Exploiting Locality to Provide Adaptive Routing of Real-Time Flows in Global Internets: Abstract

Lee Breslau, Deborah Estrin, Daniel Zappala, and Lixia Zhang[1]

1 Introduction

This paper addresses the role of routing in supporting real-time applications across large integrated services internetworks. The increasing speed of underlying transmission media is enabling new, and perhaps more importantly, integrated applications. However, because of the size and decentralized nature of the global internet, we cannot assume that the network will be homogeneously capable of all new services. Therefore, routing must identify routes with specific type-of-service (TOS) capabilities. Moreover, we suggest that TOS-routing for performance-sensitive applications (e.g., real-time videoconference, real-time interaction with high-bandwidth sensor or experimental data) should be sensitive to significant load-changes, as well as to topology changes. At the same time, global scale means that we must minimize reliance on global consistency of routing databases and global distribution of dynamic routing information.

Today's internet routing *does not* adapt to load shifts for two sound reasons–stability and overhead. Load-dependent algorithms can become unstable, simultaneously directing all nodes to respond to load changes in the same way, causing oscillation. Further, distributing fine grain load information throughout the internet (a requirement for loop-free routing in today's hop-by-hop routed Internet) would introduce excessive overhead. The Internet avoids global distribution of fine grained information by using hierarchical aggregation. If it distributes load information, it does so only within local domains, hiding fine detail between domains.

Hierarchical aggregation is one of the most powerful ways of dealing with scale[2]. However, hierarchical aggregation does not provide information about the availability of real time routes across a wide-area, heterogeneous internet. For example, if the internetwork routing updates only contain information about configured, inter-domain, topology, then multiple routes to each destination will be computed and selected based only on the relatively-static information. If at call setup time the selected route is not available, or has insufficient resources to carry the flow, an alternate route will be tried. However, such a trial-and-error approach may introduce intolerable session-startup delays, especially in a high-speed network environment where many data transfers could be completed during the several round-trip times that it might take to discover a route. On the other hand, the overhead of flooding dynamic link or path status information is unacceptable in a large network.

Our routing architecture addresses this problem by (a) advertising a different level of detail for those routes that are currently carrying real-time multimedia applications, than for routes that are carrying more generic (less performance sensitive) traffic, and (b) distributing this information to current users of the resource in question, not to the global network. We thereby exploit the *route locality* of internetwork traffic. Route locality is measured by the mapping of each source to the set of network resources traversed to reach the destinations with which it is communicating. The proposed mechanisms are used to complement, not replace, traditional hierarchical-aggregation techniques (i.e., inter-domain routing).

We present what might appear to be an unsupportable requirement, i.e., different levels of routing detail from and to different nodes in the network. However, our proposed solutions can be made to work in the context of source routing. Our source routing uses a link-state style algorithm to compute routes [3] [4]. Domain-level,

[1] L. Breslau, D. Estrin, and D. Zappala are with the Computer Science Department, University of Southern California, Los Angeles, CA. e-mail: breslau@jerico.usc.edu, estrin@usc.edu, dzappala@jerico.usc.edu. L. Zhang is with Xerox Palo Alto Research Center. e-mail: lixia@parc.xerox.com

[2] Leonard Kleinrock and Farouk Kamoun. Hierarchical routing for large networks: Performance evaluations and optimization. *Computer Networks*, 1:155-174, 1977.

[3] Lee Breslau and Deborah Estrin. Design and evaluation of inter-domain policy routing protocols. *Internetworking: Research and Experience*, 2(3), 1991.

[4] David D. Clark. Policy routing in internetworks. *Internetworking: Research and Experience*, 1(1):35-52, 1990

configured topology information is distributed globally to route server(s) in each domain (by using event-driven distribution combined with very low frequency periodic updates). This information is relatively stable so the overhead of distribution and storage is acceptable; the size of this global map is proportional to the number of domains and inter-domain links in the internet. Similar functionality might be achievable with a hop-by-hop routing approach in a connection-oriented network. However, hop-by-hop routing decisions must be consistent across all nodes. Therefore the dynamic information would have to be distributed globally, to allow *all* the nodes along the path to choose the *same* route that meets an individual source's specific requirements. Source routing allows us to route without loops even when route computation and selection is based on *inconsistent network maps.*

