Towards a Landmark-based Flat Routing

Rafael Pasquini[1], **Fábio L. Verdi**[1] **and Maurício F. Magalhães**[1]

[1]Department of Computer Engineering and Industrial Automation (DCA)
School of Electrical and Computer Engineering (FEEC)
State University of Campinas (Unicamp)
C. P. 6.101 – 13.083-970 – Campinas – SP – Brazil

{pasquini,verdi,mauricio}@dca.fee.unicamp.br

Abstract. *Two main groups of Flat Routing proposals are found in the literature. The biggest group considers the existence of an underlay network providing direct communication between neighbors at the flat identity layer. On the other hand, a smaller set of proposals consider a scenario that has none underlay network, i.e., routing directly on flat identifiers/names. Our interest is concentrated on the second group due to the perspective of a new internetworking model in which the network layer has no information regarding location and, in this paper, our Landmark-based Flat Routing proposal is introduced. We also present a tool for evaluating different topologies and flat routing protocols. The quantitative results show the signaling overhead and the trade-off between routing table size and route stretch. The results were collected using the tool under two distinct topologies, a regular mesh and an Internet-like topology.*

1. Introduction

The investigative scenario of this paper is related to routing directly on flat identifiers/names. This approach considers that the network layer does not contain any location information to drive the packet forwarding. Examples of proposals in this scenario include Identity Based Routing (IBR) [Caesar 2007], Virtual Ring Routing (VRR) [Caesar et al. 2006a] and Routing on Flat Labels (ROFL) [Caesar et al. 2006b]. Some of the benefits introduced by this approach include: 1) native support for mobility and multihoming; 2) simpler allocation of identifiers by requiring only uniqueness (the IP requires uniqueness and topological adherence); 3) no need for new mapping services since data is forwarded based on the flat identity (no need for identifier to locator mapping) and 4) better support for network access controls which can be applied on the identifier.

In this line, our Landmark-based Flat Routing (LFR) proposal is introduced. The landmark term comes from Compact Routing[1] [Krioukov et al. 2007] work and its definition found in [Tsuchiya 1988] is: "A Landmark is a router whose neighbor routers within a certain number of hops contain routing entries for that router".

In order to investigate the features and the behavior of our proposal, we developed a tool that is capable of emulating flat routing without the usage of any address-based substrate. The ideas behind this tool are: 1) development of flat routing mechanisms for verifying its behaviors under different scenarios and 2) simplification of the analysis for

[1]Refer to the Dmitri work [Krioukov et al. 2007] as a starting point for the vast amount of Compact Routing information available in the literature.

the protocol characteristics. Some of the characteristics that can be analyzed using the tool include route stretch, routing table size, signaling overhead and protocol behavior under dynamic scenarios.

Two different topologies with 256 nodes for testing were defined. The first encompasses a regular (ad hoc like) mesh topology and the second is a power-law (Internet-like) topology. We implemented our LFR proposal as introduced in this paper and another modified version in which landmarks are not used. Both implementations were evaluated in both topologies, resulting in four scenarios whose main objective was to demonstrate the benefits that landmarks bring to the overall system. In order to demonstrate the influence of the number of landmarks, two other scenarios with different number of landmarks were also evaluated using the regular network topology.

The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 details our Landmark-based Flat Routing proposal. Section 4 explains the tool that is being developed in order to support the analysis and comparisons of flat routing algorithms. Section 5 presents the quantitative results for both implementations evaluated on both topologies and also presents the results for the scenarios with different number of landmarks. Section 6 brings the next steps and concludes the paper.

2. Related Work

One pioneer related work in line with LFR is the IBR [Caesar 2007] proposed by Matthew Caesar and organized in three main phases: 1) Routing on an abstract graph; 2) Flat (ad hoc) wireless network routing and 3) Flat Internet scale routing.

The IBR protocol developed in the first phase organizes nodes into a logical ring and a node's position is determined by its identifier. Routing packets is only a question of making progress along the ring through the usage of pointers to adjacent nodes. The second phase, in which the IBR was extended to work in an ad hoc wireless scenario, is denominated VRR. Such a scenario allows the investigation of the protocol under certain failure modes and resource constraints. In the third phase, the IBR was extended to work in the Internet scenario. Named ROFL, it uses a single identity space for naming hosts and introduces the concept of stable/ephemeral hosts tracked by their hosting routers. These hosting routers act on behalf of nodes while discovering successors, predecessors and transmitting data.

The successor and predecessor requirements in this virtual ring scenario is a strong condition and, originally, none proximity relation is contemplated. In the VRR and ROFL scenarios, Matthew introduced a cache mechanism aimed at reducing stretch and contemplating proximity. Under this scenario, if a node does not have a successor or predecessor, the ring is broken and needs to be merged.

Another important related work is Unmanaged Internet Protocol (UIP) [Ford 2003] proposed by Bryan Ford. In his Ph.D. thesis [Ford 2008], he proposes an architecture to contemplate social networks over the current Internet called Unmanaged Internet Architecture (UIA). He developed a UIA prototype and one simple routing mechanism was included in it. The thesis also includes two different approaches (more sophisticated) for flat routing. One proposal is the UIP and the other one is based on name-dependent Compact Routing ideas.

