A Compact Routing Architecture for Mobility

Cedric Westphal, James Kempf,
DoCoMo Labs USA
3240 Hillview Ave
Palo Alto, CA, USA
{cwestphal, kempf}@docomolabs-usa.com

ABSTRACT

We propose a compact routing architecture to support mobility in a scalable manner. Our routing architecture requires a route table of size of $O(\sqrt{n \log(n)})$ in order to provide a path stretch with a provable upper bound of 3. This is the optimal path stretch.

Our architecture is built upon the theory of compact routing, which has so far been utilized in static networks only. This is the first attempt to the best of our knowledge to transpose such architecture into a network with mobility support. Our contribution is to adapt the static theory of compact routing to a class of networks with mobile leaf nodes and a static core infrastructure.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.5 [**Computer-Communication Networks**]: Local and Wide Area Networks; G.1.m [**Mathematics of Computing**]: Numerical Analysis—*Miscellaneous*

General Terms

Compact Routing

Keywords

Routing Architecture, Compact Routing, Mobility Support, Scalability

1. INTRODUCTION

The issue of scalability in routing has been identified in the wired networks context [11, 10, 1, 15]. In order to contain the growth of the route table size, new architectures are required to support the explosion of route tables (please refer to [15] for a list of recent works dealing with this topic). The field of compact routing addresses the scalability of the routing table infrastructure by defining routing schemes with a small route table size, at the expense of a slightly suboptimal path instead of the shortest path.

Dealing with mobility inserts a layer of complexity in the routing architecture that impacts scalability. In particular the consistence of the naming with the topology might get distorted by the mobility of the nodes. Mobility thus imposes frequent updates of the routing information distributed in the network layer.

Updating the network topology due to mobility of the nodes requires disseminating and updating route information. A natural idea is to apply the concept of compact routing to a mobile environment, so as to keep the amount of information small in each node, and as a consequence, to limit the exchange of information throughout the network due to the mobility of the nodes.

We present necessary and sufficient conditions to support mobility using a compact routing algorithm located strictly inside a static infrastructure.

The paper is organized as follows. In Section 2 we present some related work. In Section 3, we give a brief description of compact routing. We describe our network model in Section 4 and introduce our compact routing architecture for mobility in Section 5, as well as the necessary and sufficient conditions introduced by the architecture. We then discuss the properties of the architecture in Section 6 and offer some concluding remarks in Section 7.

2. RELATED WORK

Mobile routing architectures are commonly based on expanding the wired network infrastructure to a mobile network. Most architectures attempt to extend IP routing to work in a mobile scenario. Two approaches to mobile routing have been taken in this context: extending routes to mobile nodes through overlays and modification of routing tables within a local domain to reflect the new identity to location mapping after a movement event. Other approaches, such as HIP [18] maintain routing tables at end hosts and therefore involve no change in the routing and addressing architecture, so these approaches will not be discussed further here.

The overlay approach is represented by many schemes, including some that have seen commercial deployment. Examples of such schemes include Hierarchical Mobile IP [4, 22], Mobile IP [19, 12, 20] and the Internet Indirection Infrastructure (I3) extension ROAM [25]. In these schemes, the identity function of the IP address is fixed by assigning the mobile node an IP address in a fixed subnet. Depending on the scheme, the subnet may be either within the local

MobiArch'08, August 22, 2008, Seattle, Washington, USA.

access network (Hierarchical Mobile IP) or at a potentially remote network somewhere in the Internet (Mobile IP), or potentially either (ROAM). A specialized router within the fixed subnet contains a routing table mapping the fixed identifier address to a variable locator address. The mobile node maintains the binding between the identifier address and locator address by signaling to the specialized router when a movement to a new subnet requiring a new locator address occurs.

