Title

Ordering in Time: A New Routing Approach for Wireless Networks

Author

Garcia-Luna-Aceves, J.J.

Publication Date

2010-11-08

Peer reviewed

Ordering in Time: A New Routing Approach for Wireless Networks

Stephen Dabideen and J.J. Garcia-Luna-Aceves
Department of Computer Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064
Email: dabideen, jj@soe.ucsc.edu

***Abstract*—The ordering of nodes with respect to destinations of interest by means of spatial information (e.g., distances, path constituency, complete or partial topology) has been a fundamental aspect of all routing protocols in wireless networks. This spatial ordering has also included the use of geographical or virtual coordinates denoting the location of nodes. We propose the use of ordering of nodes based on time rather than space, and without the need to establish any clock synchronization among nodes. We demonstrate for the first time that using the relative times when each node receives and transmits packets is sufficient to establish multiple loop-free paths to destinations, and that such time-based ordering renders more efficient loop-free routing than the spatial ordering of nodes. With the use of self-adjusted delays, nodes can manipulate their ordering so that the resulting routing choices are more robust to failures than routing choices based solely on times driven by the physical topology. Furthermore, we show that the problem of resetting sequence numbers, which is a network-wide operation with traditional spatial ordering, is trivial with temporal ordering. We introduce the Time Ordered Routing Protocol (TORP) and compare it against routing protocols based on spatial ordering to demonstrate that temporal ordering can lead to superior performance in multi-hop wireless networks.**

I. Introduction

The successful dissemination of information in a packet switching network relies on the correct ordering of sources and relays with respect to destinations of interest. Many approaches have been proposed for routing in wireless networks over the past 40 years; however, as our summary of prior work in Section II shows, most of the previous work on routing in wireless networks has focused on the ordering of nodes with respect to destinations using spatial information, such as distances to destinations, path constituency, complete or partial topology, absolute location of nodes, or relative location with respect to special nodes.

While shortest path routing works well in wired networks, it is not very efficient in wireless networks especially in the face of mobility. For example, minimum-hop routing favors paths built with fewer links, some of which may span longer physical distances and hence render lower transmission quality. In a dynamic network, establishing multiple paths is more important than findng the single shortest path, given that link failures are inevitable and teh latency with which shortest paths may be attained.

We advocate the use of a temporal ordering as an alternative to the spatial orderings used in most routing protocols. We present a methodology to order nodes in a network based solely on the time they transmit and receive signaling packets. While there may be some resemblance between the resulting temporal ordering and the corresponding spatial ordering, temporal ordering is different and not completely dependent on a spatial ordering. Nodes have some control over when they choose to retransmit their packets and therefore have some control over the resulting ordering. This contrasts with spatial ordering, which is predetermined based completely on the topology and in some cases the use or quality of the links. We argue that this temporal ordering can increase substantially the number of paths between a source and its destination and therefore provide robust routing in the face of mobility. Unlike routing approaches based on link quality, our algorithm is light weight, quick to converge and effective in the face of mobility. Furthermore, all time measurements needed for the ordering of nodes are carried out individually by each node using its local clock, which obviates the need for any clock synchronization among nodes.

The use of a temporal ordering provides further advantages as we discuss in Section III. Because time is always increasing, the protocol can use non-sequential identification numbers to ensure that packet duplicates are not forwarded. Such numbers can be significantly smaller than traditional sequence numbers used in some routing protocols today, and can be reset locally. By contrast, sequence numbers used today must be drawn from large sequence number spaces, are aged out periodically (and hence require periodic transmissions by the origin of updates based on sequence numbers) and incur network-wide overhead to be reset.

Section IV presents the Time Ordered Routing Protocol (TORP) as a simple example of routing in wireless networks using time-based ordering. TORP orders nodes in two phases. Route request (RREQ) messages are used to build successor and predecessor relations in the network, and route reply (RREP) messages are used to order the relative priorities of the successors established by the RREQs. Routing is then based on the resulting ordering, which provides multiple paths to the destinations of interest and is therefore resilient to link failures.

Section V presents the results of simulation experiments showing that TORP outperforms traditional routing protocols based on spatial ordering (OLSR [5], AODV [18], DYMO [4]), a recent spatially-ordered protocol that provides multiple routes per destination (CaSH [7]), and the Authenticated Routing for Ad-hoc Networks (ARAN) [21] which uses limited time-based ordering. We varied both the load and mobility of the network and under all conditions, in terms of delivery ratio, CaSH was the closest in performance and TORP delivered around 25% more packets than CaSH with an even greater margin for OLSR, AODV, and DYMO. What is even more impressive is that this improvement did not come at the cost of overhead or delay. In fact, TORP had less than half the overhead of AODV, DYMO and ARAN, and had marginally more signaling overhead than CaSH in some scenarios; however, we must point out that much work remains to be done to make the signaling in TORP more efficient. The end-to-end delays and packet-delivery rates attained with TORP were much better than with the other routing protocols; while end-to-end delays with AODV were slightly better than with TORP, AODV delivered far fewer packets.

II. Related Work

The goal of any routing protocol is to ensure that a packet being forwarded gets closer to the destination after traversing each relay of a route. Thus, the most intuitive notion of spatial ordering is that of distances to destinations. The simplest notion of distance is the hop count in the route to a destination, and it has been employed in many routing protocols. However, it is well known that attempting to establish spatial ordering of nodes solely on the basis distances can lead to counting-to-infinity and looping problems. As a result, several approaches have advocated the use of destination sequence numbers to establish correct spatial ordering based on distances to destinations (e.g., DSDV [17], AODV).

There has been recent work aimed at eliminating the need for a distance metric for the establishment of spatial ordering in routing. Examples are destination-controlled, source-sequenced labeled routing protocol (DSLR) [20] and the sequence-number window routing protocol (SWR) [8]. In DSLR, nodes are ordered based on labels given by source-originated sequence numbers carried in route requests and these cached labels are used to relay route replies that set up the necessary successor-predecessor relationship. In SWR, nodes are given some leeway in choosing their sequence numbers for a given destination within a rage of values (a window) dictated by the sequence numbers of its neighbors in such a way that the sequence numbers assigned to nodes become larger as the destination is approached and thus mirror the spatial ordering of hop counts. The advantage of this approach is that it tends to allow more relays to provide intermediate replies than previous distance-vector routing protocols.

