Traffic Engineering Based on Local States in Internet Protocol-Based Radio Access Networks

David A. Barlow, Vasos Vassiliou, Sven Krasser, Henry L. Owen, Jochen Grimminger, Hans-Peter Huth, and Joachim Sokol

Abstract: **The purpose of this research is to develop and evaluate a traffic engineering architecture that uses local state information. This architecture is applied to an Internet protocol radio access network (RAN) that uses multi-protocol label switching (MPLS) and differentiated services to support mobile hosts. We assume mobility support is provided by a protocol such as the hierarchical mobile Internet protocol. The traffic engineering architecture is router based—meaning that routers on the edges of the network make the decisions onto which paths to place admitted traffic. We propose an algorithm that supports the architecture and uses local network state in order to function. The goal of the architecture is to provide an inexpensive and fast method to reduce network congestion while increasing the quality of service (QoS) level when compared to traditional routing and traffic engineering techniques. We use a number of different mobility scenarios and a mix of different types of traffic to evaluate our architecture and algorithm. We use the network simulator ns-2 as the core of our simulation environment. Around this core we built a system of pre-simulation, during simulation, and post-processing software that enabled us to simulate our traffic engineering architecture with only very minimal changes to the core ns-2 software. Our simulation environment supports a number of different mobility scenarios and a mix of different types of traffic to evaluate our architecture and algorithm.**

Index Terms: **Differentiated services (DiffServ), multi-protocol label switching (MPLS), quality of service (QoS), radio access networks, traffic engineering.**

I. INTRODUCTION

This research develops and evaluates a router-based traffic engineering architecture and algorithm for a radio access network (RAN). The RAN supports mobile nodes with quality of service (QoS) traffic requirements. Our goal for the traffic engineering architecture is a fast, inexpensive, and effective system. Most previous traffic engineering research has focused on the core Internet and optimization and has not considered the decision speed requirements of a RAN. Mobile nodes may spend very little time in any particular radio cell, depending on their speed and the RAN physical design. In order for a traffic engineering system to be effective in this type of environment, it must be fast enough to enable the specific routing of all, or nearly all, QoS packets. This traffic engineering system must also be inexpensive in terms of resources required (money and information), as any traffic engineering system must compete with over-provisioning as a solution to the QoS problem.

A question to consider is: When do you need to use traffic engineering in an Internet protocol (IP) RAN? A RAN is by definition a relatively small network, so it contains no end-to-end paths with large numbers of hops to delay QoS traffic. Therefore, in an uncongested network, traffic engineering should not be necessary. In a heavily, uniformly congested network, there is no place to alternately route additional traffic, so traffic engineering will have little or no benefit. It is when there is congestion in a limited number of specific points in the RAN that traffic engineering could possibly be useful since there is likely to be alternate capacity elsewhere in the RAN that can be used to alt-route traffic. This is the type of traffic scenario, perhaps unique to mobile communications access networks, that we hope to address with our protocol—point loads on an IP RAN that move over time.

There are two major items that can affect the speed with which one can implement and operate traffic engineering in an IP network. First, the setup of specific paths has to be considered. Second, collection and dissemination of accurate network QoS state is another factor. Our work addresses these issues as well as the need for a simple solution that would be easy and inexpensive to implement.

In order to apply traffic engineering to any network, some set of multiple paths between routers is necessary. If there are only single paths between network routers, then there is nowhere to move traffic; no space exists to conduct traffic engineering. Our assumption is that a RAN is a specialized network. It is relatively small (less than 50 nodes) and designed to be replicated as part of a commercial information network package that can be installed in a standardized configuration. Our simulation work uses a RAN configuration that has a minimum number of links yet provides alternate routes within the network. Other configurations are possible, but we leave this to further study. Our simulation goal is to show that our traffic engineering architecture can work, not that it is a general solution for all possible network configurations.

We begin with some essential background information in Section II; follow with an overview of our research, proposed traffic engineering architecture, algorithm, and simulator in Section III and Section IV; present some simulation results in Section V; and finally discuss our conclusions in Section VI.

Manuscript received July 20, 2004; approved for publication by Wha Sook Jeon, Division III Editor, February 21, 2005.

D. A. Barlow is with the Department of Electrical and Computer Engineering, United States Military Academy, West Point, New York, USA, email: david.barlow@usma.edu.