The UIP algorithm employs routing tables divided into columns (called buckets) and, based on XOR operations, it inserts neighbors in these buckets taking into account the number of bits returned from these XOR operations. It has a connectivity invariant that requires at least one neighbor in each bucket to assure traffic delivery. It is also possible to use a factor k to multiply the number of neighbors present in each bucket aimed at generating routes with smaller stretch. Regarding the layer-3 connectivity, the UIP adopts an hybrid approach. In this scenario, as always as possible, the UIP uses the connectivity provided by any underlay network. Conversely, in case of connectivity gaps left by address-based protocols such as IP (for example, in case of NAT and/or firewalls discontinuities) the UIP circumvents this gaps at the identity layer.

We consider the connectivity invariant requirement of UIP a strong requirement because, under this scenario, neighbors are searched in the complete network. Another restriction is related to the usage of the identity space. If it is completely allocated, it is certain that all nodes will fulfill their buckets. However, if the identity space is not completely allocated, nodes can search for non-existent neighbors on the whole network.

For example, in an identity space of 4 bits, node 0 (0000 in binary) can have neighbors from 8 to 15 (1xxx) in its first bucket; nodes from 4 to 7 (01xx) in the second bucket; nodes 2 and 3 (001x) in the third bucket and only node 1 (0001) in its last bucket. The same occurs for all nodes, only one neighbor fits in the last bucket. Considering a homogeneous identifier distribution, it is probably easier to find a neighbor that fits in the first buckets than in the last ones. In the UIP proposal, if node one does not exist in the network, node zero may search the complete network to satisfy the connectivity invariant of UIP, and worse, it will never satisfy the invariant since node one does not exist.

However, the UIP infrastructure seems to be more interesting for mesh networks instead of the virtual ring scenario of VRR and ROFL. The introduction of landmarks as proposed by LFR relaxes the UIP connectivity invariant and, also, simplifies the identity space usage. For this scenario, only neighbors inside the same region are inserted in the routing tables, allowing the existence of empty buckets and contemplating proximity.

3. Landmark-based Flat Routing Proposal

The Flat Routing term is related to the flat identity space usage that requires a flat routing mechanism and the Landmark term comes from Compact Routing [Krioukov et al. 2007] work. In this section, the neighborhood discovery process to construct the routing tables and the routing mechanism to provide communication between nodes are presented.

In the LFR proposal, a wider definition is given to the term router. It can represent a node in an ad hoc network or an autonomous system (AS) in the inter-domain routing system. From now on, the terms router, domain and node are interchangeably used in this paper to represent the routing entity in the LFR proposal. Regarding the landmark definition found in [Tsuchiya 1988], we extended it to create a Landmark-based context:

"A Landmark is a node in the network whose neighbor nodes within a certain number of hops not only contain routing entries for that node but also are registered on it to constitute a landmark region".

Based on the definition above, in this work the landmark has two main functions: 1) to delimite regions in the network graph and 2) to work as a "default router" inside a region. Both functions has relation to the LFR routing tables. As mentioned before, the LFR routing tables (derived from UIP) have buckets according to the identity space number of bits and neighbors are positioned respecting XOR operations. However, the main difference between UIP and LFR is that LFR allows the possibility of having empty buckets in the routing tables. These empty buckets are a consequence of the first landmark function that limits the scope for neighborhood discovery (Sec. 3.1) to a region. The second goal of the landmark is to be used as a "default router" to which the packets are forwarded when a node is reached whose required bucket is empty.

In Fig. 1 it is possible to verify an example of a LFR routing table for node zero in an identity space of 4 bits. The figure shows the XOR operations necessary to introduce nodes 1, 3, 5, 10 and 13 in the routing table of node zero. According to the LFR scenario, if node one was not present in the landmark region of node zero, the bucket 3 of node zero would be left empty after the discovery process.

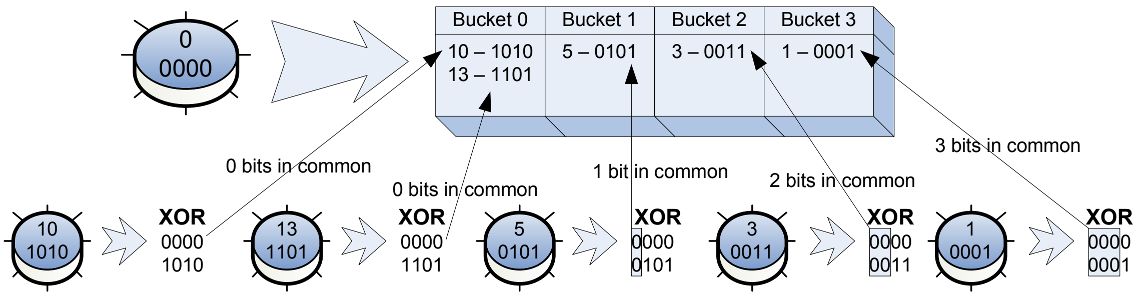

Figure 1. Routing table for node 0 - Buckets and XOR operations example.

3.1. Neighborhood Discovery Process

In the neighborhood discovery process, nodes are aimed at constructing their routing tables. The search for neighbors is based on the empty buckets of each individual node. The whole process involves some steps that are depicted in Fig. 2. To simplify the explanation, the scenario presented in Fig. 2 only describes the process from the perspective of node zero.