Yet another approach to spatial ordering is the use of path information. In some protocols, the data packets include the path to be traversed (e.g., DSR [9]) or the portion of the path that has been traversed (e.g., [22]) in order to eliminate routing loops, given that spatial ordering can be done easily by inspecting the nodes in the path stated in the packet header to determine if the current relay has been visited already. More recent routing protocols based on path information (e.g., the Feasible Label Routing protocol (FLR) [19]) eliminate the need for including path information in data packet headers by enforcing the lexicographic ordering of paths reported by nodes in their signaling messages.

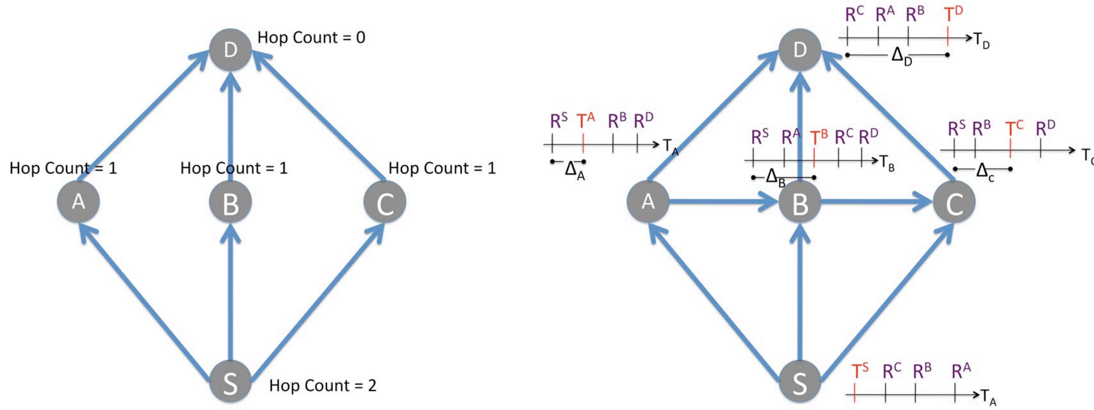
A very different approach to the use of spatial information consists of using the location of nodes. The recent availability of GPS devices allows the possibility for nodes to establish spatial orderings to destinations based on their geographic position and that of their destinations. Examples of this approach are the Greedy Perimeter Stateless Routing (GPSR) [10] and Weak State routing [1]. Most approaches based on geographical coordinates use greedy search algorithms that attempt to ensure that the packets being forwarded are geographically closer to the destination at every step of the search. However, the use of geographical coordinates and greedy heuristics can lead to local minima, and more sophisticated solutions are needed to solve this issue [11]. Furthermore, knowing the position of the destination beforehand remains a critical assumption in many of these protocols.

Estimated transmission count (ETX) [6] takes into account the number of transmissions needed to successfully transmit a packet to a one-hop neighbor then finds a path with the lowest ETX. This approach favors reliable paths without explicitly relying on the notion of distance. However, calculating reliable values for ETX requires several transmissions and while this may be suitable in static networks, as nodes become more mobile, more and more of these ETX values must be calculated as neighborhoods are constantly changing.

Another alternative to spatial ordering is the use of virtual structures or virtual coordinates that may be independent of physical distances. In protocols based on virtual coordinates, nodes maintain routes to fixed virtual neighbors and routing is based on this virtual topology. Examples include the virtual ring routing (VRR) [16] protocol and the virtual coordinate assignment protocol (Vcap) [3]. A virtual neighbor does not have to be physically close; in fact, in VRR virtual neighbors are based on node identifiers. Virtual coordinates replace geographic coordinates, which alleviates the problems of local minima [2].

To the best of our knowledge, the Authenticated Routing for Ad-hoc Networks (ARAN) [21] protocol is the only prior routing scheme that relies on time-based ordering. The authors of ARAN seek to add security to the protocol by eliminating a need for recorded distances. In ARAN, packets are routed along the quickest path from the source to the destination. This ordering, however, only creates a single path and would be strongly co-related to a distance based ordering such as AODV. Nodes immediately retransmit overhead packets. Thus the ordering in ARAN is more dependent on shortest path than it is on time.

Lastly, the use of multiple paths has been explored to improve performance in the context of routing based on spatial ordering. The protocols can use edge disjoint paths [23], node disjoint paths [14], or make no guarantees of uniqueness on paths used [13]. In many cases, the resulting performance



(a) Spatial Ordering (b) Temporal Ordering

Fig. 1. Spatial versus Temporal Ordering

increase comes at the cost of additional overhead needed to maintain multiple paths. To establish and maintain disjoint paths, path information may be needed, although less costly methods such as keeping track of the first hop in the path, as is the case with the Ad-hoc On demand Multi-path Distance Vector (AMODV) [15] routing protocol, can be sufficient. The use of multiple paths translates into shorter time to recover from link failures [13] as other paths may be readily available and even in use.

III. Time-Based Ordering for On-Demand Loop-Free Routing

A. Motivation

From our brief survey of prior work, we can observe that the establishment and maintenance of correct spatial ordering in wireless networks require the assignment of labels to nodes that somehow reflect the geometry of the network, such that routing loops are not created. To do so, labels are assigned to nodes according to different distributed algorithms that order such labels lexicographically (e.g., "12 $<$ 15", "BC $<$ CDA", and "[2, ABC] $<$ [3, EBC]"), which then allows relaying nodes to select as next hops to destinations those nodes with "smaller" labels without causing routing loops. These spatial orderings are usually not the most efficient solutions for mobile networks.

Temporal ordering can be established so as to maximize the number of paths from the source to a destination, rather than merely reflecting the geography of the network. In previous works [7] we demonstrated that the more links that are used in the loop-free ordering, the better the protocol performs. Consider the simple example illustrated in Figure 1. A spatial ordering using hop counts is used in Figure 1(a) while a temporal ordering is used in Figure 1(b). Nodes A, B and C have the same hop count in the spatial ordering and they cannot route through each other (to ensure loop freedom) unless the ordering is changed. However, in the temporal ordering, the probability that the transmission times of two nodes are the same is small. Therefore, Nodes A, B and C may be able to route through each other and this allows more paths than the corresponding spatial ordering. If links $A-D$, $B-D$ and $S-D$ fail, then there are no remaining $S-D$ paths in the spatial ordering but there is still a usable path in the temporal ordering. Furthermore, a temporal ordering is not as strictly constrained by the geography as a spatial ordering and allows nodes to manipulate the ordering to some extent (by the introduction of deliberate delays) to create more robust ordering.

