MARio: Mobility-Adaptive Routing Using Route Lifetime Abstractions in Mobile Ad Hoc Networks

Yong Wang **Margaret Martonosi** **Li-Shiuan Peh**
yongwang@cs.princeton.edu *mrm@ee.princeton.edu* *peh@ee.princeton.edu*
Depts. of Electrical Engineering and Computer Science, Princeton University, NJ 08544

Mobile Ad hoc NETworks (MANETs) often have fast-changing and unpredictable mobility. This is especially true for recently emerging mobile sensor networks [7] and delay tolerant networks [2], where extreme network conditions are prevalent. To better adjust to such network conditions, we propose to deconstruct the underlying node mobility into a set of abstractions that can be readily leveraged for driving dynamic routing decisions. In particular, we investigate the feasibility of using route lifetime to abstract mobility. Through extensive simulations on DSR, we demonstrate up to 31% improvement in packet delivery rate and 53% improvement in average end-to-end packet latency, using our proposed route prefetch and decay techniques that leverage knowledge of route lifetimes.

I. Introduction

With the emergence of new applications in sensor networks [7] and wireless ad hoc networks [2], the performance of data routing (including data delivery, data dissemination and sensor reprogramming) under varying degrees of mobility has become more challenging and pressing. Previous work mainly views mobility as a black box: a constant characteristic of the system. Decisions regarding when to perform route discoveries and maintenance operations are based on "magic numbers". These constants are normally hand-tuned for typical scenarios and remain fixed after protocol deployment.

In real life, however, mobile networks display different degrees and types of mobility over time. This can lead to inefficiencies for protocols using pre-programmed constants, due to their inability to react to such dynamic changes. For these protocols, decisions concerning route request and route maintenance are made implicitly — with protocol designers fixing parameters based on some expected mobility and workload to address. For a routing protocol to be efficient in such environments, it needs to dynamically and automatically adapt to situation changes, especially mobility changes.

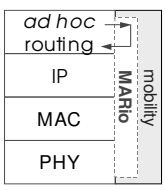


Figure 1: Protocols can exploit mobility knowledge through the interface provided by MARio.

In this paper, we propose MARio, a mobility-adaptive routing scheme that enables situation-aware protocols to adapt to varying mobility changes by learning and exposing mobility knowledge. Figure 1 illustrates a network stack with MARio support. Generally, MARio provides a plane of knowledge about node mobility. One can readily derive mobility changes by constantly monitoring the information exposed by MARio in the form of various abstractions, such as recent route lifetimes and disconnection intervals. In particular, we propose exposing node mobility using the abstraction of *route lifetimes* that captures the generational behavior of node mobility in a way that applies to any mobility model and can be exploited directly by protocols.

The key challenges of using MARio are two-fold: (1) **Accuracy:** The accuracy of estimating route lifetimes is crucial to the performance of our approach. We discuss this here and provide practical solutions. (2) **Overhead:** Since network resources are highly constrained, the overhead of gathering route lifetimes needs to be small.

In the following discussions, we use DSR [6] as a case study to demonstrate that to achieve good routing performance under varying degrees of mobility changes, knowledge of underlying mobility should be leveraged. Specifically, our proposed optimization uses the route lifetime statistics exposed by MARio and issues route maintenance operations, such as route prefetch and route invalidation (decay) selectively and proactively, whenever such operations are needed. The benefits are two-fold: (1) it saves network resources by selectively triggering operations, while still capturing potential proactive opportunities, and (2) most importantly, it allows a single protocol to cover a broad range of mobilities without hand-tuning parameters.

While we demonstrate the efficacy of MARio using DSR, the idea of MARio is general enough that it can be readily used with other ad hoc routing protocols as

well.

Related Work. PATHS [9] also studies path-related metrics. However, they focus on how statistics of path duration vary with different mobility models; we focus on exploiting such statistics, such as route lifetimes, in an adaptive protocol design.

Our approach is fundamentally different from the recently proposed mobility control approaches [3, 10, 8]. Mobility control requires hardware support for controlling the movement of nodes to achieve required routing properties, which has a very different design space. MARio does not alter node movements. It simply monitors and harnesses them for protocol adaptation.