Correspondent nodes communicate with the mobile node through the fixed identifier address which does not change as the mobile node moves. The specialized router takes care of forwarding packets through overlays constructed to the locator address. Newer schemes such as Proxy Mobile IP [9] transfer the burden of maintaining the binding between the fixed identifier address and changing locator address to the access router. This approach allows the mobile node to maintain a fixed and unchanging identity address while it moves, since the mapping between the identity address and locator address is handled by the network. The routing tables involved in all the various overlay schemes scale linearly as the number of mobile nodes handled by the specialized router because the routing table must include an identity to location mapping for each mobile node.

The routing table modification approach essentially requires a new intra-domain routing protocol that functions over a restricted topological area at the wireless edge of the access network, called a *micromobility routing domain*. Two examples of such schemes are Cellular IP [24] and HAWAII [21]. The micromobility routing domain is isolated from that part of the networking using standard IP IGP routing by a gateway router. All packets to and from a mobile node within the micromobility domain are routed through the gateway router, including packets between two mobile nodes within the micromobility domain.

When a mobile node moves between two access routers, the routing tables in the routers on the path between the new access router and the crossover router are modified to include host routes reflecting the new forwarding path from the gateway router. The crossover router is the first router on the path back from the new access router to the gateway router where the path to the old access router intersects the path to the new. Since packets in transit between the crossover router and the old access router are dropped if no additional measures are taken, the micromobility protocols include various measures to ensure seamless handover; that is, in-order forwarding of in-transit packets, to maintain good TCP performance and avoid noticeable glitches in real time traffic. The routing tables in these schemes also scale linearly in the number of mobile nodes covered by the router; in particular, the size of the routing table in the gateway router reflects all mobile nodes within the micromobility routing domain which limits the size of the micromobility domain.

Some efforts have been made to provide a naming architecture independent of the topology. VRR [2] presents a naming structure which is independent of the node location. This scheme is expanded in [3] into a naming and routing architecture based on flat identifiers. A flat identifier is a name drawn from an address space with no location semantics. The naming of the nodes does not vary with the node mobility, but support for the mobility of the node should be embedded into the routing substrate, by way of a distributed hash table maintained at each node for routing. Krioukov [14] reviews results showing that routing tables in topology independent (what is called “name independent” by the compact routing community) schemes scale polynomially in general, with optimal stretch 3 routing. These results apply to VRR as well.

3. COMPACT ROUTING

Compact routing [16, 14] studies the trade-off between the route table size at any given node, and the *path stretch*. If all nodes have route information for any other node in the network, then routing can be performed on the shortest path. However, if nodes only have partial route information, in order to reduce the route table size explosion, then packets will not follow the shortest path, but a longer path. The ratio of the longer path followed by the packets over the shortest possible path is the path stretch. The path length is computed along some graph metrics based on link costs to define a distance measure.

To achieve a path stretch of 1, [8] shows that, in a network with N nodes and degree d, the route table size must be of size $\Omega(N\log(d))$ bits at $\Theta(N)$ nodes. Since it is easy to construct a routing scheme which achieves stretch 1 routing with $N\log(d)$ bits at each node, the bound is tight. This means that to reduce significantly the route table size, then one must give up some slack in the path stretch.

At the opposite of the route table size/path stretch trade-off, we find [3] for instance, which provides a route table size which scales as $\Theta(\sqrt{N})$, but with no guarantees on the path stretch. The architecture presented in [3] is built upon the VRR protocol introduced in [2], which can have a path stretch of $\Theta(N)$ in the worst case.

In a generic static network, the theory of compact routing provides algorithms to achieve a path stretch of 3. Before we describe such algorithms, it is important to note that a path stretch of 3 is the best value one can hope to achieve with a route table size scaling as $o(N)$. Indeed, Gavoille and Gengler [7] showed that for any path stretch strictly less than 3 requires almost the same amount of memory as shortest path routing, namely $\Omega(N)$ bits at some nodes.