A key component of the **Adaptive Source Routing** architecture proposed is the limited distribution of routing updates, using the **Reverse Path Update (RPU)** protocol. RPU provides each domain's route server(s) with more detailed, and more timely information about the parts of the network that are used by clients of that particular route server (i.e., users in that particular domain). The RPU update information consists of averaged load measures and up/down status. To the extent that the load averages are a good predictor of actual load in the future (i.e., load does not fluctuate too rapidly so that the average interval can be significantly longer than the round-trip-time), then path selection will have enough information to select a route that has sufficient resources to carry the flow. The trial-and-error, session startup delays, referred to earlier, will only occur when a user initiates communication that traverses previously unused portions of the internet; the frequency of this is inversely proportional to the amount of route locality. Our objective is to achieve performance comparable to that achievable with global distribution of dynamic information, but with significantly reduced overhead (depending on route locality). Moreover, the overhead of this approach does not increase linearly with the size of the network, as does global flooding.

RPU works in the following manner. When an inter-domain connection experiences a change in state (i.e., fails, resumes service, passes a load-threshold), the inter-domain router that detects the change distributes the information only to the sources of routes that currently pass through the domain. This does not mean that information is distributed only to sources that pass through the affected physical router/link; in order to exploit route locality, information is distributed to active routes that pass through any of the border routers belonging to that same domain. The update is propagated as in an intelligent global flood; however instead of propagating over the full network graph, the updates are forwarded only over the subgraph defined by the active routes that traverse the affected resource. This is achieved by including the route identifiers of active routes inside the update. Each router then propagates the update only along links that are part of at least one of the route IDs listed. As a result, the updates do not travel to parts of the network that are not concerned with the update information.

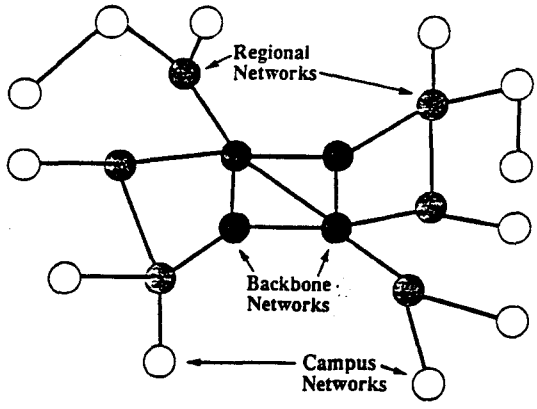
An extended version of the paper describes RPU in more detail and evaluates its overhead savings relative to global flooding using simulations of many networks with varying topologies and route locality. To summarize our findings, we found that RPU achieves significant savings over global flooding for what we believe are realistic route locality measures. That is to say, when the the percentage of transit nodes through which each source routes its traffic is small (which we expect to be in future internets), RPU propagates fewer update messages than global flooding. For example, for 6 different networks of 50- and 100- nodes, when source nodes send their traffic through 30% of the transit nodes, the overhead of RPU was between 14% and 33% of the overhead of flooding. Since each source uses only a relatively small number of transit nodes, each time a transit node initiates an RPU update, that update will be sent only to a small number of source nodes. RPU savings are increased on networks with higher connectivity, since flooding will send an update over all links, while RPU controls propagation by sending the updates along active source routes. As route locality approaches 1.0 (which is not likely in actual internets), the overhead of RPU update distribution approached that of global flooding.

In addition to continuing development of the limited distribution protocol described here, we are working on other critical aspects of the architecture, in particular route selection. Through simulation, emulation, and experimentation we hope to show that our adaptive source routing architecture provides significantly improved service to real-time communication in large, heterogeneous internets.

Exploiting Locality to Provide Adaptive Routing of Real-Time Flows in Global Internets

Lee Breslau (USC)
Deborah Estrin (USC)
Daniel Zappala (USC)
Lixia Zhang (Xerox PARC)

April 3, 1992



How can routing provide good support to real-time applications across a heterogeneous wide-area internet?