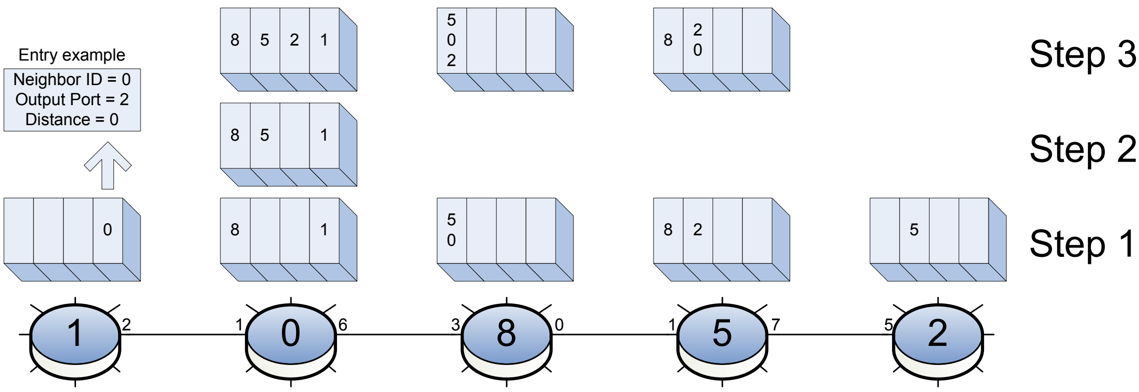

Figure 2. Learning process during the neighborhood discovery phase.

The first step is to bootstrap nodes in the network. As can be seen in step 1 of Fig. 2, all nodes learn their adjacent neighbors that are attached to their physical interfaces. After this initial step, the nodes that still have empty buckets continue to search neighbors inside their region. In order to find neighbors in this second phase, a query containing information about the empty buckets is generated and sent to the neighbors already present in the routing table. To limit the discovery to a specific landmark region, nodes include their landmark ID in the discovery messages. In this way, nodes with different landmark ID located at the edges are able to honor their region boundaries.

In step 2 of Fig. 2, node zero generates a query and asks for nodes in the intervals between 2 - 3 (bucket 2) and 4 - 7 (bucket 1). As a result of this iteration, node zero discovers node five from its neighbor node eight.

As the LFR proposal does not consider any underlay network providing direct communication between nodes, it is only possible to provide communication at the flat identity layer. The UIP proposes two different approaches: the usage of source routing or performing recursive tunneling. Unfortunately, both scenarios are packet centric, i.e., the source route or tunnels information must be carried in the packets, resulting in variable header size. In the first scenario, it is also necessary to include the whole path information in the routing table entry to be able of reaching neighbors. In this case, the routing table deployment is difficult since the routing table entries also have variable size.

The LFR proposal adopts a network centric approach as opposed to the packet centric approach of UIP. Under this network centric scenario, the packet header needs only to carry one information regarding the source and destination nodes (fixed packet header size). It also simplifies the routing tables construction by fixing the size of entries. An example of the routing table entry is shown in Fig. 2 at the top left corner. Associated to each entry, there are the information about the interface towards the neighbor and the distance to it. During the discovery process, if nodes learn the same neighbor through different interfaces, all of this information can be stored for path diversity.

In order to construct this network centric approach, nodes in the path between two neighbors learn path information based on the exchanged messages during the neighborhood discovery. The learned information not only includes the node ID of the message generator, but also the IDs carried inside the messages.

In step 3 of Fig. 2, node zero asks for nodes in the interval 2 - 3 and learns about node two through its neighbor five. At the same time, nodes five and eight learn information to grant the connectivity between nodes zero and two. Node five learns about node zero (the query message generator) and node eight learns information about node two (information contained in the answer message). All this learned information spreads “path information” in the network and does not introduce an overhead problem related to the number of routing entries since it has only the scope of a region.

This mechanism preserves the XOR routing model based on buckets. Suppose that node zero would like to send a packet to node two. It performs an XOR operation, receives 2 bits in common and searches a neighbor on bucket 2. In this case, it finds the node two in its table towards its interface 6 (node eight). Once the packet reaches node eight, it performs the XOR operation and discovers 0 bits in common. As can be seen, this represents a retrogression in the binary space. In UIP, such retrogression is not allowed. The packet had to be directly delivered to node two using one of its approaches (source routing or tunneling). In this proposal, although the packet is delivered to node eight and the XOR operation returns 0 bits in common, node eight will search a neighbor in its bucket 0 and it is granted that the same node found by node zero (node two) will be present at this bucket because of the learning process. In such a case, node eight forwards the packet through its interface 0. The same occurs in node five.

Another interesting property of this mechanism is the possibility to shortcut the packets forwarding. Consider that node zero would like to send a packet to node four (node four is not present in the figure). It performs the XOR operation, receives 1 bit in common and finds node five as the next hop; then it forwards the packet towards its interface 6 (node eight). When the packet reaches node eight, it performs the XOR operation, receives 0 bits in common and searches a neighbor in its bucket 0. Suppose that at bucket 0 of node eight (1000) routing table, it not only has node five (0101) in its table but also has node four (0100). As the routing function returns the biggest common prefix inside the bucket, the packet that is being forwarded to node five (next hop towards node four) can be deviated to node four before reaching node five.

3.2. Routing Mechanism

After the discovery and learning processes be executed, the LFR is ready for routing. It can occur at common nodes level (Sec. 3.2.1) and at landmark nodes level (Sec. 3.2.2). In both scenarios, the packet header is the same as depicted in Fig. 3.