An inherent limitation with spatial ordering is the need to use labels that must be stated explicitly by each node (e.g., a node states its distance and destination sequence number for a destination) and drawn from a finite name space. This results in some nodes having to "reset" their labels when (a) they are unable to find neighbor nodes that are spatially ordered with respect to some destination (e.g., a node with some distance and sequence number for a destination cannot find a neighbor with a larger sequence number for a destination or the same sequence number but smaller distance); (b) they recover from failures, having lost the routing state they had acquired previously; or (c) the labels reach a maximum and their values must be recycled. Most of the complexity in the signaling of routing protocols based on spatial ordering stems from this limitation. In contrast, routing based on temporal ordering can take advantage of the simple facts that time is unbound and always increasing, and every step taken in the routing process happens in time. Accordingly, the relative times when a node receives and transmits signaling packets can be used as the implicit labels for temporal ordering adopted by each node, and therefore as the basis of a new approach to on-demand loop-free routing.

B. Time-Based Ordering

To provide efficiency, temporal orderings must allow a node to acquire multiple paths to destinations without incurring excessive signaling overhead. An additional restriction we impose on temporal ordering solutions is that nodes should not rely on any absolute time derived from a global clock to determine the times used for ordering. This is because network-wide clock synchronization is as complex as ordering nodes in space, which would defeat the objective of using temporal ordering for routing.

According to our temporal ordering approach, a node classifies each of its neighbors as a potential successor, predecessor, or neutral with respect to a particular destination, and does this based on the times when it receives and relays route requests (RREQs), and then prioritizes each of its successors based on the time it receives the corresponding route replies (RREPs).

A node simply records the time at which it receives and transmits RREQS and RREPs according to its local clock. In Figure 1 R^X denotes the local time at which a RREQ was received from Node X and T^X denotes the local time at which Node X transmitted the RREQ. RREQs can be delayed to allow better ordering and in Figure 1, the delay at node X is denoted by Δ_X and its impact on the number of successors is clear.

Note that this ordering of neighboring nodes is based on a simple comparison of the transmission and reception times recorded using a local clock, and therefore clock synchronization is not required. In addition, because time only moves forward and RREQs propagate over time and space (i.e., the nodes), this approach enables the partial ordering of all the network nodes with respect to any destination for which a RREQ is issued.

Definition 1: Node A is a *successor* of Node B on a path to destination C if $R_B^A > T_B^B + \delta$ or if A is the destination, where R_B^A is the local time node B received a RREQ from A and T_B^B is the local time at which node B retransmitted the RREQ.

We assign a value δ as an upper bound on the transmission and propagation delay of RREQs over an active link. If $R_B^A > T_B^B + \delta$, then Node B can be certain that Node A transmitted the RREQ after Node B and that they both can agree on this. If the time difference is less than δ then Nodes A and B may not be able to accurately determine which route request was transmitted first, given that the packets may experience different delays. Consequently, a node may have neighbors that are neither successors nor predecessors. While the resulting partial ordering provides fewer potential paths from sources to destinations, it is far simpler to enforce in practice than a total ordering (for a given destination) in which a node only has successors and predecessors as neighbors. In our simulations, a value of 10ms assigned to δ was sufficient to ensure loop freedom and resulted in very few neutral neighbors. This ordering is further refined with the propagation of route reply messages as described in the next section.

Theorem 3.1: If nodes route data packets only through their *successors*, as defined in Definition 1, then there are no routing loops.

Proof: The proof is by contradiction. Assume for contradiction that data packets are always routed through successors and that there is a loop. Let nodes X and Y be two successive nodes in this loop. By Definition 1, (using the same notation) we know that $R_X^Y > T_X^X + \delta$. We also know that $R_X^Y < T_Y^Y + \delta$ since the packet must be received at most δ seconds after transmission. Therefore $T_Y^Y + \delta > R_X^Y > T_X^X + \delta$ hence $T_Y^Y > T_X^X$. This ordering relationship must hold for each successive pair of nodes in the path. Because the relation $>$ is transitive and Nodes X and Y are part of a loop, it follows that $T_X^X > T_Y^Y$, which is the desired contradiction. Thus, routing through successors as defined in Definition 1 cannot lead to routing loops. ■

IV. TORP

A. TORP Overview

TORP was designed for fast convergence of multiple loop free paths from a source to a destination on-demand, based on a temporal ordering. The mechanisms used to build this ordering are similar to many previous approaches used for spatial ordering and requires the flooding of control packets. Despite its simplicity, TORP delivers remarkable performance.

The role of route requests (RREQs), route replies (RREPs) and route errors (REERs) messages are fundamentally the same in TORP as other on-demand routing protocols; however, there are some important differences. Firstly, nodes record the local time at which they receive and transmit RREQs and RREPs and use this as the basis of ordering instead of a metric that must be carried in the packet. Secondly, nodes do not retransmit signaling packets immediately or even after a completely random delay, but instead wait for a calculated period so as to attempt to maximize the number of usable paths they attain.

Once a path is established it is proactively maintained by having the destination send periodic updates to rebuild a current time-based ordering.

B. Route Discovery

A RREQ consists of three fields: a destination address, a source address, and a request identification number. When a node needs a path to a destination for which it does not have a known route, it floods a RREQ using a request identification number not used in the last two hello intervals (this identification number will be discussed later). Upon receiving a RREQ, a node records the time it received the RREQ and the corresponding neighbor from which the RREQ was received. If it is a new RREQ, as defined by the request identification number, the node retransmits the RREQ after a small calculated delay Δ, and records the time of transmission. This results in the RREQ being disseminated throughout the network, as shown in Figure 2.