The work described in [4] focuses on constructing an empirical mobility model based on movement data. In the development of their model, different metrics and statistics of mobility are studied to reconstruct a theoretical model. However, we focus on metrics that reflect important mobility knowledge and that can directly be leveraged by routing protocols.

II. Understanding Route Lifetimes

Definition 1 (Definition of route lifetime) *For any multi-hop path $P(n_1, \cdots, n_k)$ consisting of k nodes connecting source n_1 and destination n_k, with n_{i+1} the next hop of n_i, the route lifetime of $P(n_1, \cdots, n_k)$ is the time from when the path first exists to when any link (n_i, n_{i+1}) in the path breaks.*

Harnessing Route Lifetime: Route Prefetch and Decay. Given statistics of route lifetime in the past, MARio can predict the lifetime of routes currently in use. A routing protocol can then leverage such generational knowledge and perform various route maintenance operations just-in-time. In particular, we propose two optimizations to illustrate how to harness route lifetime knowledge in a MANET. They are route prefetch and route decay, described as follows:

Route prefetch involves loading a route into the route cache before it is needed. However, prefetching too early will take up precious route cache capacity, possibly replacing a valid route. Prefetching too late, on the other hand, will not hide much of the route discovery latency. Unused prefetched routes also introduce unnecessary route discovery traffic into the network.

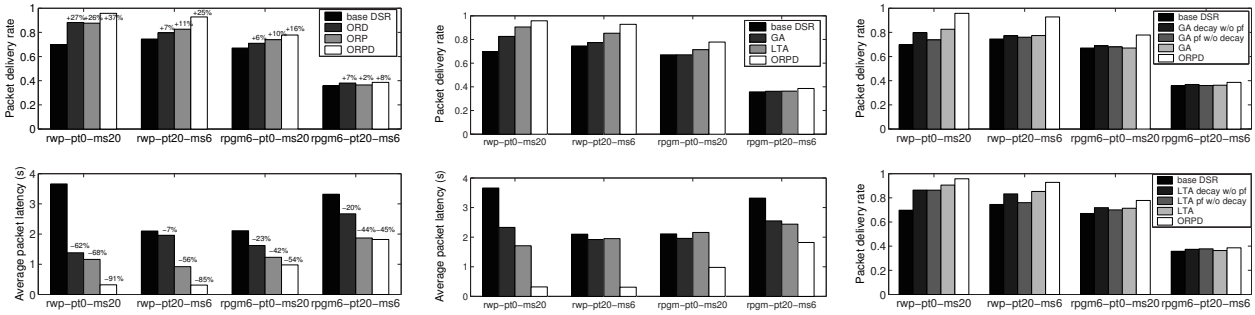
Conversely, *route decay* operates on the intuition that when a route is about to break, we can invalidate it just-in-time to make room for new routes, and avoid unnecessary traffic and packet drops imposed by following stale routes. One potential pitfall of route decay is that it could invalidate routes too early, introducing extra route discoveries. Another is that it could decay routes that may revive and continue to be valid after a short disconnection. Therefore, the accuracy of route lifetime estimations is crucial to the efficacy of route prefetch and decay.

Before we go into the details of optimizations, it is worthwhile to study their performance potential using ideal route lifetimes. This will give us an upper bound of performance gains by using route lifetime knowledge. Specifically, we study three oracle protocols with varying degrees of route lifetime knowledge: Oracle Route Decay without prefetch (ORD), Oracle Route Prefetch without decay (ORP) and Oracle Route Prefetch and Decay (ORPD). We use two performance metrics for this study: packet delivery rate and data latency.

Simulations are carried out in *ns-2*. Since no single mobility model is representative of all mobility characteristics, we looked at two different mobility models, the Random WayPoint model (RWP) and the Reference Point Group Mobility model (RPGM) [1], to evaluate the feasibility of MARio across different models. Each mobility scenario is denoted as [rwp,rpgm]-pt$X$-msS_{max}, where X is the pause time in seconds, and S_{max} the maximum speed. Constant Bit Rate (CBR) traffic using UDP is used for all simulations. Packet injection rate is fixed at 4pkts/s with 10 connections out of 30 mobile nodes. The nodes are randomly distributed across an area size of $1100\times1100m^2$ initially. All simulations are run for 1900s. We choose a warm-up period of 1000s because with a warm-up time larger than the longest possible route lifetime, the chance of tracing routes with different possible length are equal. The last 900s includes 890s data traffic and 10s cool-down time. Performance metrics are only collected for the last 900s. All subsequent simulations, if not specified, use the same setups discussed here.