While a path stretch of 3 seems like a high bound, one must keep in mind that it is the *worst case* stretch for a generic network topology, and that in practice, the *average* path stretch is much closer to 1. For instance, [14] computes the average stretch of a stretch-3 algorithms on an Internet topology, and finds an average path stretch of 1.1, with 70% of the paths actually finding the shortest path. This is the reason for us to focus on stretch-3 schemes.

In [6], Cowen presents a routing scheme with a route table size of $O(N^{2/3}\log^{4/3}N)$ which achieves a worst case path stretch of 3. In [23], Thorup and Zwick are able to achieve the same path stretch with a storage size of $O(\sqrt{N\log(N)})$.

Both schemes in [6] and [23] function roughly along the same principles. Given a graph $G=(V,E)$ with positive weight associated to the edges, and a distance δ so that $\forall u,v\in V, \delta(u,v)$ is the distance between u and v on the graph, the idea is to define a set $A\subseteq V$ of vertices, denoted *centers* [23] or *landmarks* [6]. One can define the distance from a node v to a set B as:

$$\delta(B,v)=\min_{u\in B}\delta(u,v)$$

For a node v, $A(v)$ denotes the closest landmark, namely

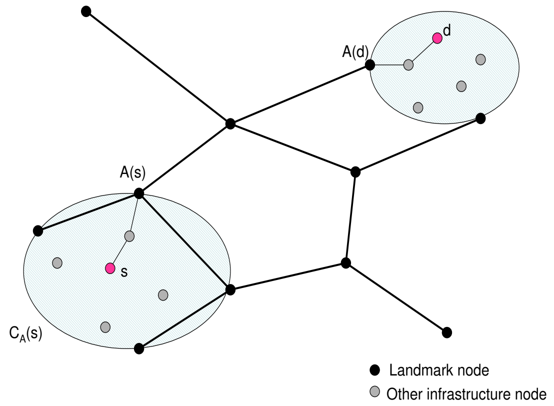

Figure 1: s routes to d by sending the packet to $A(s)$. The identifier of d contains $A(d)$, and $A(s)$ knows the shortest path to all other nodes in A, including $A(d)$. d is in $C_A(A(d)$, thus $A(d)$ knows the path to d. The path stretch of 3 comes from the 3 phases in the route, $s - A(s)$, $A(s) - A(d)$ and $A(d) - d$.

$u \in A, \delta(u,v) = \delta(A,v)$, with ties broken by ordering in the address name space. For each $w \in V$, one can define the *cluster* of w as the set of vertices closer to w than to any landmark:

$$C_A(w) = \{v \in V | \delta(w,v) < \delta(A,v)\}. \quad (1)$$

One can view the set $C_A(w)$ as some type of Voronoi tessellation. The construction of the set A is different for [6] and [23]. The routing and addressing however is similar. Every node v keeps track of the shortest path to:

- every node in $C_A(v)$;
- every node in A.

.

This entails a hierarchical structure, where nodes in close proximity have a direct path maintained in the route table, whereas nodes further away are routed by way of the nearest landmark. The bounds on the route table size come from computing clusters and landmarks sets with size satisfying the right properties.

Routing works as follows: the identifier is a triplet of the node address v, its closest landmark address $A(v)$, and the port at $A(v)$ routing towards v, denoted $p_{A(v)}(v)$. Any node u which receives a packet with the triplet $(v, A(v), p_{A(v)}(v))$ can then route according to the following algorithm:

- If $u = A(v)$, then route along $p_u(v) = p_{A(v)}(v)$;
- Else, if $u \in C_A(v)$, route along $p_u(v)$;
- Else route along $(A(v), p_u(A(v)))$.

The algorithm is depicted on Figure 1, where a packet is routed between a source s to a destination d using the clusters $C_A(s)$, $C_A(A(d))$ and the landmark set A.