When a node receives a RREQ and is named as the destination in that RREQ, it does not retransmit the RREQ but instead issues a RREP. RREPs contain four fields: destination address, destination request identification number, source address, and source request identification number contained in the corresponding RREQ.

Definition 2: The Reply Acceptance Condition (RAC): A node can only accept and process a RREP if it is received from a successor, as defined in Definition 1.

Algorithm 1 TORP

```
HandleReqest(RREQ)
if NewIdentificationNumber then
    Record time and Identification Number
    if Node is the Destination then
        InitiateRREP()
    else
        RetransmitRREQ())
    end if
else
    if CurrentSourceIdentificationNumber then
        Record RREQ time
    else
        Ignore Packet
    end if
end if

HandleRouteReply(RREP)
if NewDestinationIdentificationNumber then
    if ReceivedfromaSuccessor then
        Record time and Identification Number
        RelayRREP(RREP)
    else
        Drop RREP
    end if
else
    if CurrentDestinationIdentificationNumber then
        Record Time and Identification Number
    end if
else
    Drop RREP
end if

HandleRERR(RERR)
Remove Route Table Entry
if NoRemainingSuccessor then
    TransmitRERR()
end if
```

Once a RREP is accepted, the node records the time and the neighbor from which it was received. Except for the source of the RREQ, the first time a node accepts a RREP with a new request identification number it retransmits the RREP.

The roles of RREQs and RREPs are intertwined in the ordering used in TORP. The propagation of RREQs is essential for a node to build successor-predecessor relationships with its neighbors according to Definition 1. These successor-predecessor relationships cannot be used until they are validated with a RREP. At the same time, a node cannot accept a RREP unless it comes from a successor defined by the propagation of RREQs.

Much like protocols such as AODV, when a source node issues a RREQ it sets a timer. If this timer expires and the source node is yet to receive a RREP, it issues a new RREQ with a different request identification number. For the duration of a data flow, destination nodes initiate proactive RREPs every 30 seconds. This allows for updating of the ordering of the nodes between the source and the destination. Pseudocode for some of the important functions of the TORP routing protocol is given in Algorithm 1.

C. TORP Example

A simple example of temporal ordering is given in Figure 2. Node S needs to send packets to Node D for which it does not have a route, so it initiates a RREQ. The solid arrows show the direction of propagation of RREQs, and the time at which the node transmits the route request is given. Assuming the

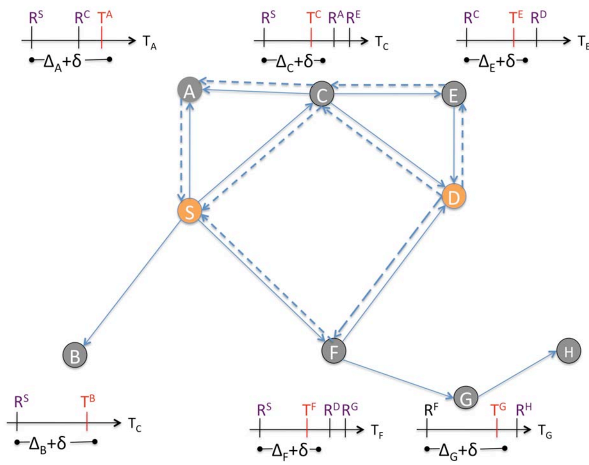


Fig. 2. Time-Based Ordering

network is connected, as is the case in this example, the RREQ is propagated to all nodes. A timeline showing the arrival time and transmission time of RREQs at each node is shown. R^X denotes the local time at which the RREQ was received from Node X and T^X denotes the local time at which the node transmitted the RREQ. Successors and predecessors can be easily identified by looking at the time RREQs were received relative to the node's transmission times, and is all based on the local clock.

Once Node D receives the RREQ, it initiates a RREP. The dashed arrows indicate that the node *accepted* a RREP from that neighbor. It is clear that the direction of the propagation of accepted RREPs is always opposite to the propagation of RREQs. Notice that RREQs are always propagated, but some nodes may never receive a RREP, such as Node B.

Node F would consider Node G a possible successor, but this relationship is never validated, because Node G does not receive a RREP from any of its possible successors (in this example, just Node H).

Interestingly, the delay (Δ) can affect the direction of the successor relationship between adjacent nodes, but never causes routing loops. If $\Delta_A > \Delta_C$, Node A would be classified as a successor for Node C, which is actually opposite to the ordering obtained from an ordering based on hop count. If this were to happen, the link from Node A to Node C would not be used, because the successor-predecessor relationship between nodes A and C would not be validated, given that Definition 2 would never be satisfied at Node A. However, in a larger network, it is more likely that Node A is part of a non-shortest path rather than no path at all.

The time-based ordering can lead to the establishment of non-shortest length paths in some instances, because of the value of Δ. However, it never leads to routing loops according to Theorem 3.1. However, this delay is necessary to reduce the probability that multiple adjacent nodes do not transmit within the δ period in Definition 1, which results in neutral relationships between nodes that is less preferable.

Lemma 1: Whenever a node retransmits a RREP, it is an indication that it had successors to the destination at the time the paths were initially set up.

Proof: Assume for contradiction that all nodes adhere to *RAC* and a given node X retransmits a RREP but does not have a usable path to the destination. All RREPs must originate at the destination, so the retransmitted RREP must have followed a path from the destination to Node X, and by the assumption, at some point in that path the link is not usable (because of the direction of the successor relationship). Say the link between Nodes Y and Z, where Node Y received the RREP from Node Z. All nodes obey RAC, and hence Y retransmitted the RREP after receiving it from Z. Therefore, Z must be a successor to the destination. Thus, the link from Y to Z is usable for routing to the destination, giving the desired contradiction. ■

D. Route Maintenance

When a link fails, a node can route data through any of its neighbors as long as they are successors, as defined by Definition 1. Among the potential successors, nodes are ordered by the time the RREPs are received, with the earliest RREP received time having the highest priority. As long as the destination is receiving packets, it periodically initiates proactive updates, set to 30 seconds. These updates refresh the ordering to better reflect the current topology of the network and provide alternate paths in the event of link failures.

If a node no longer has a path to the destination, either because of link failure or after receiving a route error message (RERR) from its last remaining successor to the destination, it issues a RERR indicating it no longer has a path to the destination. This RERR serves to prevent data packets from being routed through this node.