V. Vassiliou is with the Department of Computer Science, University of Cyprus, Nicosia, Cyprus, email: vasosv@ucy.ac.cy.

S. Krasser and H. L. Owen are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA, email: {sven, owen}@ece.gatech.edu.

J. Grimminger, H.-P. Huth, and J. Sokol are with Siemens AG, Munich, Germany, email: {jochen.grimminger, hans-peter.huth, joachim.sokol} @siemens.com.

II. BACKGROUND

In this section, we outline the technologies and concepts used in our research. It is important to note that we actively use multi-protocol label switching (MPLS) and differentiated services (DiffServ) in our simulations. However, we assume the existence of the mobile IP protocols described below but make no effort to include them in our simulations or measure any possible interactions they may have with our traffic engineering architecture. We save this as a topic for additional research.

A. MPLS and Traffic Engineering

MPLS is a packet forwarding technology that assigns packet flows to label switched paths (LSPs). Packets are classified at the network edge, labeled, and then transported over an LSP. LSPs can be explicitly routed, and thus MPLS enables traffic engineering in IP networks. Current IP routing protocols employ a shortest-path algorithm. During periods of heavy network loading, shortest path routes can become congested despite underutilized capacity elsewhere in the network. Traffic engineering, by choosing routes using some algorithm, can shift some of the load to underutilized parts of the network. Traffic engineering also allows flexibility and economy in network design. By enabling efficient use of network resources, traffic engineering supports alternate physical paths and reduced link size to achieve throughput and reliability goals.

B. Mobile IP

Mobile IP allows a mobile node to move from one link to another without changing the mobile node's home IP address. A home address is an IP address assigned to the mobile node within its home subnet prefix on its home link. Packets may be routed to the mobile node using this address regardless of the mobile node's current point of attachment to the Internet, and the mobile node may continue to communicate with other nodes (stationary or mobile) after moving to a new link. While a mobile node is attached to some foreign network, it is also addressable by one or more care-of addresses. While away from home, a mobile node registers one of its care-of addresses with a router on its home link, requesting this router function as the home agent for the mobile node. The home agent intercepts and forwards packets to the mobile node [1].

C. Hierarchical Mobile IP

Hierarchical mobile IP (HMIP) is a mobile IP micro-mobility management model. Its purpose is to reduce the amount of signaling to correspondent nodes and the home agent and improve the handoff speed performance of the mobile Internet protocol (MIP). Hierarchical mobile IP version 6 (HMIPv6) [2] introduces a mobility anchor point, and minor extensions to mobile node and home agent operations. The mobile node has two addresses, a regional care-of address on the mobility anchor point's subnet and an on-link local care-of address. The mobility anchor point acts as a local home agent that maps the mobile node's regional care-of address to a local care-of address. When the mobile node moves locally (i.e., its mobility anchor point does not change), it needs only to register its new local care-of address with its mobility anchor point. The regional care-of address stays unchanged. A mobile node can inform correspondent nodes who share the same mobility anchor point of its local care-of address instead of its regional care-of address. Packets can then be routed directly without going through the mobility anchor point. This direct routing is essential to successful traffic engineering within a RAN. Otherwise, all intra-RAN traffic will be sent through the mobility anchor point (assuming one mobility anchor point in the RAN).

D. Differentiated Services

Differentiated services (DiffServ) [3] is an Internet QoS system designed to provide different types of IP service to meet general classes of user requirements. DiffServ aggregates flows that require similar QoS treatment and sends them through the network as a behavior aggregate. The compromise of aggregate traffic handling is that the QoS enjoyed by each traffic flow is dependent on the behavior of the other traffic flows with which it is aggregated. When combined with the graceful degradation capabilities of IP, DiffServ offers a range of QoS dependent on the network traffic rather than a specific set of guarantees. DiffServ-enabled routers handle traffic aggregates according to per-hop behaviors (PHBs). Several PHPs have been defined, which includes most importantly

- expedited forwarding (EF), which has a goal of providing minimal queuing delay,
- assured forwarding (AF), which has the goal of providing an expected level of throughput, and
- best effort (BE), equivalent to the current Internet service.

E. Path Setup and Selection

An LSP consists of a set of links from the source label switching router (LSR) to the destination LSR. On-demand LSP creation occurs as each traffic flow arrives at the network. An LSP is created specifically to meet the requirements of each flow. Pre-computed LSP creation occurs periodically as network QoS state is updated through a signaling protocol. Traffic flows with similar QoS requirements, to the same destination, are usually routed over the same pre-computed LSP. This aggregation concept is consistent with the DiffServ protocol. Static LSP creation occurs periodically and includes traffic flow aggregation, but is based on the network physical state (adding or subtracting links and nodes) [4]. On-demand LSP creation requires the most time per traffic flow, and static LSP creation the least.