Figure 2(a) demonstrates that perfect route lifetime knowledge can lead to substantial performance improvements for various mobility scenarios when harnessed appropriately. Prefetch is essential for reducing packet latency for both high and low mobility scenarios. Decay is needed for delivering more packets in a mobile environment. Combining prefetch with decay using ideal route lifetimes can potentially improve latency by 91% and packet delivery rate by 37%. With the RPGM model, the improvements are not as large as those for RWP. This is because nodes in RPGM model move in groups and the data routing between groups experiences frequent disconnections, which makes leveraging route lifetimes less beneficial.

Route Lifetime Gathering. For MARio to be useful at system runtime, it needs a light-weight route lifetime gathering method. We propose a *sampling* method based on DSR as follows: it tracks every route that has ever been discovered by DSR, with a route cache large enough that no route is ever displaced. Once a route is discovered, it stays in the route cache.



(a) Potential improvements of oracle approaches for different mobility models.

(b) Performance comparisons of realistic approaches to oracle, for different mobility models.

(c) Performance breakdown of GA and LTA.

Figure 2: Oracle and realistic performance comparisons.

Using a special tag in the route cache structure that tells a "shadow " route (routes that exist just for the purpose of measurement) from active routes used for routing, no adverse effect from a stale shadow route will be introduced.

These routes then form the sampled set from which we periodically monitor route lifetimes. By sampling, we avoid measuring all potential routes; the number of these could explode exponentially[1]. The differences between sampling and an exhaustive approach are negligible from our validation. Due to space constraints, we omit the empirical validation results here.

III. Optimizations using Route Lifetimes

III.A. Details

We have demonstrated that by leveraging route lifetime knowledge appropriately, route prefetch and decay can improve packet delivery rate and reduce packet delivery latency. The key challenge is to determine **when** to issue prefetch/decay operations. We study two predictors for route lifetimes: (1) **Global Average (GA):** the route lifetime averaged across *all* nodes with all routes within a node for the entire simulation time. (2) **Local Temporal Average (LTA):** the route lifetime averaged across routes within a single node for one time window.

Basic Protocol. The gist of our protocol is to start prefetching a route when we expect the routes in the cache will soon fail, based on the knowledge of past route lifetimes. Similarly, a route can be decayed just-in-time, near the end of its expected lifetime. Algorithm 1 lists the pseudo-code of the basic protocol, which is the basis for the offline and online protocols discussed next. ph_th denotes the prefetch threshold and $decay\_th$ the decay threshold respectively. Their values control the timeliness of prefetch and decay decisions.

[1]Actually it is not necessary to track all potential routes for a source-destination pair because only routes in the route cache could have an impact on the routing behavior.

Algorithm 1 Pseudo-code of the basic protocol.

```
1:  for every mobile node do
2:    if a new route is discovered and entered into the route cache then
3:      route_entry_time ← now
4:    end if
5:    loop {Periodically check each route in its route cache}
6:      dest ← this route's destination
7:      if (now − route_entry_time > pf_th) then
8:        if there is no overlapping route request for dest then
9:          if now ≥ next (next is the next prefetch time based on
            route discovery backoff) then
10:           Initiate a prefetch for a new route with the same source
              and destination;
11:           update next
12:         end if
13:       end if
14:     end if
15:     if (now − route_entry_time > decay_th) then
16:       Invalidate this route as it is likely to be stale already
17:     end if
18:   end loop
19: end for
```

Offline Approach Using GA. The offline protocol is built upon the basic protocol and determines its decay and prefetch threshold based on values calculated through offline mobility trace analysis as follows: $decay\_th = GA$ and $pf\_th = decay\_th - T_{rd}$, where GA is the global average of route lifetimes and T_{rd} the average route discovery latency, which can be readily achieved by monitoring all route discovery latencies.

