only if all bit positions returned from $h_i(x)$ are set in the Bloom filter for set S. If not, then x is not a member. If all bit positions are set, $x \in S$ is true with some probability. False positives are possible, depending on the size of set S and length of the filter. For a given set of size n, we can reduces the false positive rate by tuning parameters m and k [12, 16].

3.2 Building and Using Routing Tables

A TARP node stores its reachability information to all nodes as a *stack* of Bloom filters, where the filter at the i^{th} *layer* is a summary of signatures of all nodes reachable in $i-1$ hops. The filter at layer 1 is a filter with only the local node's hashed positions turned on. By searching for node N's signature through layers of filters, one can determine the length of the shortest path (in hops) to that N. This *LayerTable* is similar to attenuated Bloom filters previous proposed for object location [16].

Building a node's LayerTable is a recursive process. Each node periodically broadcasts its LayerTable to its immediate neighbors. N determines its route to all destinations by choosing the shortest path (in hops) to each node from its neighbors and incrementing the hop count by one. A TARP node's LayerTable at layer i is computed by performing Bit-wise `OR` across the $(i-1)^{th}$ layers of all of its neighbors. This process is illustrated in Figure 2.

A node also uses its neighbors' LayerTables to determine which neighbor is the next hop for its messages. While the LayerTable tells it a destination is h hops away, its neighbors' LayerTables at layer $h-1$ tell it which neighbor is the next hop. Hence each node maintains a set of its neighbors' LayerTables, which we call a *NeighborTable*.

In summary, a node maintains two sets of routing state, a local LayerTable that encodes distance to every destination, and a NeighborTable that lists similar information for its immediate neighbors. To route to destination D, a node first searches its LayerTable (in increasing layer order) to determine distance to D in hops. For a destination h hops away, it searches the layer $h-1$ filters in the NeighborTable to locate the next hop route. If multiple neighbors have routes to D with $h-1$ hops, a secondary metric such as link loss rate can be used to determine the optimal next hop.

Finally, we add several enhancements to optimize TARP for stable networks. First, TARP decouples link failure detection from routing updates, using small *heartbeat* broadcasts to detect mobility and link failures. Second, delta compression is used to further compress periodic broadcasts of routing state, each node only sends *diffs* of LayerTables, along with a version number for consistency. Third, to guard against failed broadcasts, each node acknowledges routing updates from all neighbors via a single *update acknowledgment filter* embedded in its own route update message. This filter is a Bloom filter summarizing the signatures of all neighbors who the node has received updates from in the last update period.

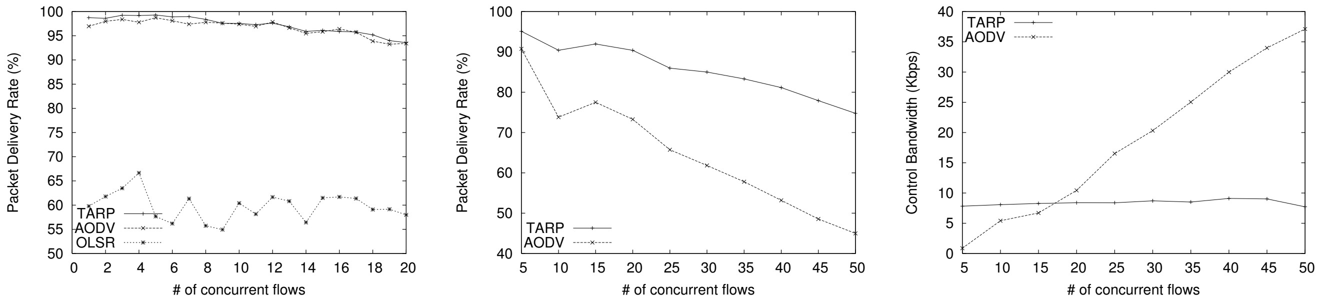
3.3 Scalability via Self-organized Beacons

To support efficient routing in larger networks, a TARP node can limit the number of Bloom filter layers kept in its LayerTable to L_{max}. To route to nodes more than L_{max} hops away, we use a distributed, self-organized variant of landmark routing. Beacons are self-elected nodes that publish its view of the local network region to the entire network. Beacons self-organize so that they are spread sparsely across the network, minimizing the number of beacon broadcasts. Each broadcast includes the beacon's LayerTable, allowing distant nodes to locate local nodes and acting as a breadcrumb trail into the local network. Once within L_{max} hops of its destination, a message uses local nodes to route.

This approach to "wide-area" routing is similar to landmark routing [13], but evenly balances traffic load away from the landmark node. In fact, beacon nodes in TARP forward the same amount of long-distance traffic as any non-beacon node in a region. This approach is also reminiscent of data location in structured overlay networks such as Tapestry [18], where distant clients route toward an object's root node, and diverge when a local path is found.

4. EVALUATION

For evaluation, we implemented a full version of TARP under Qualnet simulator 3.8 [17]. We evaluated TARP on Qualnet using a number of topologies. Here we focus on results from two static topologies, one with 100 nodes placed uniformly on a 1500x1500 meter grid, and a second with 500 nodes on a 3350x3350 grid. We compared TARP's performance against built-in implementations of AODV and



(a) Average PDR, 100 nodes (b) Average PDR, 500 nodes (c) Average Overhead, 500 nodes

Figure 3: A comparison of TARP, AODV and OLSR in a 100 node static topology on packet delivery rates, end-to-end delays, and bandwidth overhead of control messages.