Networks can employ a wide variety of QoS path selection strategies. Previous comparative studies have demonstrated that algorithms that prioritize using minimum-hop routes almost always outperform algorithms that prioritize balancing the network load. [5] Minimum-hop algorithms like widest shortest path [6] tend to reduce the consumption of network resources, minimizing the effect of current traffic on future traffic. Use of other than minimum-hop paths tends to consume network resources that might be needed for future traffic.

F. Network State

Both the on-demand and pre-computation LSP creation techniques require knowledge of the global network QoS state. Ideally, this global network QoS state will be accurate at every node that requires the knowledge. This is a very difficult task to accomplish in a network with a lot of traffic from mobile nodes. Continuously updating the global network QoS state is very expensive in terms of network bandwidth and time. Periodic updates can be used, at a cost in accuracy of the routing decision, as the period between updates gets longer [7]. Static LSP creation requires only network physical state. This information is readily available at each node from the IP routing information. Thus, neither special signaling nor global QoS state information is required.

G. Related Frameworks

Nelakuditi and Zhang propose and evaluate a framework both to select a number of possible paths between a source and a destination node (a set of candidate paths) and to pick paths out of this set for new flows in [8]. A widest disjoint path (WDP) algorithm is used to calculate the set of candidate paths with the maximum width. The width of a set reflects the total amount of bandwidth that can be transmitted over this set of paths. If a path does not increase the width because it shares a bottleneck with another path in the set, it is pruned. The latter calculations are based on global state information that is distributed in sparse intervals. Then, a path for a new incoming flow is picked such that all paths in the set have equal blocking probabilities. This decision is based on information locally available at the node. This scheme is compared to best path routing based on a widest shortest path algorithm. Best path routing requires global knowledge of the network state. Since the update interval has to be reasonably long to keep the signaling load on the network small, best path routing often operates on stale information. This leads to synchronization problems where multiple nodes simultaneously congest similar paths because their stale information indicates spare capacity. The results show that the widest disjoint path/equal blocking probabilities scheme outperforms best path routing while minimizing the signaling overhead.

In [9], we proposed a connection admission control and flow-based traffic engineering framework for small DiffServ domains based on path queue states. The proposed algorithm renders its decision based on path queue state information gathered by edge routers. Each edge router gathers information on the states of the queues on all paths to each peer edge router it has. Then, the expected QoS properties for each path are computed. Edge routers render admission decisions based on this information and pick a suitable path for newly admitted traffic flows.

III. RESEARCH OVERVIEW

The focus of this research is to develop and evaluate a router-based traffic engineering architecture for use in a RAN. The basic premise of our architecture is that each radio access router (RAS) has a set of shortest and disjoint LSPs set up between itself, the Internet gateway, and every other RAS within the RAN. These static LSPs are set up ahead of network operations, either through a self-discovery process using the IP routing information or a centralized download. Disjoint paths in the path set help ensure that each set of paths has routes that avoid congestion at any one point in the network. The LSPs have no reserved bandwidth. This architecture scales on the order of N^2 where N is the number of nodes. Therefore it is not suitable for large networks but is quite well suited for small to medium sized networks (10 to 50 nodes) such as RANs.

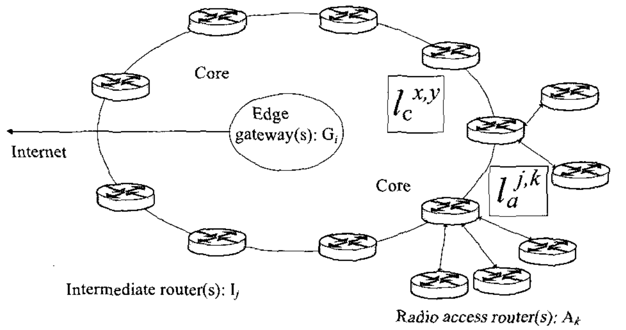


Fig. 1. Generalized radio access network.

The RAS and/or edge gateway router (EGW) respond to a QoS traffic flow request by determining if adequate bandwidth is available at the local link to support the request. Next, using a widest shortest path algorithm [6], the RAS/EGW chooses a path (LSP) from the set of paths to the destination. It then forwards the traffic flow along the path. For best effort traffic, the RAS places the traffic on a path to the destination in round-robin fashion.