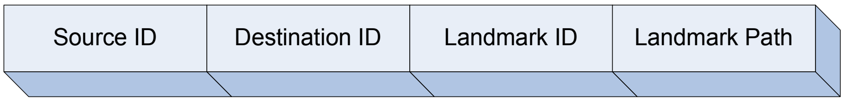

Figure 3. LFR packet header.

- **Source and Destination IDs:** these fields carry the IDs of the communicating entities;
- **Landmark ID:** this field is filled with the ID of the landmark through which the packet must be forwarded to. It occurs in the case that a packet reaches a node whose necessary bucket to forward the packet is empty;
- **Landmark Path:** this field is used when a packet reaches the first landmark and starts to be routed at the landmarks level. In order to avoid loops, this field carries information about the landmark IDs (regions) which the packet has already visited. To fix the header size, the Landmark Path field is defined as a Bloom filter [Broder and Mitzenmacher 2004].

3.2.1. Routing Mechanism in Common Nodes

The routing algorithm that is performed in common nodes is shown in Alg. 1. In a first moment, the packet forwarding decision is done using the XOR routing model through a query in the Forwarding Information Base (FIB) using the *findNextHop* method in which the XOR function is implemented to return the better next hop available in the FIB. Note that this approach contemplates the concept of network proximity by preserving the communication inside a region as much as possible. During this process, if the destination node is not inside the region, the packet will reach a node with a gap in the correspondent bucket and, in this case, this packet will be marked with the regional landmark ID and forwarded to it (similar to the IP default route forwarding).

When a packet is marked with a landmark ID, common nodes only are allowed to deviate this packet if the exactly destination is present in its routing table. The *findSpecificID* method only returns a next hop if the exact destination is found in the FIB. Otherwise, the packet is forwarded towards the landmark informed in the packet.

Algorithm 1 commonNodesPacketForwardingEngine($packet$)

if $packet$.getLandmarkID().isEmpty() = true **then**
  $nHop$ = fib.findNextHop($packet$.getDestinationID())
  if $nHop \neq NULL$ **then**
    forwardPacket($nHop$);
  else
    $packet$.setLandmarkID($myLandmarkID$);
    forwardPacket($myLandmarkID$);
  end if
else
  $nHop$ = fib.findSpecificID($packet$.getDestinationID())
  if $nHop \neq NULL$ **then**
    forwardPacket($nHop$);
  else
    forwardPacket($packet$.getLandmarkID());
  end if
end if

3.2.2. Routing Mechanism in Landmark Nodes

Once the packet reaches the first landmark, it starts to be routed at the landmark level and only leaves it when the destination is found as can be seen in Alg. 2. In the communication between landmarks, common nodes are used to forward the packet. For this scenario, the Landmark ID field is filled up with the next landmark ID and the common nodes in the path follow the Alg. 1 as described before.

Algorithm 2 landmarkNodesPacketForwardingEngine($packet$)

$packet$.traversedLandmarkPath.append($myID$);
$nHop$ = fib.findSpecificID($packet$.getDestinationID());
if $nHop \neq NULL$ **then**
  forwardPacket($nHop$);
else
  $nHop$ = lib.findNextLandmark($packet$.getTraversedLandmarkPath());
  forwardPacket($nHop$);
end if

Similar to the routing inside regions, at the landmark level the network proximity is also preserved. To this aim, the forwarding decision taken by the *findNextLandmark* method considers the already visited landmark path present in the packet header while searching the next landmark in the Landmark Information Base (LIB) and, among other possible landmarks, the closest one is chosen.

Currently, the LFR routing model at landmark level is greedy, i.e., if it is necessary, all regions are visited to deliver the packet. We are working in an inter-region mechanism for the LFR proposal that eliminates the greedy approach. This is left for future work.

3.3. Discussion

In terms of scalability, one would argue that the landmark is a bottleneck because it concentrates all the registry information of a region. However, no aggregation is possible in a flat identity space scenario. Consequently, all proposals that consider a scenario like this require at least *n* registry information in the network (where *n* is the total number of nodes in the network). Under this circumstance, the LFR proposal has an important benefit that is the possibility to create as many landmark regions as necessary to achieve better system scalability.

Another bottleneck that landmarks may introduce is related to traffic delivery centralization for the inter-region scenario. Nevertheless, some of the communications between two nodes located at different regions can occur without going through the landmarks. Presently, this feature is only achieved if border nodes receive the packet and its physical neighbor (located at the other landmark region) represents progress in the XOR operation. We are also working on some extensions aimed at reducing this traffic delivery concentration at the landmark.

4. A Tool for Flat Routing Study

It is a challenging task to evaluate new routing proposals like the LFR. The most common limitations include 1) the complexity or impossibility to introduce new protocols in the network equipments; 2) the cost of equipments and 3) the difficult to build the topology for evaluations (for example, the Internet topology is very complex).

These restrictions lead us to the development of a tool that emulates network elements and contemplates their independence as if they were real hardware. The tool allows a) the evaluation of new routing protocols in the emulated network elements; b) the simplification of the topologies deployment and 3) the gathering of statistics about the protocol behavior. It is possible to emulate any kind of physical topology and perform the evaluation of new routing protocol implementations.

The tool has a forwarding plane that encompasses the basic interconnection functions. Links and physical ports in a network element are emulated using TCP connections. This decision has the benefit of not having packet loss and the initial TCP three handshakes' overhead is absorbed by keeping the connection open during the whole evaluation like if it were a real physical link. Another important characteristic is that a network element can open as many ports as necessary, resulting in an unspecified variety of network elements with different number of ports.