Each node in A stores a route to all other nodes in A, while all nodes v not in A store a path to all nodes in $C_A(v)$. Cowen [6] describes a scheme to construct a set A of size $\tilde{O}(N^{2/3})$ such that $|C_A(w)| = \tilde{O}(n^{2/3})$, while Thorup and Zwick [23] achieve $|A| = \tilde{O}(N^{1/2})$ and $|C_A(w)| = \tilde{O}(N^{1/2})$, which is optimal up to a polylogarithmic factor. The notation $\tilde{O}(f(N))$ means $O(f(N)\log^{\beta}(N))$ for some β, that is: $O(f)$ up to a polylogarithmic multiplier. We denote the Thorup-Zwick scheme as the TZ construction.

The schemes described so far have been considered in static contexts. [17] has defined a routing protocol for sensor networks called S4, for Small State, Small Stretch routing protocol. [5] have defined a different routing protocol for use in ad hoc networks. The goal is to use compact routing to minimize the amount of information during proactive route maintenance in ad hoc networks. There has been some efforts to consider dynamic networks (see for instance [13]), but the dynamic parameter is not the topology, which is static, but the cost associated with an edge in the graph (V,E).

We study now an architecture that uses compact routing to support mobility in wide area wireless networks.

4. NETWORK MODEL

We consider the issue of compact routing in a network where mobile nodes are attached to a static infrastructure (cellular networks, or wide-area mesh networks). The distance δ we consider is the hop count. We consider a fixed infrastructure with mobile nodes as leaf nodes.

The network is composed of n static nodes, and m mobile nodes, so that the total number of nodes $N = n + m$. The relationship between n and m will be discussed later on. Each mobile nodes attaches to one, and only one, of the static nodes. In essence, we assume that each mobile node only has one WAN wireless interface.

The static infrastructure is represented by a graph $G = (V,E)$, with $|V| = n$. The whole graph is a dynamic graph $G' = (V',E')$ where $|V'| = N = n + m$ and $|E'| = |E| + m$, where each one of the m mobile node is attached to one node $v \in V$ by an edge in $E' \setminus E$. Technically, V' depends on the time instance t, but for simplicity of notation, we keep the time dependency implicit.

We do not allow direct communication between two mobile nodes. That is, the minimum hop count between two mobile nodes, even if they are within range of each other, is two. We impose this requirement since it is true in practice, either because the mobile device does not have the capability to communicate directly with another device on its WAN interface (cellular network case) or because the network policy requires the communication to go over the infrastructure (as is the case in most wireless mesh networks).

Mobile nodes have no routing information, and only forward packets to the infrastructure node at the edge of the network they are connected to. There is no routing intelligence in the mobile device, only in the network. Again, this assumption is required in many practical situations. We do not consider for now multi-homed terminals.

We make no assumption regarding the topology of the network in this work. We consider the naming and routing for the mobile nodes, as well as the impact of mobility on the protocols within the fixed network infrastructure.

We consider a routing algorithm which is adapted from the TZ scheme as follows. A set A of landmarks is constructed. We will see below how this set is constructed to conform to a mobile scenario. The set A is a subset of the static infrastructure: $A \subset V$.

The routing is performed as follows: each mobile node u, connected to the infrastructure through the node v is identified as a quadruplet composed of u and the triplet of identifiers of v in the graph (V, E). This means the identifier u is added to the routing header. A packet is thus routed first to u's access point using the TZ scheme on (V, E), and is forwarded by the access point to u over the last hop.

5. COMPACT ROUTING ARCHITECTURE FOR MOBILITY

While the compact routing architecture described above has been defined only for use in static networks, the attractive features of the architecture merit to be extended to the mobile situation. In particular, a key issue of supporting mobility is to update the routing tables after a mobility event has occurred. One would hope that the compactness of the routing facilitates updating the route information in a dynamic network. This is what we set to investigate in the reminder of this document.