Each RAS/EGW must have a method for calculating the amount of RAN core network bandwidth that it is entitled to manage. The widest shortest path algorithm assumes that the executing node has accurate knowledge of the state of the links being managed. Because we do not distribute global QoS state, each RAS/EGW has knowledge only of its own use of RAN core bandwidth, not of the total bandwidth usage itself. Therefore, each RAS/EGW must be able to determine some share of the RAN core bandwidth that it then manages using the widest shortest path algorithm.

In our network model, we consider mobile IP related issues that directly affect traffic engineering algorithms. We use mobile traffic sources and we evaluate the effects of mobility on the amounts of traffic presented to the RAN. We do not simulate or consider issues that do not directly affect traffic engineering. For example, we do not simulate signaling for roaming nodes and the changes in delay and jitter due to handoffs as we do not believe they significantly affect our proof of concept evaluation, which we have conducted.

A. Algorithm

We propose a new algorithm that we call the local state fair share bandwidth (LSFSB) algorithm. LSFSB uses a proportional allocation of the RAN core bandwidth, allowing each RAS/EGW to continuously determine its fair share of the RAN core network bandwidth without external signaling. The proportional core bandwidth allocation is primarily determined by the RAS's individual connectivity into the RAN. The algorithm is also a function of the destination of the traffic in the RAN (local or external), and the bandwidth of the various links that make up the RAN.

The generalized RAN as seen in Fig. 1 consists of a core IP network with RASs on the edges. All routers are MPLS, Diff-

Serv, and HMIP enabled. The mobility anchor point is located at an EGW. These are our assumptions and observations about the generalized RAN:

- The network structure is hierarchical and includes alternate paths in the core to enable traffic engineering.
- RASs (A_k) are each connected to only one Intermediate router (I_j).
- There is at least one EGW (G_i) that is also the HMIP mobility anchor point.
- The bandwidth of bi-directional RAN core links between edge points x, y is $l_{\mathrm{c}}^{x,y}$.
- The bandwidth of bi-directional access links between each A_k and an I_j is $l_{\mathrm{a}}^{j,k}$.
- There is traffic on the network, some fraction g_{T}^{k} of which passes from RAS (A_k) through any EGW (G_i) to the Internet. Each RAS A_k will calculate its own g_{T}^{k} based on its local traffic statistics.

B. Radio Access Router LSFSB Algorithm

We propose the following set of calculations as part of the LSFSB algorithm to enable each RAS (A_k) to continuously calculate its fair share of the RAN core bandwidth:

B.1 Stage One

Let β_{c} represent the total transmission bandwidth available in the generalized RAN core, such that

$$\beta_{\mathrm{c}} = \sum_{x,y} l_{\mathrm{c}}^{x,y}. \tag{1}$$

Define a gateway factor Γ to split β_{c} between the A_k and G_i as a function of g_{T}^{k}

$$\Gamma = 1 - (0.5 \cdot g_{\mathrm{T}}^{k}). \tag{2}$$

The use of the multiplier 0.5 in (2) assumes two-way traffic flows of equal magnitude. This function causes a RAS to assume less core bandwidth if it is handling more EGW traffic. This in turn causes the RAS to switch to using alternate paths faster when sending more traffic towards the EGW, a congestion point in the RAN. The idea is to delay congesting EGW links as long as possible as traffic density builds at a specific RAS.

Let β_{a} represent the core transmission bandwidth available for all A_k. Then

$$\beta_{\mathrm{a}} = \Gamma \beta_{\mathrm{c}}. \tag{3}$$

Next, define an access factor α_{a}^{k} to split βa between all A_k. For any $\mathrm{A}_{k=m}$ connected to $\mathrm{I}_{j=n}$

$$\alpha_{\mathrm{a}}^{m} = \frac{\sum_{y \cap n} l_{\mathrm{c}}^{n,y}}{\sum_{y}^{j} l_{\mathrm{c}}^{j,y}} \cdot \frac{l_{\mathrm{a}}^{m,n}}{\sum_{k \cap n} l_{\mathrm{a}}^{k,n}}. \tag{4}$$

The access factor α_{a}^{k} is a function of the intermediate router's (I_n) relative connectivity into the core compared with other intermediate routers (first part of (4)) and the relative connectivity of the RAS (A_m) compared with other access routers off the same intermediate router (second part of (4)).