Theorem 4.1: TORP is loop-free at all times.

Proof: From Lemma 1 we know that a node must have a usable path to the destination when it receives and accepts a RREP in response to a RREQ it issued. Based on RAC and Theorem 3.1, this initial path must be loop-free. If a loop is formed, it must be after a node chooses a different set of successors. Assume for contradiction that all nodes obey RAC and yet such a loop forms. From Theorem 3.1, it follows that this can occur only when a node routes data through a node that is not a successor. However, this means that the node must accept a RREP from at least one node that is not a successor, which violates RAC and thus is a contradiction. Therefore, TORP must be loop-free at all times.

■

In MANETs, excessive overhead can degrade the performance of the network, so it is desirable to limit the propagation of signaling packets without disrupting route computations. In TORP, RREQs are flooded throughout the network, but RREPs are restricted to a region between the source and the destination. The RAC results in RREPs flowing in the opposite direction of the corresponding RREQs and within the region between the source and destination. For example, in Figure 2 Node G would have received a RREP from Node F (and only Node F), but F is not a successor; therefore, G does not retransmit the RREP and consequently Node H never receives that RREP.

E. Adjustable Ordering and Mobility in TORP

Nodes attempt to adjust the ordering of their neighbors so that they have almost equal number of successors and predecessors. This is done by adjusting the time at which they retransmit RREQs. By delaying the retransmission, a node is likely to increase the number of predecessors it has, and by speeding up the retransmission it is likely to decrease the number of its predecessors, thereby increasing the number of successors.

The value of the retransmission delay Δ is initially set to a different random value at each node using the node identifier as the seed. A node associates each destination with its own value of Δ, which is initially the same for all destinations at a particular node. Each time the node receives a new RREQ it updates the value of Δ for the corresponding destination. If the node had more successors than predecessors in the previous ordering, it multiplicatively increases the value of Δ such that $\Delta = 1.1 * \Delta$. In the situation is reversed, the value of Δ is multiplicatively decreased such that $\Delta = 0.9 * \Delta$. The value of Δ is bounded between 0.1ms and 0.6ms, using a modulo operation, to ensure the value does not become too large so as to hinder the performance of the network.

In a mobile environment, any ordering eventually becomes obsolete as nodes move out of range of each other. Rebuilding a new ordering is costly in terms of overhead and network resources and it is therefore critical to keep the ordering up-to-date and to minimize the number of times the ordering must be rebuilt. In TORP, the ordering is designed to increase the number of paths from the source to the destination; therefore, the ordering can tolerate a larger number of faults before it becomes obsolete. Also, the use of destination-driven periodic updates serves to update the ordering, detect broken links and update the neighborhood information.

F. Request Identification Number

As we have stated, many routing protocols based on spatial ordering use a strictly increasing sequence number to ensure loop-freedom. In contrast, the request identification number included in a RREQ in TORP merely labels the ordering rather than indicating its freshness. The purpose of request identification numbers in TORP is simply to limit the retransmission of signaling packets and to associate a RREP with the corresponding RREQ.

We assume that the propagation of a packet across the network takes less than the time between two periodic updates. TORP uses a request identification number between 0 and 511 (nine bits), and divides this space into two equal disjoint groups, call them group A and group B. Each time a node needs a request identification number for the first period of the network (thirty seconds), it chooses randomly one that has not already been used in that period from group A. For the next period (thirty seconds) the node uses request identification numbers from group B that have not already been used in that period. Each successive period, the node alternates between group A and B when selecting the request identification number, and these identifiers can be recycled at the start of each period.

Simply put, we use a small identification number space of nine bits, and this can be reset every minute. The alternating use of groups means that there is at least thirty seconds between the successive use of any request identification number and by our assumption there cannot be simultaneous propagation of two packets with the same request identification number that originated from the same source labeling different floodings for the same or different destinations. If a node receives a packet with a request identification number that it has not received in the last thirty seconds, then it must be a new packet from a new flood, rather than a packet from the previous flood with the same identifier.

This simple implicit reset mechanism represents a significant improvement over the network-wide resets needed for traditional sequence numbers.

Theorem 4.2: Resets of request identification numbers in TORP require no global procedures and cannot lead to routing loops.

Proof: TORP attempts to avoid loops by having a node forwarding packets only to successors that have used the same request identification number it has used. With a reset of the request identification number, it is conceivable to have fragments of two different DAGs with the same identification number. Assume for contradiction that there is a loop in TORP. Let us call the set of nodes in the newer DAG A and the set of nodes in the older DAG B (where both DAGs have the same identification number). Clearly there cannot be a loop if the data packet stays in DAG A alone or if it stays in DAG B alone, because each is a DAG. If there is a loop it must be the case that the packet is transmitted from DAG A to DAG B and then again from DAG B to DAGA, or vice versa. For the packet to be transmitted from A to B there must be two adjacent nodes, call them a and b, where $a \in A$ and $b \in B$ and such that b is a successor of a. This necessarily means that node b must have transmitted a RREQ after node a and with the same request identification number used by node a in its RREQ. However, node b is in the older ordering and node a is in the newer ordering; therefore, for node b to have transmitted the RREQ after node a, it must be in the newer ordering, which is a contradiction. Therefore, even if there are two DAGs with the same identification number in TORP, there cannot be routing loops.

■

By contrast, protocols that rely on strictly increasing sequence numbers are vulnerable to the type of routing loops addressed in Theorem 4.2, unless there is some global resetting of the sequence numbers (e.g., see [20]).

V. Performance

We use simulation experiments to show that the temporal ordering used in TORP achieves better performance than the spatial ordering of several other routing protocols, and that multiple successors are necessary for temporal ordering to be efficient under heavy load and high mobility.

We compare the performance of TORP to that of OLSR, AODV, DYMO, CaSH and ARAN. This represents a wide variety of proactive and reactive approaches to ordering in MANETs, based on distance, path vectors, link states, and even implicitly time. While some of these protocols are no longer state of the art, such as AODV, they are well known protocols and there is still some merit in the comparison. Also, CaSH is a very recent protocol that we have been shown to have good performance [7]. DYMO is the successor of AODV and its relative performance is therefore of particular interest. OLSR is a link state routing protocol where all nodes attempt to maintain up-to-date routing information to all other nodes in the network. This ordering is based on physical characteristics of the network, but gives a complete ordering through all nodes rather than a subset as is the case in on-demand routing protocols.