Mobility introduces several issues with regards to the compact routing architecture defined above. The issues pertain to:

- Dimensioning the network. Due to the dumb terminal assumption, the set A has to be included in the static infrastructure, and such a set can be constructed only if some relationship is satisfied between the number of mobile terminals and the number of static nodes.
- Creating the set of landmarks. Landmarks are chosen so that the cardinality of A and $C_A(w)$ satisfies some nice properties to ensure the bounds on the route table size. However, if A were set once and for all, mobility alters the cardinality of $C_A(w)$ for some nodes w.
- Adapting the routing procedures to the case of mobile nodes.
- Updating the routing path after a mobility event has occurred.

We will detail each one of these issues in the following sections.

5.1 Necessary Condition on the Network Composition

We investigate the relationship between n and m in order to successfully implement compact routing in the mobile network architecture described in Section 4.

We assume that the number m of mobile node is related to n according to:

$$\exists \alpha > 0, m = n^{\alpha}. \tag{2}$$

Theorem 5.1. *For the TZ compact routing algorithm in Section 3 to be applicable in a mobile environment, a necessary condition is that $\alpha < 2$.*

Proof. We first prove that $\alpha \geq 2$ leads to a contradiction. Assume $\alpha \geq 2$, then the set A constructed according to the TZ algorithm, has cardinality:

$$|A| = \sqrt{N \log(N)} = (n + n^{\alpha})^{1/2} \log^{1/2}(n + n^{\alpha}) \tag{3}$$

As $\alpha \geq 2$, the term n^{α} is dominant over n, and Equation (3) becomes:

$$|A| = O(n^{\alpha/2} \log^{1/2}(n)) \tag{4}$$

Since per assumption, the mobile nodes have no routing information, the set A is a subset of the static infrastructure. We then have that $|A| \leq n$, and combining this with Equation (4) yields a contradiction if $\alpha \geq 2$. $\square$

There are two scenarios to consider which satisfy the necessary condition:

- Scenario (i): In the first scenario, and in many common practical applications, *individual* nodes attach to the infrastructure. The number of mobile nodes attaching to each edge node of the static infrastructure will be independent of n, and each edge node can support only *gamma* mobile nodes, so that $m \leq \gamma n$ for $\gamma > 0$. This means that for practical purposes, the necessary condition of Theorem 5.1 is satisfied. We take $m \leq \gamma n$ in the remainder of this document.
- Scenario (n): In the second scenario, it is whole *networks* which attach to the mobile node, as opposed to single devices. In this case, the number of nodes might be greater than γ per edge node, but the necessary condition imposes that the number be $o(n)$, so that $m < n^2$.

In this initial step to bring mobility to compact routing, we focus exclusively on the first scenario. This is also consistent with our assumption that mobile nodes do not communicate directly with each other.

5.2 Construction & Update of the Landmark Set

The TZ algorithm specifies a landmark set A once and for all, since it focuses on a static network. However, in a mobile environment, the set A must be updated as the distribution of the nodes evolves within the network. In particular, the cardinality of the sets A and $C_A(w), \forall w \in V$ must always satisfies the proper bounds to ensure the scalability of the route table at each node.

[23] constructs the set A using the following algorithm: the set A is initially set to empty. A set W is initialized as equal to V. Set a value $s = \sqrt{n/\log(n)}$. Then, while $W \neq \emptyset$, the following procedure is repeated:

- A is replaced by $A \bigcup W_s$. W_s is a random subset of W where each element of W is included independently in W_s with probability $s/|W|$. If $s/|W| > 1$, then $W_s = W$ is added to A.
- $C_A(w)$ is computed for each $w \in V$ as per Equation (1).
- W is replaced by the set $\{w \in V | \, |C_A(w)| > 4n/s\}$

When W is equal to $\emptyset$, then the landmark set A is returned. For the value of s chosen above, this ensures the proper cardinality of A and $C_A(w)$ for all $w \in V$.

In our mobile architecture, we now consider the added constraint that the set A is included in the static infrastructure.