The problems with an offline approach are twofold. First, even for a mobility model as simple as RWP, the route lifetimes distribution could have large spatial and temporal variance. It is thus very difficult for GA to approximate the route lifetime accurately. Such a single approximation could lead to losses in optimization opportunities when most route lifetimes are actually much shorter than GA and wrong decisions when most route lifetimes are longer than GA.

Second, the offline algorithm is sometimes impractical because it needs *a priori* knowledge of route lifetimes average which is not always available.

Online Approach Using LTA. To capture the diverse spatial and temporal variation inherent in a mobile network, we next look at a more accurate and realistic route lifetime predictor, the LTA predictor. This approach gathers its local route lifetimes and calculates LTA separately for each node, without exchanging any route lifetime knowledge between nodes. It also captures the temporal variation by dividing the running time into consecutive *windows* and using Exponentially Weighted Moving Average (EWMA) to smooth estimations across *windows*: $Ave_t = \alpha \times Ave_t + (1 - \alpha) \times Ave_{t-1}$, with α the *EWMA factor*.

The route prefetch and decay thresholds for the online approach are calculated in a similar way to the offline approach. The only difference is that the decay threshold for each window is determined by the current moving average LTA.

More sophisticated predictors, such as Markov predictors or Bayesian predictors, are attractive and potentially have much higher accuracy. However, in a mobile environment, it is desirable to have a simple and light-weight predictor with reasonable accuracy. EWMA, though simple, has proved to be an effective candidate for our purpose. We leave it as an interesting future work to study the feasibility of sophisticated predictors for such purpose and the sensitivity of routing performance to the accuracy of predictors.

III.B. Performance Evaluation

In the following discussions, we refer to the offline protocol as GA and online protocol as LTA. We evaluate the performance of GA and LTA on four different mobility scenarios, which show different mobility characteristics and oracle performance. We use a window size of 100s and an EWMA factor (α) of 0.5 in our evaluation. This factor gives the same weight to routes ending in the most recent window slot as to all routes ending in previous window slots.

As Figure 2(b) shows, the performance of LTA is close to ORPD (Oracle Route Prefetch and Decay), with up to 31% improvement in packet delivery rate and 53% improvement in average packet latency, with little increase in routing overhead over that of DSR. Figure 2(c) demonstrates the performance breakdown of prefetch and decay for both GA and LTA. The purpose here is to understand the individual contribution of each technique to the overall performance.

For all the mobility scenarios, both GA and LTA decay without prefetch have a higher packet delivery rate than that of baseline DSR. The improvement for rwp-pt0-ms20, however, is much higher than for rwp-pt20-ms6. This is because rwp-pt0-ms20 has more stale routes on average in the route cache, which provides more opportunities for route decay.

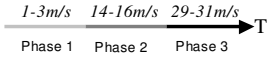


Figure 3: Illustration of a phase mobility scenario. Each phase lasts 600 seconds.

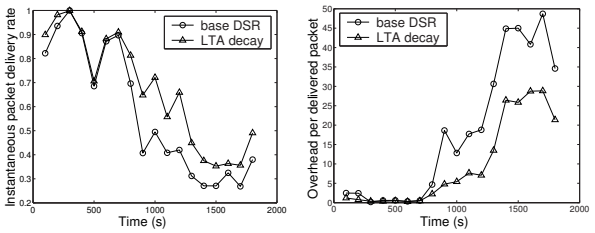
Similarly, for all mobility scenarios, both GA and LTA prefetch without decay always have a higher packet delivery rate than DSR, with rwp-pt0-ms20 having the highest improvement. Comparing GA and LTA with only prefetch vs. only decay, route decay alone always outperforms route prefetch alone (except for LTA in rwp-pt0-ms20 where both are on par). This is because inaccurate route prefetches incur unnecessary broadcasts, affecting the route caches not just at the local node, but at multiple nodes. These results indicate that care must be taken when using route prefetch.

For all RWP scenarios, combining prefetch and decay always results in a higher packet delivery rate than using any of them alone. This demonstrates that prefetch and decay, used cooperatively, can lead to additional performance gains. However, for RPGM, the results are not as clear. Simple averages do not fully capture the node clustering in RPGM; a better possibility might be to use a cluster average in addition to the global and local averages.