AODV uses hop counts and destination-based sequence numbers to establish spatial ordering of nodes along a single path between the source and the destination. DYMO improves over AODV with the use of path vectors to make the protocol more resilient to path failures. The Constrained Scalable Hybrid (CaSH) routing protocol [7] establishes multiple paths from the source to the destination in an on-demand manner and proactively updates the multi-dimensional ordering of nodes, which tends to create more paths than a spatial ordering based on a single metric. The key difference between CaSH and TORP is that CaSH uses spatial ordering while TORP uses temporal ordering, and both provide multiple paths to destinations. Hence, their comparison yields key insights on the advantages of temporal ordering of nodes. As we have stated, ARAN [21] establishes a single path using temporal ordering and hence its comparison with TORP illustrates the importance of providing more than one successor to a destination at each node.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Simulation Time	900s
Number of Nodes	100
Simulation Area	1000m x 1000m
Node Placement	Uniform
Mobility Model	Random Waypoint
Min-Max Speed	1-10m/s
Pause Time	30s
Propagation Model	Two-ray
Physical Layer	802.11
Antenna Model	Omnidirectional
MAC Protocol	802.11 DCF
Data Source	CBR
Number of Packets per Flow	400
Packet Rate	4 packets per second
Number of Flows	50

The simulations were performed using the Qualnet 4.5 network simulator. Two scenarios were used and the parameters

TABLE II
VARIED SIMULATION PARAMETERS

Parameter	Scenario A	Scenario B
Number of Nodes	100	100
Simulation Area	1000x1000m	1000x1000m
Radio Range	150m	200m

are summarized in Table I. The first scenario was designed to test the performance of the protocols in a dynamic environment with volatile links. This choice of parameters satisfies the minimum standards for rigorous MANET protocol evaluation as prescribed in [12], because it results in an *average shortest path hop count* [12] of 4.03 and *average network partitioning* [12] of 3.9%. This ensures that packets travel several hops from source to the destination and thus tests the robustness of the protocols.

Scenario A uses a radio range of 150m, and was designed to stress test the protocol, ensuring that packets travel several hops on average before they arrive at the destination. Consequently, the results are less favorable than many published papers, which test these protocols in less strenuous environments, and in many cases with routes of only two hops, for example. We emphasize that these rigid parameters are necessary to show the true performance potential of the routing protocols in practice. We test the protocols under a wide variety of loads and mobility to demonstrate the performance of TORP under all conditions.

Scenario B uses a greater radio range, 200m, to add more stability to the links and create more paths between the source and the destination. Consequently, the average network partitioning as well as the average shortest path hop are reduced. These parameters do not adhere to the minimum standards required to test MANETs, but deliver results more comparable to previous works.

Each experiment lasted 900s and for each protocol the experiment was repeated 20 times with random node placement and mobility. In each experiment, constant bit rate (CBR) sources were used, which started generating packets at a random time to a randomly chosen destination. Each CBR source generated 400 packets at a rate of 4 packets per second. The parameters are summarized in Tables I and II.

Three metrics were used to evaluate and compare the performance of the protocols. Delivery ratio is the fraction of packets that arrive at the corresponding destination by the end of the simulation. Latency is the average end-to-end delay experienced by the data packets. Overhead is the number of control packets (RREQs, RREPs, RERRs, Hellos, and TC messages) which were initiated or forwarded, divided by the number of data packets sent. This last metric gives an indication of the average number of control packets needed to send a data packet from the source to the destination.

The results of the experiments with 100 nodes and varying flows are given in Table III, Table IV and Table V.

TABLE III
SIMULATION RESULTS WITH 20 FLOWS: SCENARIO A

	Delivery Ratio	Latency	Overhead
AODV	0.64 ± 0.06	0.05 ± 0.01	12.4 ± 2.5
OLSR	0.22 ± 0.05	0.03± 0.01	16.2 ±0.3
ARAN	0.62 ± 0.07	0.27 ± 0.12	17.7 ± 4.2
CaSH	0.85 ± 0.06	0.08 ± 0.03	5.7 ± 1.3
TORP	0.88 ± 0.05	0.07± 0.01	4.9 ± 2.2
DYMO	0.67 ± 0.04	0.06 ± 0.01	2.8 ± 2.2
DSR	0.15 ± 0.02	0.30 ± 0.20	2.9 ± 0.6

TABLE IV
SIMULATION RESULTS WITH 40 FLOWS: SCENARIO A

	Delivery Ratio	Latency	Overhead
AODV	0.60 ± 0.07	0.08 ± 0.2	15.9 ± 3.1
OLSR	0.13 ± 0.03	0.03± 0.02	16.2 ±0.2
ARAN	0.54 ± 0.07	0.32 ± 0.11	19.9 ± 2.2
CaSH	0.75 ± 0.06	0.15 ± 0.04	6.5 ± 1.1
TORP	0.83 ± 0.03	0.10± 0.01	6.1 ± 1.0
DYMO	0.62 ± 0.04	0.08 ± 0.01	3.2 ± 2.2
DSR	0.11 ± 0.03	0.25 ± 0.23	3.2 ± 0.7

A. Delivery Ratio

The fraction of packets successfully delivered to the intended destination is a good measure of the effectiveness of a routing protocol. Although ARAN is based on time ordering, it only sets up one path from the source to the destination and its performance is worse than that of AODV and DYMO for all numbers of flows. OLSR performed significantly worse than the other protocols. This reflects the looping problems associated with link state protocols in mobile environments. Once there is a change, some time is required for convergence and until then data packets can be sent in loops and eventually dropped after the TTL is expired.

CaSH improves on AODV by setting up multiple paths and proactively updating the ordering and thus delivers more data packets than AODV under all load scenarios. Although CaSH uses a multi-dimensional spatial ordering, which is better than simple hop-count ordering, the temporal ordering in TORP results in better performance than that of CaSH.