Outline

Internet Model

Requirements for Routing
- performance sensitive
- scale

Routing Architecture

Update Strategies
- exploiting locality
- limited distribution

Simulation Results

Conclusions

Internet Model

1000s of interconnected administrative domains

Stub domains -- sources/sinks of traffic
- e.g., campus networks

Transit domains -- carry traffic for stubs
- e.g., backbone, regional, private networks

Heterogeneity -- not all networks offer the same services

Topology -- generally hierarchical, but with exceptions
- --> Multiple routes available between src/dest

Applications -- diverse requirements
- e.g., **multimedia**

Requirements for Routing

Provide paths meeting application requirements

- Not all nets capable of all services (e.g., service guarantees)
- Variability/unpredictability of traffic patterns
- A route's ability to provide required service varies over time
- Shortest path (in terms of number of hops) may be highly loaded, making a longer alternate path more desirable

Requirements for Routing

Scale

- 1000s of domains. Routing should not be O(N
- Avoid frequent, global distribution of routing information
- Avoid dependence on globally consistent routing information

Source Routing Architecture

Why Source Routing?

- Eliminate need to maintain consistent routing information at all nodes while still avoiding routing loops
- Relieves transit nodes from computing routes

Overview of Routing Architecture

Global distribution of configured topology

Source computes multiple routes to each destination

Distribution of dynamic routing information

Route selection based on available dynamic information

Forwarding information installed during route setup

Information Distribution

Route Selection can be improved when based on dynamic information
- fewer failed setups
- better quality routes

However, distribution of dynamic information implies increased overhead

Goal: Develop a scalable information distribution strategy that allows good route selection decisions

Information Distribution Strategies

Option 1: No dynamic information -- trial and error approach
- acceptable overhead -- only distribute configured topology globally
- lack of dynamic information leads to more failed setups and/or degraded service

Option 2: Global distribution of dynamic information
- improved route selection based on complete information, but
- unacceptable overhead

We are motivated to find partial distribution strategies that balance the distribution overhead with quality of routes selected

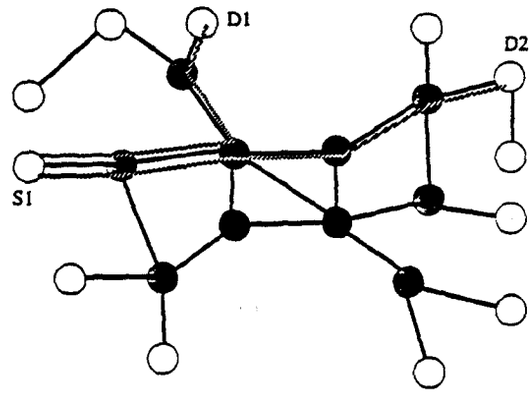
Exploiting Traffic Locality

Traffic patterns are not uniform

Single source communicates with only a small subset of destinations

Routes to these destination traverse only a subset of all transit domains

Source wants dynamic information about the subset of transit domains through which it routes its traffic



Routes currently used by S1 to reach D1 and D2 traverse only a subset of the transit nodes

Reverse Path Update

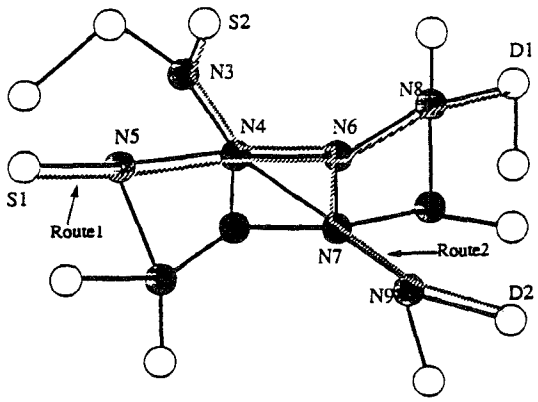
Distribute routing updates along currently active source routes

A node experiencing a change (up/down or load) uses its forwarding table to determine where to send updates

Source learns about changes in status of domains through which it forwards its traffic

Use this dynamic information in future route selection decisions (or to reroute existing sessions in some cases)

RPU Example



When N6 initiates an RPU update:

1.N6 sends an update along Route1 to N8

2.N6 sends an update along Route2 to N7

3.N6 sends a single update along Route1 and Route2 to N4

4.N4 sends one copy of this update along Route2 to N3, and a seco copy along Route 1 to N5