Let β_{a}^{k} represent the core bandwidth allocated to A_k. Then

$$\beta_{\mathrm{a}}^{k} = \alpha_{\mathrm{a}}^{k} \beta_{\mathrm{a}}. \tag{5}$$

Finally, each A_k allocates β_{a}^{k} to the links $l_{\mathrm{c}}^{x,y}$ that connect its I_j to the core proportionally to the bandwidth of those links. Then for any $\mathrm{A}_{k=m}$ connected to $\mathrm{I}_{j=n}$ and link $l_{\mathrm{c}}^{n,y}$

$$\beta_{\mathrm{a}}^{m(l_{\mathrm{c}}^{n,y})} = \beta_{\mathrm{a}}^{m} \frac{l_{\mathrm{c}}^{n,y}}{\sum_{y} l_{\mathrm{c}}^{n,y}}. \tag{6}$$

We assume that A_k will accept QoS traffic up to the limit of $l_{\mathrm{a}}^{j,k}$. If β_{a}^{k} is greater than or equal to $l_{\mathrm{a}}^{j,k}$ for A_k, this first stage of the LSFSB algorithm will allow A_k to put as much traffic into the RAN as its local link $l_{\mathrm{a}}^{j,k}$ can handle.

B.2 Stage Two

If β_{a}^{k} is less than $l_{\mathrm{a}}^{j,k}$ for A_k, we have two possible operating conditions. By operating only in the first stage of the algorithm and not admitting additional traffic, the RAN will not have more traffic than its core bandwidth can accommodate. However, we wish to be able to take advantage of the graceful degradation ability of DiffServ and IP. To accomplish this, an additional stage is required—an overload stage, which we propose as follows:

- Distribute incoming traffic across the possible paths in proportion to the total path distribution achieved under the first stage of the algorithm. Do not split flows.
- Track the first stage assignments and replace them as they depart to update the path distribution.

C. Edge Gateway LSFSB Algorithm

We propose a similar set of calculations to enable each G_i to continuously calculate its fair share of RAN core network bandwidth so that it can apply the widest shortest path algorithm. The main difference is that each G_i has a g_{T}^{i} fixed at 1.0, since each G_i only applies the LSFSB algorithm to incoming Internet traffic.

IV. SIMULATION

In order to evaluate our traffic engineering architecture and LSFSB algorithm, we constructed a simulation environment. This simulation environment allowed us to test our architecture and algorithm under a variety of different network operating conditions and traffic mobility scenarios.

A. Simulation Environment

The simulation environment is shown in Fig. 2. It is composed around an unmodified ns-2 version 2.1b6a simulator. To this unmodified version of ns-2, we added the MPLS Network Simulator version 1 [10], and a DiffServ system [11]. Our software only performed pre-processing and data collection tasks. This distinct separation allowed us to make as few changes to the ns-2 software as possible, which in turn freed us from the testing, troubleshooting, and verification that extensive modifications would have required. Our simulation software was

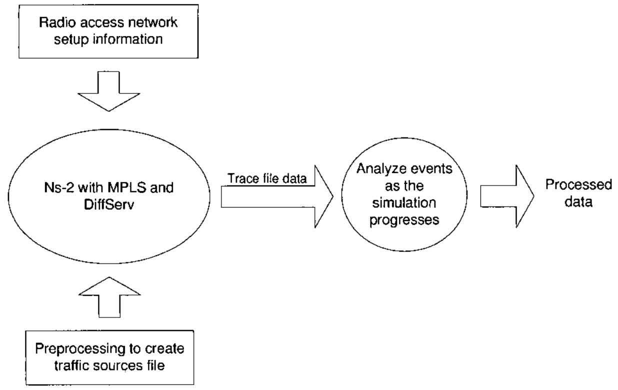


Fig. 2. Simulation environment.

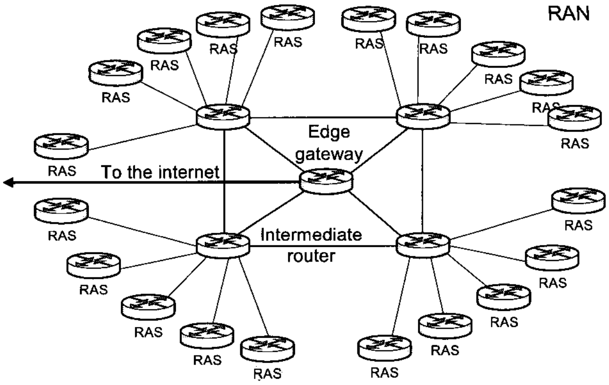


Fig. 3. Modeled radio access network.

designed to implement our traffic engineering environment and LSFSB algorithm.