The fraction of packets delivered depends on the protocol's ability to find a path to the destination and repair it when it breaks. Protocols like ARAN and AODV require more complex procedures to repair paths than CaSH or TORP, which may have alternative paths readily available. Packets can also be lost if they are routed in a congested area of the network and are excessively delayed. In TORP, the faster paths are prioritized and these would be through less congested nodes, unlike spatially ordered protocols which ignore the current network conditions. As the number of flows increases, congestion increases and it can be seen in Tables III, IV and V that the performance of all the protocols decreases and the performance of CaSH degrades at a much faster rate than of TORP. This reinforces the idea that the temporal ordering is better in the face of congestion.

B. End-to-End Delay

The main factors that affect the end-to-end delay are the time taken to set up a path from the source to the destination and how often this must be done. Proactive protocols are

TABLE V
SIMULATION RESULTS WITH 50 FLOWS

	Delivery Ratio	Latency	Overhead
Scenario A			
AODV	0.58 ± 0.05	0.09 ± 0.02	17.8 ± 4.5
OLSR	0.10 ± 0.05	0.03± 0.01	16.3 ±0.2
ARAN	0.48 ± 0.07	0.31 ± 0.08	22.2 ± 5.0
CaSH	0.71 ± 0.06	0.17 ± 0.065	7.1 ± 1.7
TORP	0.80 ± 0.07	0.11± 0.02	7.1 ± 2.5
DYMO	0.38 ± 0.09	0.24 ± 0.07	5.53 ± 0.9
DSR	0.05 ± 0.02	0.99 ± 0.3	3.4 ± 0.23
Scenario B			
AODV	0.78 ± 0.05	0.07 ± 0.02	7.9 ± 1.9
OLSR	0.23 ± 0.03	0.03 ± 0.0‘	12.6 ± 0.2
ARAN	0.82 ± 0.07	0.17 ± 0.05	13.4 ± 4.0
CaSH	0.93 ±0.05	0.06 ± 0.02	4.0 ± 0.46
TORP	0.95 ± 0.03	0.07 ± 0.01	2.13 ± 0.6
DYMO	0.53 ± 0.07	0.24 ± 0.08	5.2 ± 1.0
DSR	0.06 ± 0.04	1.2 ± 0.7	3.6 ± 0.5

appealing for the promise of low delays and this is shown in the results of OLSR which has much lower delays that the other protocols. However, this must be taken in the context of extremely low delivery ratio. All the on-demand routing protocols tested use a similar method of flooding RREQs and waiting for a RREP to set up the path. Once a link breaks, AODV immediately initiates the route discovery process and finds a new path. This results in low delay. TORP and CaSH, on the other hand, use the existing ordering to try to find new paths. The ordering changes due to mobility and nodes may believe they have a path to the destination but not discover the path is no longer valid until they try to use it. Therefore, there are two opposing factors affecting the end-to-end delay in these protocols. If there are indeed alternative paths to the destination, the delay would be smaller. However, if the node must try several paths that no longer lead to the destination, then the delay could be larger. Furthermore, AODV and DYMO deliver fewer packets to the destination and the end-to-end delay is reported based only on the packets delivered. This can give a false impression of the protocol because it could be that only packets that need to travel a short path to the destination.

This is especially true in OLSR, which has a very low delivery ratio but also the lowest end-to-end delay. Congestion also affects the delay, and because TORP favors faster paths, which are also less congested, it enjoys lower end-to-end delay than CaSH.

C. Overhead

OLSR is proactive and, not surprisingly, incurs even more overhead than most of the on-demand protocols tested. The overhead of OLSR will not change much with increased loads or mobility as it is mostly periodic updates. It is interesting that AODV and ARAN incur even more overhead than OLSR and this is a result of the frequency at which they flood the network. Protocols that set up only a single path, such as ARAN and AODV require significantly more overhead than CaSH and TORP, because each time the path breaks the network is flooded. Both CaSH and TORP use proactive periodic updates initiated by the destination to update the ordering of the nodes in the network. Both CaSH and TORP use proactive periodic updates initiated by the destination to update the ordering of the nodes in the network. In CaSH, these updates are restricted to a region of interest with diameter bounded close to the length of the shortest path between the source and destination. In TORP, the propagation of RREPs is also restricted but to a much lesser extent than in CaSH, because there is no notion of a distance between the source and destination. However, the temporal ordering in TORP is more resilient to link failures than the spatial ordering in CaSH and therefore there is less need to flood to discover new paths. Although TORP has much less overhead that AODV, CaSH incurs even less overhead than TORP in scenarios with larger number of flows.

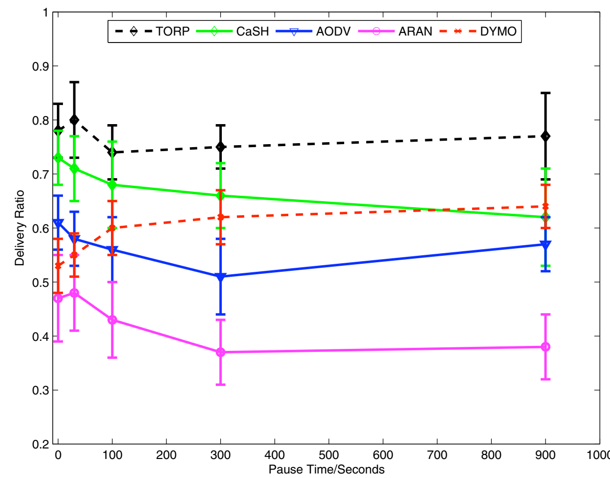


Fig. 3. Delivery Ratio with varying Pause Time

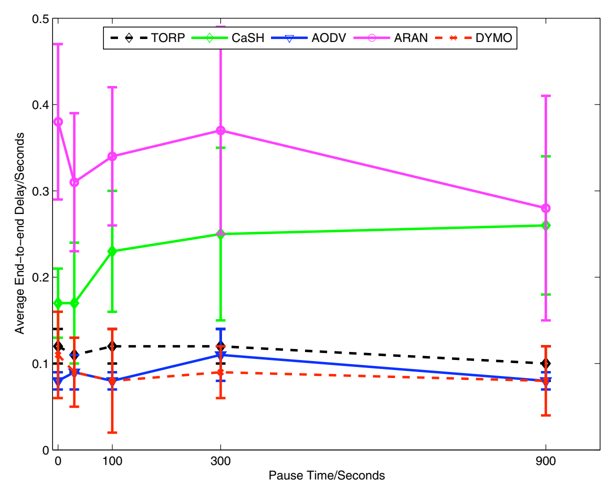


Fig. 4. Delay with varying Pause Time

D. Performance with Mobility

We show the performance of the protocols as the mobility of the nodes is varied. We use the parameters of Scenario A with 100 nodes and 50 sources and vary the pause time