The network elements in the tool have an engine in the forwarding plane that contains the routing protocols implementations. It defines the format and the respective treatment for the exchanged signaling messages and is also responsible for the data forwarding according to the routing algorithm. It is important to emphasize that each network element is implemented as an isolated thread. Consequently, the communication between different elements only occurs through message exchange according to the routing protocol specification. In this way, all actions taken by one element are considered real, i.e., they are the same as if they were running in real equipments. There is no protocol simulation in the tool.

Another possibility of the tool is to distribute the topologies in different computers. Consider an evaluation scenario with 256 nodes as depicted in Fig. 5. It is possible to run this topology using just one computer or to distribute it across *n* (e.g. 256) computers. The limit of nodes in a computer is defined by the resources available in that computer (processor and memory) and the 16 bits TCP ports available in a computer (65k ports). Each network element maintains a multi-thread TCP server running in a port associated to its identifier; it also has one TCP port dynamically allocated by the operating system for each link between the element and each of its neighbors. To perform load distribution, it is necessary to allocate the total number of nodes among the participating computers by defining a file with the distribution rules. Fig. 4 shows a distribution example: nodes from 0 to 100 are located at computer 192.168.0.10; nodes from 101 to 150 are located at computer 192.168.0.11; nodes from 151 to 255 are located at computer 192.168.0.12.

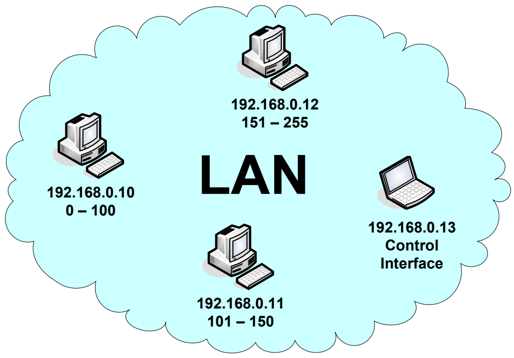

Figure 4. Example of node distribution.

In order to simplify the management of scenarios, there is a control interface to trigger actions on all nodes. This control interface can run on any computer in the LAN (192.160.0.13 in Fig. 4). The nodes in the tool have their port 0 reserved for the control interface. This port is used to dispatch commands to nodes inside the topology. Examples of commands include requests to start physical connections, to start neighborhood discovery and to send data.

The tool includes useful functions to calculate the route stretch. As it has independent nodes and generates real routing tables, it is possible to perform real packet forwarding. To this aim, the control interface injects a data packet in the source node and the packet is routed at each node in the path. At the same time, the node identifier of each visited node is appended to the packet. When the packet reaches the destination node, it sends the packet back to the control interface that is responsible for logging the information carried in the data packet (the traversed path). In this way, none simulation is used to determine paths. The control interface allows sending packets from one specific node to another specific node; from one specific node to all other nodes; from all nodes to one specific node and from all nodes to all nodes.

5. LFR Evaluation

The objective of this section is to present some quantitative results regarding the flat routing proposal presented in this paper. These results were collected using the developed tool and, as such, this section has also the objective of demonstrating its usability. In this work, two different flat routing protocols were implemented and some data were collected to compare both implementations. The first protocol implements a flat routing scenario in which all participant nodes must follow the connectivity invariant as done in UIP. In this case, there is no landmark. What differs this implementation from the original UIP is the lack of a layer-3 connectivity between neighbors and, as such, this implementation was denominated UIP'. The second scenario implements our LFR proposal as described in Sec. 3 in which the connectivity invariant is respected inside regions and all nodes are registered at the landmark.

To evaluate both protocols, two different 256 nodes topologies and an eight bits identity space were used. The first topology presented in Fig. 5 has a regular (ad hoc like) mesh format and the node identifiers (0 to 255) were randomly generated. The second topology shown in Fig. 6 has also 256 nodes and is extracted from the RouteViews BGP AS links [CAIDA 2008]. To generate this last topology, the Embratel BGP ID (4230) was found and other 255 neighbors were selected. All links between the 256 selected nodes were preserved. The identifiers were normalized according to the original BGP identifiers in a way that the smallest identifier was set to 0 and the biggest to 255.