B. Network Traffic

The ns-2 simulator uses traffic generators to create traffic in the simulation. These traffic generators are normally entered as a series of commands as the simulation is initialized. The traffic types we simulated are voice traffic using the EF PHB and the user datagram protocol (UDP) as its transport agent, video traffic using the EF and AF PHBs and UDP for transport, and data traffic using the transmission control protocol (TCP) and the AF and BE PHBs.

Although ns-2 contains radio-link simulation support, we chose not to use this support in our simulation environment. We were interested in evaluating our traffic engineering architecture and algorithm. Excluding the radio links from the simulation eliminated an unnecessary variable from our results. Instead, we simply attached, started, stopped, and detached traffic generators to simulate mobile node movement. This simplified approach provided enough realism as far as the traffic engineering architecture and algorithm were concerned. Our mobile node movement scenarios are summarized below:

- Mobile nodes may move in a uniformly random manner such that the traffic load remains roughly evenly distributed across the network. Initially, each node randomly chooses a point to move towards and then moves towards that point as the simulation progresses. We used a lognormal distribution to simulate cell dwell time [12].
- Mobile nodes may move in a linear manner such that the traffic load travels along the network as a group, roughly synchronized, towards the same destination. This places a point load on the network. We again used a lognormal distribution to simulate cell dwell time.

C. Simulating Traffic Engineering

We compared the results of using LSFSB and our traffic engineering architecture to two other algorithms for routing and path selection. The first was standard IP routing, which uses only the shortest path between any source and destination. The second employed our traffic engineering architecture, but instead of LSFSB used global network state combined with widest shortest path to select LSPs to carry traffic flows. We called this "global-WSP." The traffic engineering architectures, LSFSB and Global-WSP, are implemented in the final step of traffic file processing. Both these traffic engineering programs use a list of network paths and links to assign flows to LSPs and track link bandwidth usage.

All three routing/traffic engineering programs produce identical loads to present to ns-2 for simulation. The loads differ only in the way that they are directed across the RAN. This allowed us to directly compare the three different routing/traffic engineering methods. The two traffic engineering programs that implement LSFSB and global-WSP both employ the same traffic admissions policy, and produce identical output except for the LSPs selected to route the particular flows. The IP program takes the output of either traffic engineering program and removes the MPLS commands. This causes all packets to be routed using standard IP routing.

D. Simulation Data Processing

The data gathering during the simulation was done following a "warm-up" period for the RAN. During this warm-up period, lasting 30 seconds of simulation time, we instantiated all the LSPs and randomly started all of the traffic sources specified by the user to be started during this period. This warm-up period enabled us to fill the RAN with traffic and achieve a less-transient set of network operating conditions. Following the warm-up period, we collected network data for a period of four minutes of simulation time. We hesitate to characterize this measurement period as "steady-state" given the rapid and continual changes in the traffic source locations throughout our simulations. However, it does represent a relatively stable RAN operating condition. We arrived at the 30-second warm-up and four-minute data collection times through some experimentation.

E. Simulated Network

The RAN we modeled for the purposes of proof of concept is shown in Fig. 3. Each link in the modeled RAN has capacity 1.536 Mbps. We do not model the Internet; our simulations are confined to the interior of the RAN. The capability to traffic engineer in this model is provided by the links between the intermediate nodes. These links provide up to three paths between

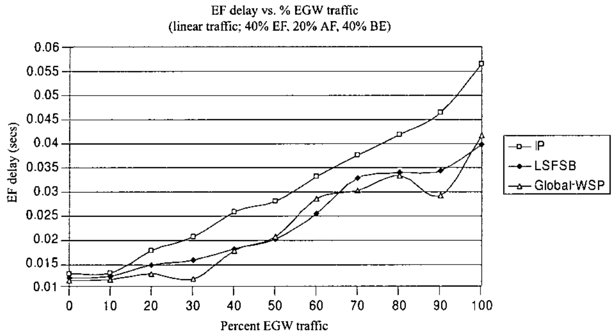


Fig. 4. EF delay for linear traffic scenario.