Theorem 5.2. *For scenario (i), the TZ cluster construction performed on (V, E) at the exclusion of the mobile nodes, applies to the mobile network as well. In particular, $|A|$ and $|C_A(w)|$ are $O(\sqrt{n \log(n)})$ for all $w \in V$.*

Proof. For scenario (i), if we apply the TZ cluster construction on (V, E), we end up with a set A with the required cardinality, and cluster $C_A(w)$'s such that $|C_A(w)| = O(\sqrt{n \log(n)})$. However, because we constructed A and C_A using exclusively (V, E), $|C_A(w)|$ does not account for any mobile nodes. In the scenario (i), for each nodes v in V, there are at most γ nodes attached to v. We define by $C'_A(w)$ the set:

$$\begin{aligned} C'_A(w) &= \{v \in V | \delta(w,v) < \delta(A,v)\} \bigcup \\ & \quad \{v \in V' \backslash V | \delta(v, C_A(w)) = 1\}. \end{aligned} \tag{5}$$

For $w \in V$, $C'_A(w)$ is the set of nodes v in V closer to w that to A plus all the mobile nodes attaching to such v. Thus $C'_A(w) \leq \gamma C_A(w)$. Thus $|C'_A(w)| = O(\sqrt{n \log(n)})$ and the sets A and $C'_A(w)$ satisfy the required properties for scalable routing. $\square$

One important point is that, despite the mobility, under the assumptions of scenario (i), the cluster construction is independent of the node mobility, and does not need to be updated as nodes move about. This means that the routing infrastructure stays simple to manage, since the landmark set and the cluster sets do not have to be recomputed at every mobility event.

5.3 Routing Algorithm

After constructing the set A of landmarks, and the clusters $C_A(w)$ associated with each node $w \in V$, we can now describe the routing in a mobile compact routing architecture.

Routing works as follows. For a mobile node $v \in V' \backslash V$, we define by $e(v)$ to be the infrastructure edge node it attaches to. For $v \in V$, then $e(v) = v$. The identifier is then a quadruplet $(v, e(v), A(e(v)), p_{A(e(v))}(e(v)))$. The naming can be simplified if one is allowed to re-name the nodes. For instance, for a node in V, the identifier is redundant as $v = e(v)$, and the redundancy can easily be removed. However, this only affects the routing overhead for each packet, but not the scalability or the stretch of the routing algorithm.

Any node u which receives a packet with the quadruplet $(v, e(v), A(e(v)), p_{A(e(v))}(e(v)))$, then the routing works as follows.

- If u is a mobile node, $u \in V' \backslash V$ forwards packets with destination other than u to $e(u)$.
- Otherwise, if $u = e(v)$, then u hands off the packet to v.
- Otherwise, if $u = A(e(v))$, then route along $p_{A(e(v))}(e(v))$;
- Else, if $u \in C_A(e(v))$, route along $p_u(e(v))$;
- Else route along $(A(e(v)), p_u(A(e(v))))$.

5.4 Route Updates

The identifier depends on the v and $e(v)$. When node v moves from one access point node to the next, then its identifier must be updated. In order to preserve the scalability, the use of an indirection architecture, as described in Section 2, would be required.

In order to reach a node $v \in V' \backslash V$, one needs to know the quadruplet of node names which is used as an identifier. v thus should update a location repository with this identifier. At *minima*, the mapping between v and $e(v)$ should be provided to a fixed node inside the static infrastructure, since it is the least amount of information required to reconstruct the identifier from v.