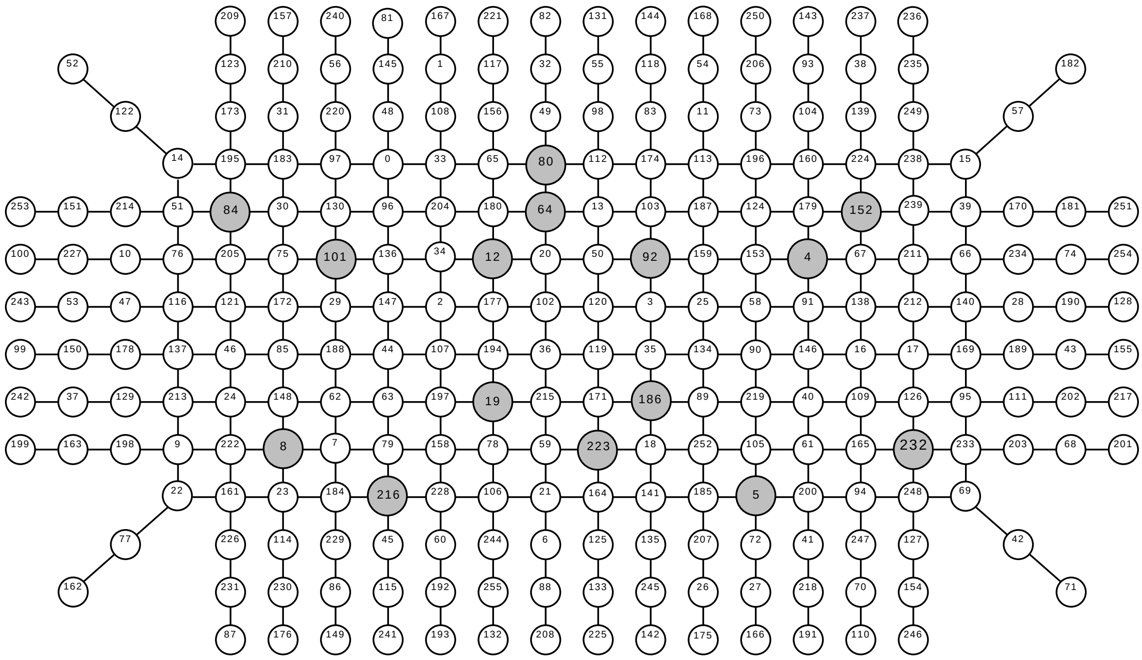

Figure 5. Landmark positioning for the Regular mesh topology with 256 nodes.

Landmarks are required for the tests using the LFR proposal. In Figs. 5 and 6 their positions are highlighted by bigger gray circles. As can be seen in Fig. 5, there are fifteen landmarks defined in it. Four of them are used in the first test which compares LFR and UIP' implementations and all of them are used in the second test that shows the influence of the number of landmarks in the system.

The chosen landmark IDs for the regular mesh topology in the first test are 12, 19, 92 and 186. Their positions were selected in a way that the topology was uniformly divided into four regions. In the case of the Embratel topology[2], the selected landmarks positions are based on the cluster organization commonly found in Internet-like topologies. The chosen companies include Embratel (BGP ID 4230), Bahia Telecommunications (BGP ID 7738), California State University (BGP ID 2153) and Telefônica (BGP ID 10429). As mentioned before, the BGP IDs were normalized and, as a result, Embratel is Flat ID 21, Bahia Telecom is Flat ID 31, California State University is Flat ID 14 and

[2] In the printed version of this paper, it is only possible to see the IDs used for the landmarks. However, in the digital version, the figure has high resolution and the original BGP ID and the normalized Flat ID for all nodes can be seen using zoom on the computer screen.

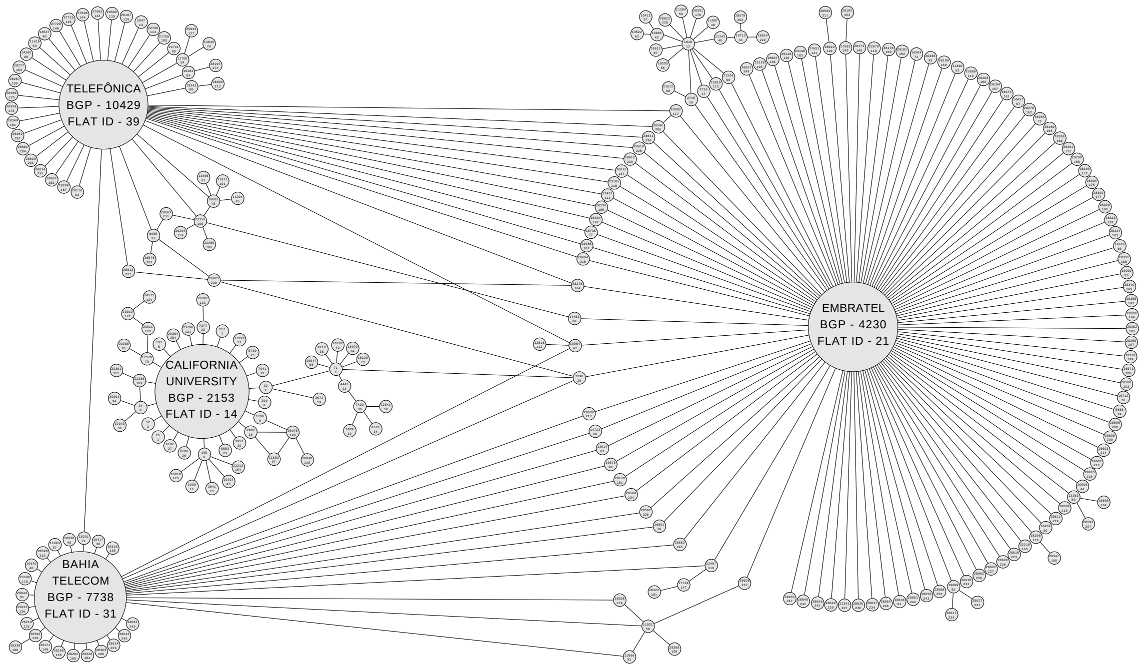

Figure 6. Landmark positioning for the Embratel topology with 256 nodes.

Telefônica is Flat ID 39.