OLSR-INRIA. Each simulation runs for 100 seconds, with an initial startup period to initialize routing state in all protocols. Transmission range is 250 meters, and each flow pushes 512 Byte CBR packets at 4 packets per second with a random source and destination. The performance of TARP can be tuned using a number of parameters. Our experiments used 4KB filter length, 12 layers per LayerTable, updates every 7 seconds and 2 heartbeats per second.

We plot the packet delivery rate (PDR) of TARP, AODV and OLSR in small static networks of 100 nodes in Figure 3(a). Both TARP and AODV perform well by delivering over 90% of all packets. TARP outperforms AODV by a very small margin, while OLSR maintains an average PDR around 60% [1]. While not shown here, our results also show TARP to outperform AODV and OLSR in end-to-end delay and message overhead.

We also compared our protocols in larger networks of 500 nodes. In our tests, between 4 to 5 nodes elected themselves to become TARP beacons. Figures 3(b) and 3(c) demonstrate the PDR and overhead, respectively, of TARP and AODV in the 500 node network. It is clear that even in a 500 node network, AODV's flooding mechanisms for route discovery and repair are generating large amounts of control traffic. The resulting broadcasts significantly impact both available bandwidth and packet delivery rates. AODV's overhead increases linearly with flows, rising to a factor of 4 above TARP's overhead at 50 flows. Consequently, AODV's PDR degrades rapidly with more flows, dropping to roughly half of TARP's PDR with 50 flows. While not shown here, our results also show that on average, TARP's end-to-end delay is lower than that of AODV by 33%. OLSR's results were significantly worse than that of AODV.

5. CONCLUSION

Proactive protocols are often seen as heavy-weight in comparison to reactive protocols like AODV. The Table Attenuation Routing Protocol (TARP) provides a novel way to limit routing state to a small constant while providing scalability via a self-organizing, load-balanced beacon system. Our work shows that proactive protocols are worthy of further consideration, particularly for static environments such as multi-hop mesh or sensor networks.

[1] Despite our efforts to tune OLSR, the OLSR-INRIA implementation in Qualnet had no tunable parameters.

6. REFERENCES

[1] Bloom, B. H. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Communications of the ACM 13*, 7 (July 1970), 422–426.

[2] Broch, J., et al. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proc. of MobiCom* (Dallas, TX, Oct. 1998).

[3] Caesar, M., Castro, M., Nightingale, E., O'Shea, G., and Rowstron, A. Virtual ring routing: Network routing inspired by DHTs. In *Proc. of SIGCOMM* (Sept. 2006).

[4] Dabek, F., Zhao, B., Druschel, P., Kubiatowicz, J., and Stoica, I. Towards a common API for structured P2P overlays. In *Proc. of IPTPS* (February 2003).

[5] Das, S. R., et al. Performance comparison of two on-demand routing protocols for ad hoc networks. *IEEE Personal Communications* (February 2001), 16–28.

[6] Fonseca, R., et al. Beacon vector routing: Scalable point-to-point in wireless sensornets. In *Proc. of NSDI* (Boston, MA, May 2004).

[7] Jacquet, P., et al. Optimized Link State Routing Protocol for Ad Hoc Networks. In *Proc. of IEEE INMIC* (Lahore, Pakistan, Dec. 2001).

[8] Johnson, D. B., and Maltz, D. A. Dynamic Source Routing in Ad-Hoc Wireless Networks. In *Mobile Computing*, vol. 353. Kluwer Academic Publishers, 1996.

[9] Karp, B., and Kung, H. T. Greedy perimeter stateless routing for wireless networks. In *Proc. of MobiCom* (2000).

[10] Kim, Y.-J., Govindan, R., Karp, B., and Shenker, S. Geographic routing made practical. In *Proc. of NSDI* (Boston, MA, May 2005).

[11] Leong, B., Liskov, B., and Morris, R. Geographic routing without planarization. In *Proc. of NSDI* (2006).

[12] Papapetrou, E., Pitoura, E., and Lillis, K. Speeding-up Cache Lookups in Wireless Ad-Hoc Routing using Bloom Filters. In *Proc. of PIMRC* (Berlin, Germany, Sept. 2005).

[13] Pei, G., Gerla, M., and Hong, X. Lanmar: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility. In *Proc. of MobiHoc* (Aug. 2000).

[14] Perkins, C., and Bhagwat, P. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proc. of SIGCOMM* (Oct. 1994).

[15] Perkins, C. E., and Royer, E. M. Ad-hoc On-Demand Distance Vector Routing. In *Proc. of WMCSA* (1999).

[16] Rhea, S. C., and Kubiatowicz, J. Probabilistic Location and Routing. In *Proc. of INFOCOM* (June 2002).

[17] Scalable Network Technologies. Qualnet Network Simulator, version 3.8, 2006. `http://www.scalable-networks.com`.

[18] Zhao, B. Y., et al. Tapestry: A global-scale overlay for rapid service deployment. *IEEE JSAC 22*, 1 (Jan. 2004).