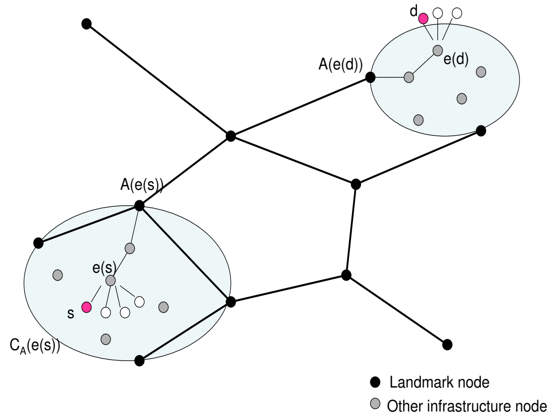

Figure 2: s routes to d by sending the packet to $e(s)$. If $e(d)$ is not in $C_A(e(s))$, the identifier of d contains $A(e(d))$, and $e(s)$ routes to $A(e(s))$. $A(e(s))$ knows the shortest path to all other nodes in A, including $A(e(d))$. $e(d)$ is in $C_A(A(e(d))$, thus $A(e(d))$ can route to $e(d)$.

Figure 2 represents the compact routing mobility architecture.

6. OPTIMALITY OF THE MOBILITY ARCHITECTURE

We now show that the scheme we constructed in the previous section sill preserves the optimality of the compact routing architecture it is based upon.

Theorem 6.1. *The compact routing algorithm for mobile architecture described above satisfies a path stretch less than 3 in the worst case scenario, while the route table size is* $O(\sqrt{n \log(n)})$.

Proof. Since by Theorem 5.2, we have that $|A|$ and $|C_A(w)|$ satisfy $O(\sqrt{n \log(n)})$, and since the identifiers draw from only these sets, then the route table size also satisfies this property.

We need to check that the stretch is less than 3. There are three scenarios to consider for the stretch of the path u, v where $u, v \in V'$. If u and v are both in V, then the TZ algorithm is identical as its static counterpart, and thus the stretch 3 is satisfied. If u and v is in $V' \backslash V$, then the path follows three legs: from u to $e(u)$, from $e(u)$ to $e(v)$, and from $e(v)$ to v.

Per our assumption on the connectivity of u, any packets leaving u will go to $e(u)$, and thus $u, e(u)$ is the shortest path between these two points. Similarly for $e(v), v$. Further, $u, e(u)$ must be included in any path from u to any other node, and thus must belong to the shortest path.

Thus, the shortest path between u and v can be divided into three legs as well: $u, e(u)$, then the shortest path between $e(u)$ and $e(v)$ and then $e(v), v$. In terms of hop count, the distance of $e(u), e(v)$ is d_{tz} in our architecture, and d_{sp} for the shortest path. The path stretch is thus, for hop count:

$$\text{path stretch } = \frac{1 + d_{tz} + 1}{1 + d_{sp} + 1} = \frac{2 + d_{tz}}{2 + d_{sp}} < 3 \qquad (6)$$

The last inequality comes from the fact that since $e(u)$ and $e(v)$ are in V, then $d_{tz} \leq 3d_{sp}$ per the TZ construction. $\square$

This is a constructive proof that, under scenario (i), the condition $m \leq \gamma n$ is *sufficient* to apply the TZ scheme in a mobile environment. Note that there is a gap of a factor n between the necessary condition of Theorem 5.1 and the sufficient condition constructed above.

7. CONCLUSIONS

In this paper we studied the implications of using compact routing models, originally designed for a static infrastructure, into a mobile system. We saw that under some reasonable assumptions, the compact routing architecture could be adapted to a system with mobile, while preserving the scalability of the routing and the path stretch bounds.

The scalability is preserved as the size of the landmark set A and the routing clusters $C_A(w)$ scale as $O(\sqrt{n \log(n)})$ in the mobile case as in the static scenario. $C_A(w)$ is larger by a constant factor, but scales according to the same polylogarithmic polynomial.

The path stretch satisfies the same optimal stretch 3. As we have discussed in the motivation, stretch 3 is optimal, as any stretch lower than 3 would require a route table scaling as $O(n)$ instead of $O(\sqrt{n})$ (up to a log factor).

Future work includes the implementation of the scheme and its numerical evaluation. Further, future work also includes the specification of the integration of the mobile compact routing architecture with an indirection scheme to locate the mobile nodes and enable all-the-time reachability. In particular, the scalability of the route look-up should be of the same order as that of the routing.