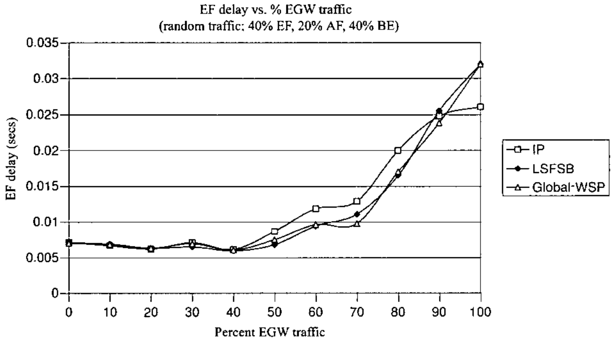


Fig. 5. EF Delay for random traffic scenario.

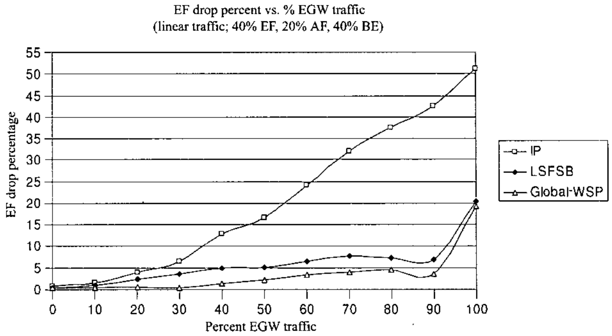


Fig. 6. EF drop percentage for linear traffic scenario.

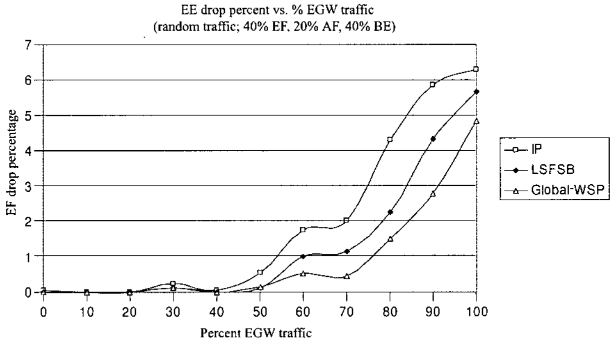


Fig. 7. EF drop percentage for random traffic scenario.

each RAS, and each RAS and the EGW. In our simulations, we used 1100 LSPs configured statically. In addition, we used the differentiated services code point for DiffServ classification and mixed all three classes of differentiated services (DiffServ) traffic on the LSPs.

V. SIMULATION RESULTS

A selection of our simulation results is shown in Figs. 4–11. Other results are available in [13]. We present this data to show that our traffic engineering architecture and algorithm is better than standard IP routing for point loads on a RAN, and not measurably better than IP routing for uniformly distributed loads. It is important to keep in mind that we performed these simulations under increasingly heavily loaded network conditions. One would not normally aspire to run a network on a continuous basis at most of these loading levels. The quality of service that the heavy loads produce in terms of packet loss, delay, and throughput would not be acceptable to most network operators and users. However, a primary reason for traffic engineering is to better cope with heavy network loading when compared to IP routing. Indeed, if a network is always lightly loaded compared to its total capacity, there would be no need for traffic engineering.

We show two sets of data for EF traffic using two different traffic loading scenarios. These two different load distribution and movement scenarios, linear and random, were described in Section IV-B. Because the goal of EF is to provide minimal delay, Fig. 4 (linear traffic scenario) and Fig. 5 (random traffic scenario) show the average EF packet delay as the percentage of traffic transiting the EGW was varied from zero to 100 percent. The overall traffic mix in the RAN was 40% EF, 20% AF, and 40% BE as noted in the figures.

The effect of increasing the percentage of traffic transiting the EGW was to increase the congestion within the RAN. As the amount traffic that is headed towards a single point in the RAN (EGW) increases, congestion on certain RAN links increased. Another important measure of the effectiveness of our algorithm is the percentage of packet loss. We include in Figs. 6 and 7 the packet loss comparison of the three traffic routing systems. This data was gathered during the same set of simulations that produced the data in Figs. 4 and 5. The data shows clearly that the two traffic engineering systems, LSFSB and Global-WSP, gave much better EF traffic performance than standard IP routing in the linear (point load) scenario. Under the random (uniform load) scenario, there was no statistically significant difference between the three traffic routing algorithms.