A Synthetic Phase Mobility Scenario. Since all the preceding evaluations are based on RWP and RPGM, two mobility models that demonstrate regular route lifetime characteristics and no sharp mobility changes, we create a synthetic mobility scenario with distinct phase behavior as shown in Figure 3. This mobility scenario is closer to real-world situations [7]. We evaluate the efficacy of our proposed technique that allows a protocol to detect and adapt to such dynamics.

We evaluate the performance of LTA decay for the phase mobility scenario as a case study here. Figure 4 shows that LTA decay outperforms DSR and adapts to the distinct mobility changes much better. Figure 4(a) shows the instantaneous packet delivery rate over time for both DSR and LTA decay. Figure 4(b) shows the average control overhead defined as the number of control packets sent out network-wide for each data packet successfully delivered.

In Phase 1, their performance are similar because the network is very static and there are thus few opportunities for route decay. In Phase 2, as nodes move faster, route lifetimes become shorter. LTA adapts to this situation by invalidating stale routes more proactively. This results in a higher packet delivery rate. LTA decay also lowers the control overhead by avoiding packets for fixing route errors in DSR. This in turn saves node energy and network bandwidth. Since LTA decay adapts to the mobility change at 600s, it does not experience a sharp performance degrada-



(a) (b)

Figure 4: Performance evaluation for the phase mobility.

tion as DSR does. In Phase 3, as nodes move even faster, route decays are even more frequent. The performance improvement, however, is not as high as for Phase 2. One possible reason is that the source-destination connectivity patterns may have changed. However, the mobility-adaptive behavior of LTA decay over DSR is clear.

IV. Discussions and Conclusions

Wireless ad hoc networks and sensor networks consisting of highly mobile nodes have gained increasing interest recently. Their high network mobility cause network connectivity to vary rapidly, presenting unique challenges to routing protocol design.

MARio must make a tradeoff between the predictor accuracy and gathering overhead and complexity. Normally the more accurate a predictor, the more overhead it incurs. So there is a balance to strike in selecting the right predictor. As demonstrated in this paper, routing performance using our proposed predictors based on EWMA improves for almost all scenarios we studied, though in different degrees. For rwp-pt0-ms20, using MARio even achieves a performance close to oracle. So we believe that even with simple predictors, MARio can achieve impressive performance. The overhead in using lifetime averages is mainly in the sampling process, and our sampling rate is pretty low; therefore, the overhead of using MARio should not be a concern. We plan to prototype a system based on mobile handhelds to investigate realistic overheads.

Thus far, we have only investigated MARio's effects on routing performance. We plan to further study how mobility knowledge affects resource usage in such systems, with a particular focus on energy saving. Furthermore, we plan to investigate MARio's application on Delay Tolerant Network routing [5] because similar knowledge about network partition could be exposed to and leveraged by the routing protocols. Finally, we plan to study MARio's performance on realistic mobility trace gathered from the initial ZebraNet deployment in central Kenya to evaluate its efficacy on a real-world scenario.

References

[1] Tracy Camp et al. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2002.

[2] Delay tolerant networking. `http://www.dtnrg.org/`.

[3] David Kiyoshi Goldenberg et al. Towards mobility as a network control primitive. In *ACM MobiHoc*, 2004.

[4] Ravi Jain et al. Towards an empirical model of user mobility and registration patterns (poster). In *ACM MobiHoc*, 2004.

[5] Sushant Jain et al. Routing in a Delay Tolerant Network. In *ACM SIGCOMM*, 2004.

[6] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, 1996.

[7] Philo Juang et al. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet. In *ACM ASPLOS*, 2002.

[8] Aman Kansal et al. Intelligent fluid infrastructure for embedded networks. In *ACM MobiSys*, 2004.

[9] Narayanan Sadagopan et al. PATHS: analysis of PATH duration Statistics and their impact on reactive MANET routing protocols. In *ACM MobiHoc*, 2003.

[10] Wenrui Zhao et al. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *ACM MobiHoc*, 2004.