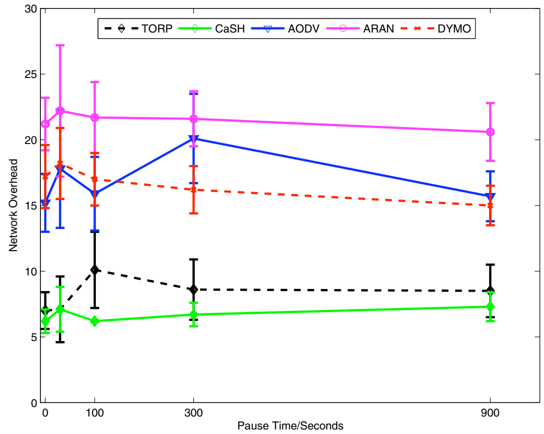


Fig. 5. Overhead with varying Pause Time

of the node from 0 seconds to 900 seconds. The results are shown in Figure 3, Figure 4, and Figure 5. We show that TORP outperforms the other protocols for a wide range of pause times. In some cases, the protocols experience worse performance at lower mobility. This can be the result of routing loops. In a mobile environment, loops will often be broken as node move out of range of each other. In a more stationary network however, once a loop is formed, it can cause more damage before it is discovered. The delivery ratio of the protocols are shown in Figure 3. There is some variation as the mobility changes, but TORP achieves the best delivery ratio and the relative performance of the other protocols is the same. In terms of end-to-end delay, only AODV and DYMO achieved better performance than TORP for the same reason as explained above. There was only small variation in the overhead as the pause time was varied. CaSH enjoyed the lowest overhead and that is mainly because of the restricted propagation of RREPs.

VI. Conclusion

We introduced the concept of time-based ordering as an alternative to the spatial ordering approaches that have been used in routing protocols for wireless networks since the DARPA packet radio network was developed. We have described the inherent advantages of temporal ordering over spatial ordering in the design of routing protocols, such as allowing more paths, factoring in network conditions implicitly, and eliminating the need for resets of parameters used in the ordering. We introduced the Time Ordered Routing Protocol (TORP) as an example of the potential of this new type of ordering and showed that it performs better than the traditional approaches based on spatial ordering.

VII. Acknowledgments

This research was partially sponsored by the U.S. Army Research Office under grants W911NF-05-1-0246 and SUB0700155 / SC20070363, and by the Baskin Chair of Computer Engineering. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] U. Acer, S. Kalyanaraman, and A. A. Abouzeid. Weak State Routing for Large Dynamic Networks. *Proc. MobiCom*, September 2007.

[2] Q. Cao and T. F. Abdelzaher. A scalable logical coordinates framework for routing in wireless sensor networks. *IEEE Real-Time Systems Symposium*, 2004.

[3] A. Caruso, S. Chessa, S. De, and A. Urpi. Gps free coordinate assignment and routing in wireless sensor networks. *IEEE Infocom*, 2005.

[4] I. D. Chakeres and C. E. Perkins. Dynamic manet on-demand routing protocol. *IETF Internet Draft*, 2008.

[5] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol(OLSR). *RFC 3626*, October 2003.

[6] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. *Proc. MobiCom*, 2003.

[7] S. Dabideen and J. J. Garcia-Luna-Aceves. Multidimensional routing. In *ANC 08: IEEE Workshop on Advanced Networking and Communications 2008*, 2008.

[8] J. Garcia-Luna-Aceves and H. Rangarajan. A new framework for loop-free on-demand routing using destination sequence numbers. *Proc. IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2004.

[9] D. B. Johnson, D. A. Maltz, and J. Broch. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In C. E. Perkins, editor, *Ad Hoc Networking*, chapter 5, pages 139–172. Addison-Wesley, 2001.

[10] B. Karp and H. Kung. Greedy perimeter stateless routing for wireless networks. *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 243–254, August 2000.

[11] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Geographic routing made practical. *USENIX/ACM Symposium on Networked System Design and Implementation*, 2005.

[12] S. Kurkowski, T. Camp, and W. Navidi. Minimal Standards for Rigorous MANET Routing Protocol Evaluation. *Technical Report MCS 06-02, Colorado School of Mines*, 2006.

[13] S.-J. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. *IEEE Internation Conference on Communications*, 2001.

[14] X. Li and L. Cuthber. On-demand node-disjoint multipath routing in wireless ad hoc networks. *IEEE International Conference on Local Computer Networks*, 2004.

[15] M. K. Marina and S. R. Das. On-demand multi path distance vector routing in ad hoc networks. *Proceedings of the Ninth International Conference on Network Protocols*, 2001.

[16] M. C. Miguel, M. Castro, E. B. Nightingale, and A. Rowstron. Virtual ring routing: Network routing inspired by dhts. *ACM Sigcomm*, pages 351–262, 2006.

[17] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Proceedings SIGCOMM '94*, pages 234–244, August 1994.

[18] C. E. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, February 1999.

[19] H. Rangarajan and J. Garcia-Luna-Aceves. Using labeled paths for loop-free on-demand routing in ad hoc networks. *Proc. ACM MobiHoc*, 2004.

[20] H. Rangarajan and J. Garcia-Luna-Aceves. Loop-free on-demand routing using source-sequence numbers. *Proc. IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2005.

[21] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. Belding-Royer. A secure routing protocol for ad hoc networks. In *Proceedings International Conference on Network Protocols*, 2002.

[22] M. Spohn and J. Garcia-Luna-Aceve. Neighborhood aware source routing. *Proc. The ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2001.

[23] K. Zeng, K. Ren, and W. Lou. Geographic on-demand disjoint multipath routing in wireless ad hoc networks. *Military Communications Conference*, 2005.