Simulation Results

Compare overhead of RPU updates to global flooding

Simulated the protocol in different networks

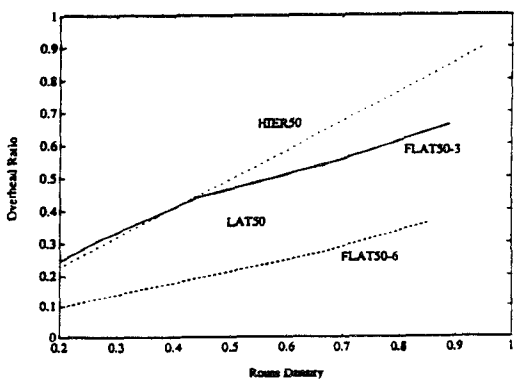
- size (18, 50 and 100 nodes)
- topology (flat, hierarchy, hierarchy w/ lateral links)
- connectivity

For each network, tested several different traffic matrices

- varied number of active src/dest pairs
- associated route density measure with each traffic matrix

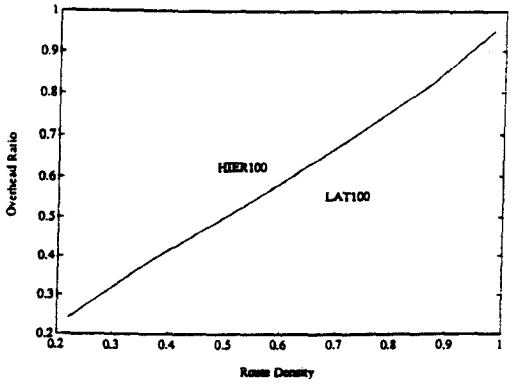
Measured overhead of RPU distribution for all single link failures

Ratio of RPU overhead to global flood for 50-node networks



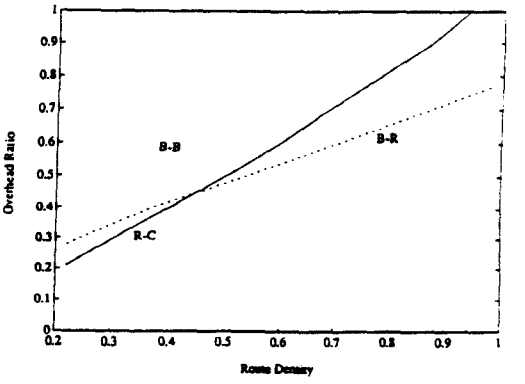
FLAT50-3 - Flat topology, average connectivity = 3
FLAT50-6 - Flat topology, average connectivity = 6
HIER50 - 3 level hierarchical network
LAT50 - 3 level hierarchical network with lateral links

Ratio of RPU overhead to global flood for 100-node networks



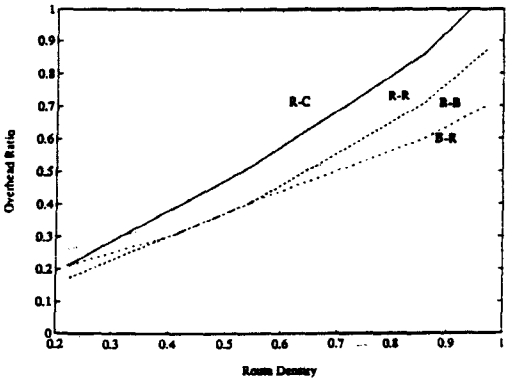
HIER100 – 3 level hierarchical network
LAT100 – 3 level hierarchical network with lateral links

Ratio of RPU overhead to global flood for 100-node hierarchical network by link type



B-B – Links connecting two backbone networks
B-R – Links connecting backbone to regional networks
R-C – Links connecting regional to campus networks

Ratio of RPU overhead to global flood for 100-node hierarchical network with lateral links by link type



B-B – Links connecting two backbone networks
B-R – Links connecting backbone to regional networks
R-R – Links connecting two regional networks
R-C – Links connecting regional to campus networks

Conclusions

RPU reduces overhead of information distribution under certain traffic conditions

Overhead reduction depends on topology and on traffic density

Future Work

Consider effect of incomplete information on route selection

Develop algorithms that take into account the partial information and assess the quality of routes computed