To demonstrate the influence that the quantity of landmarks introduces to the overall system complexity, three different scenarios were evaluated varying the quantity of landmarks using the LFR implementation under the regular mesh topology. The first scenario used four landmarks (IDs 12, 19, 92 and 186). The second scenario used five landmarks (IDs 4, 5, 80, 101 and 216). The third scenario used six landmarks (IDs 8, 64, 84, 152, 223 and 232). Their positions can be seen in Fig. 5.

The hardware used for the tests includes three desktops with Intel Core 2 Duo processor E6400 (2.13 GHz, 1066 MHz FSB, 2 MB L2 cache) with 4GB DDR2 running Debian Sarge and a notebook with an Intel Core Duo processor T2300E (1.66 GHz, 667 MHz FSB, 2 MB L2 cache) with 2GB DDR2 running Kubuntu. The distribution rules were set as the example presented in Fig. 4 with nodes instantiated on the desktops and the control interface in the notebook. The objective of distributing the topologies used in this paper in different computers is only to demonstrate the distribution feature of the tool that is important for bigger topology evaluations.

The first result that is interesting to include in the paper is related to memory consumption. The sum of the memory consumption in all the three desktops is around 120MB of RAM as indicated by the JConsole tool included in the SUN JAVA package. Each individual node consumes about 470 KB of RAM when the scenario is ready to send data. Obviously, in a bigger scenario, nodes are intended to have more neighbors and, as a consequence, bigger memory consumption is expected. However, the amount of extra memory required is not expected to be enormous. Considering this value, it seems to be possible to evaluate topologies with approximately 8000 nodes in each desktop with 4GB DDR2 RAM if the number of edges between nodes (TCP ports) required in the topology is not prohibitive.

In the first set of scenarios, four landmarks were used and four tests were performed: 1) UIP' in the regular mesh network; 2) LFR in the regular mesh network; 3) UIP' in the Embratel network and 4) LFR in the Embratel network. The quantitative results include the routing table size (Tab. 1), the number of signaling message exchanged by the protocols (Tab. 2) and the route stretch (Tab. 3). To compute the routing stretch, a Dijkstra algorithm was used to find the shortest paths.

As can be observed in Tab. 1, the amount of routing table entries for the LFR proposal is significantly smaller than UIP'. The main reason for this is the LFR concept of landmark regions that limits the scope of the neighborhood discovery process.

Table 1. Routing tables size.

Test Scenario	Average Size	Minimum Size	Maximum Size	Std. Deviation
UIP' - 256 Regular	130.06	31	226	52.89
LFR - 256 Regular	66.38	61	74	2.58
UIP' - 256 Embratel	123.54	31	251	52.02
LFR - 256 Embratel	75.26	21	131	36.03

The values presented in Tab. 2 refer to the protocol complexity from the point of view of the signaling messages used for the construction of the routing tables. In both protocols, two messages are exchanged during the neighborhood discovery process, one discovery message asking for desirable neighbors to fill the routing table and one message with the answers. Specially for the LFR scenario, two other per node messages are required to register them at the landmark (registry and acknowledge).

Table 2. Exchanged signaling messages.

Test Scenario	Average	Minimum	Maximum	Std. Deviation
UIP' - 256 Regular	139.36	34	303	53.02
LFR - 256 Regular	160.87	106	224	19.24
UIP' - 256 Embratel	140.09	14	399	73.89
LFR - 256 Embratel	99.11	4	208	45.11

In the Embratel scenarios, the LFR average value presents a significant difference when compared with the UIP'. As can be observed in Fig. 6, Internet-like topologies naturally introduces the regionalism concept (four main regions are present in the map) and the LFR proposal naturally takes advantage of this property. The distance in number of hops between nodes inside a region is small. In the Embratel landmark region, a considerable number of nodes is directly (physically) connected to the Embratel landmark. For that reason, when one node looks for neighbors, it finds several in the first message sent to the landmark.

The main result present in Tabs. 1 and 2 is related to the linearity introduced by the landmark usage. Nodes inside a region have equivalent participation on the construction of the routing tables and, consequently, exchange a similar number of messages. If the number of messages and routing table entries start to demonstrate some scalability problem, the LFR allows the definition of more landmark regions. Conversely, the UIP' connectivity invariant is directly associated to the network size. If the network grows, more signaling messages and entries in the routing table are required.

Table 3 shows the route stretch for all source and destination combinations. The topologies have 256 nodes, resulting in 65280 paths for each scenario. One interesting data found in [Krioukov et al. 2007] is confirmed by the results of Tab. 3. Compact routing proposals have a defined lower bound stretch for abstract graphs. The better proposals found in the literature reached stretch 3 on name-independent networks. However, when these theoretical models were applied on Internet-like topologies, always denominated power-law topologies, the obtained stretches were smaller. The same occurred here and specially in the LFR implementation.

Table 3. Route Stretch.

Test Scenario	Average	Minimum	Maximum	Std. Deviation
UIP' - 256 Regular	1.03	1	2.33	0.1
LFR - 256 Regular	1.48	1	9.67	0.69
UIP' - 256 Embratel	1.01	1	1.67	0.05
LFR - 256 Embratel	1.33	1	4.5	0.41

The stretch results present in Tab. 3 are expressive since LFR has considerable smaller routing tables as presented in Tab. 1. The smaller route stretch values found for the UIP' scenarios occur due to the neighborhood discovery process in which paths with smaller distances (normally shortest paths) are stored in the routing tables. Still analyzing Tab. 1, it is possible to see that the maximum number of routing entries for UIP' are almost 256 (n), i.e., the connectivity invariant leads some nodes in the UIP' scenarios to have an elevated number of neighbors. This confirms the correctness of the trade-off between the amount of information available to compute routes (number of routing table entries) and the quality of the routes that are generated (route stretch).