Also, the gap between the *necessary* condition $\alpha < 2$ and the *sufficient* condition $\alpha \geq 1$ should be studied and, if possible, bridged.

8. REFERENCES

[1] A. Broido, E. Nemeth, and kc Claffy. Internet expansion, refinement and churn. *European Transactions on Telecommunications*, 13(1):33–51, 2002.

[2] M. Caesar, M. Castro, E. Nightingale, G. O'Shea, and A. Rowstron. Virtual Ring Routing: Network routing inspired by DHTs. In *Proc. of ACM SIGCOMM'06*, pages 351–362, 2006.

[3] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, and S. Shenker. ROFL: routing on flat labels. In *Proceedings of ACM SIGCOMM*, 2006.

[4] C. Castelluccia. A hierarchical mobile IPv6 proposal. INRIA Technical Report RT-0226.

[5] M. Chen and A. Ganesh. A compact routing protocol for ad hoc networks. In *Proc. of SPASWIN*, 2006.

[6] L. Cowen. Compact routing with minimum stretch. In *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 1999.

[7] C. Gavoille and M. Gengler. Space-efficiency of routing schemes of stretch factor three. *Journal of Parallel and Distributed Computing*, 61(5):679–687, 2001.

[8] C. Gavoille and S. Pérennès. Memory requirement for routing in distributed networks. In *Proc. of the 15th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 125–133, May 1996.

[9] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil. Proxy mobile IPv6. IETF draft, work in progress.

[10] G. Huston. Commentary on inter-domain routing in the Internet. IETF RFC 3221, 2001.

[11] G. Huston. Scaling inter-domain routing -a view forward. *The Internet Protocol Journal*, 4(4), 2001.

[12] D. Johnson, C. Perkins, and J. Arkko. Mobility support in ipv6. IETF RFC 3775, June 2004.

[13] A. Korman and D. Peleg. Dynamic routing schemes for graphs with low local density. *ACM Transactions on Algorithms, to appear.*, 2007.

[14] D. Krioukov, K. Fall, and X. Yang. Compact routing on Internet-like graphs. In *Proceedings of IEEE INFOCOM'04*, March 2004.

[15] D. Krioukov and kc Claffy. Toward compact interdomain routing. arXiv:cs/058021v1, August 2005.

[16] D. Krioukov, kc claffy, K. Fall, and A. Brady. On compact routing for the Internet. *SIGCOMM Comput. Commun. Rev.*, 37(3):41–52, 2007.

[17] Y. Mao, F. Wang, L. Qiu, S. Lam, and J. Smith. S4: Small state and small stretch routing protocol for large wireless sensor networks.

[18] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson. Host identity protocol. IETF draft, draft-ietf-hip-base-10, work in progress.

[19] C. Perkins. IP mobility support. IETF RFC 2002, October 1996.

[20] C. Perkins. Mobile IP. *IEEE Communications Magazine*, 35(5):84–99, May 1997.

[21] R. Ramjee, T. F. L. Porta, S. Thuel, K. Varadhan, and S. Y. Wang. HAWAII: A domain-based approach for supporting mobility in wide-area wireless networks. In *ICNP*, pages 283–292, 1999.

[22] H. Soliman, C. Castelluccia, K. El Malki, and L. Bellier. Hierarchical mobile ipv6 mobility management (hmipv6). IETF RFC 4140, August 2005.

[23] M. Thorup and U. Zwick. Compact routing schemes. In *ACM Symposium on Parallel Algorithms and Architectures*, pages 1–10, 2001.

[24] A. Valkò, J. Gomez, and A. Campbell. An overview of cellular IP. In *IEEE Wireless Communications and Networking Conference*, September 1999.

[25] S. Zhuang, K. Lai, I. Stoica, R. Katz, and S. Shenker. Host mobility using an internet indirection infrastructure. In *Proceedings First International Conference on Mobile Systems, Applications, and Services*, 2003.