The results for the second set of evaluations are compiled in Tab. 4. It is aimed at demonstrating how the quantity of landmarks can interfere in the overall system complexity. Due to space limitations, all the results were grouped into one table with the number of routing table entries, the number of signaling messages and the route stretch for the scenarios with 4, 5 and 6 landmarks.

Table 4. Results for uncommon landmark positioning scenarios.

Test Scenario	Average	Minimum	Maximum	Std. Deviation
Number of R. Table Entries (4 landmarks)	66.38	61	74	2.58
Number of R. Table Entries (5 landmarks)	53.66	42	71	8.08
Number of R. Table Entries (6 landmarks)	45.5	32	59	7.55
Number of Sig. Messages (4 landmarks)	160.87	106	224	19.24
Number of Sig. Messages (5 landmarks)	119.2	46	184	28.01
Number of Sig. Messages (6 landmarks)	105.21	41	156	23.24
Route Stretch (4 landmarks)	1.48	1	9.67	0.69
Route Stretch (5 landmarks)	1.76	1	13.67	1.01
Route Stretch (6 landmarks)	1.87	1	15.67	1.13

As can be seen in Tab. 4, by increasing the number of landmarks, the number of routing table entries and the number of signaling messages decreases, but the route stretch increases. One explanation for it is the trade-off between the size of routing tables and the stretch described before. Other two reasons for this are the increase in the number of regions to visit when the packet reaches the landmark level and the increase in the distance between the landmarks. For the first reason, as mentioned in Sec. 3, the current routing at landmark level is greedy and a solution to this in under investigation. For the second reason, the positioning of landmarks by itself is a hot topic and requires deeper investigations in future work.

6. Conclusion and Future Work

This paper introduces the LFR and focused on how the "generic" LFR behaves performing flat routing directly on flat identifiers. The results presented in this paper are promising and show that not only the LFR proposal is interesting but also that the evaluation tool has potential to emulate bigger network topologies. In future works, extensions to LFR for specific contexts like ad hoc networks and Internet-like scenarios will appear using bigger topologies. We expect to use topologies in the order of thousands of nodes in the ad hoc scenario and the whole inter-domain topology with approximately 35000 ASes [CAIDA 2008] in the Internet scenario.

Currently, the LFR inter-region routing is greedy. Under flat scenarios, none aggregation is possible and, as such, disseminating information between regions is difficult. We consider that Bloom filters inspired data structures can have an important contribution in this scenario. The dissemination of Bloom filters with the information of the nodes in a region opens up the possibility of knowing in which regions a certain destination is not present. Note that this mechanism is a dual to the current BGP dissemination that indicates where a node is present. As the Bloom filter of the regional landmark is created, it is also interesting to disseminate it internally to the region. This internal dissemination has the benefit of giving a real knowledge to nodes inside a region that a destination is not inside it. It not only reduces the route stretch by avoiding unnecessary forwarding inside the regions, but also allows the inter-region packets forwarding without centralizing it at the landmarks.

Considering a big network topology, the number of landmarks tends to increase. As a consequence, the elevated number of landmarks can also become a scalability problem. In this way, LFR supports the definition of a landmark hierarchy; this hierarchy can be built with many levels to assure scalability. The Bloom filters inside each hierarchy level are ORed and disseminated in the above level of the hierarchy. This is also subject of future investigations.

References

Broder, A. and Mitzenmacher, M. (2004). Network Applications of Bloom Filters: A Survey. *Internet Mathematics Vol. 1, No. 4: 485-509.*

Caesar, M., Castro, M., Nightingale, E. B., O, G., and Rowstron, A. (2006a). Virtual Ring Routing: Network Routing Inspired by DHTs. *In Proc. of ACM SIGCOMM, Pisa, Italy*, pages 351 – 362.

Caesar, M., Lakshminarayanan, K., Condie, T., Stoica, I., Kannan, J., and Shenker, S. (2006b). ROFL: Routing on Flat Labels. *In Proc. of ACM SIGCOMM, Pisa, Italy*, pages 363 – 374.

Caesar, M. C. (2007). Identity-based routing. *Technical Report No. UCB/EECS-2007-114 - University of California, Berkeley, USA.*

CAIDA (2008). The Cooperative Association for Internet Data Analysis, http://www.caida.org.

Ford, B. (2003). Scalable Internet Routing on Topology-Independent Node Identities. *MIT technical report conducted as part of the IRIS project (http://project-iris.net).*

Ford, B. (2008). UIA: A Global Connectivity Architecture for Mobile Personal Devices. *Ph.D. Thesis. Massachusetts Institute of Technology (MIT), Dep. of Electrical Eng. and Comp. Sci.*

Krioukov, D., k c claffy, Fall, K., and Brady, A. (2007). On Compact Routing for the Internet. *ACM SIGCOMM Comput. Commun. Rev.*, 37(3):41–52.

Tsuchiya, P. F. (1988). The Landmark Hierarchy: A new Hierarchy for Routing in Very Large Networks. *ACM SIGCOMM 1988, Stanford, CA, USA*, pages 35–42.