In Figs. 8–11, we show two sets of data for AF traffic—the packet loss as a percent of total traffic, and the throughput of AF traffic as a comparison between global-WSP, LSFSB and standard IP routing. The goal of AF quality of service is throughput. Our AF results were similar to the EF results, in that LSFSB was effective in a linear (point load) traffic scenario and the same as standard IP routing in the random (uniform load) traffic scenario. In addition, there was no statistically significant difference between the performance of LSFSB and global-WSP in the AF results.

We collected data on BE traffic in our simulated RAN for each scenario. Although we do not present that data here, it shows that BE performance under LSFSB is at least as good as

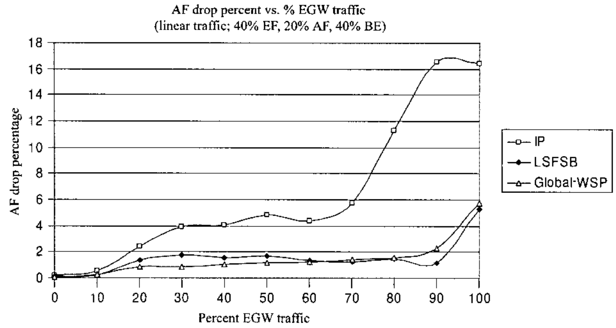


Fig. 8. AF drop percent for linear traffic scenario.

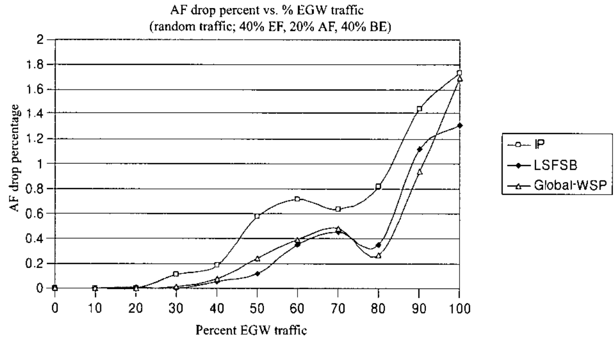


Fig. 9. AF drop percent for random traffic scenario.

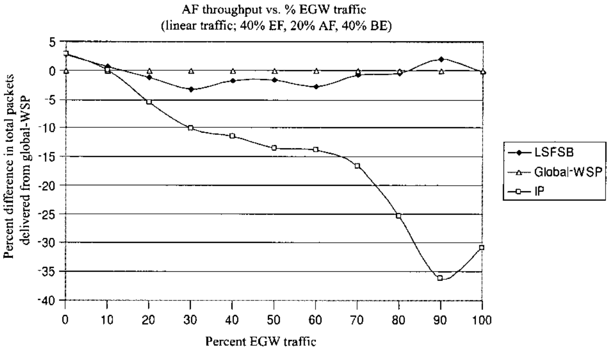


Fig. 10. AF throughput comparison for linear traffic scenario.

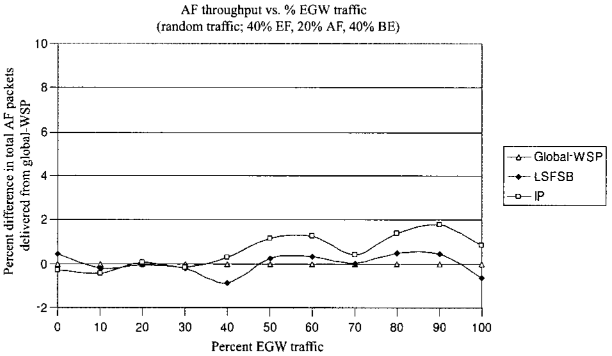


Fig. 11. AF throughput comparison for random traffic scenario.

with standard IP routing. This result is important since quality of service traffic must not starve best effort traffic in a realistic RAN.

VI. CONCLUSIONS

We set out to examine how well one could implement traffic engineering in a RAN without using a complex traffic engineering algorithm. RANs are relatively small since they are a metropolitan area type network. The equipment used to implement this type of network should not be very complicated, thus a complex traffic engineering algorithm is not desirable. Using a traffic engineering algorithm that has total global knowledge through a signaling protocol is the best solution from a performance standpoint. This is true because an all knowing algorithm can always optimize to the present situation. The complexity involved in this approach is undesirably high for a RAN. Alternatively the simplest approach is to over provision the network as is typically done today. We set out to see what was possible in the solution space between these two extreme alternatives.

Our hypothesis was that using knowledge gained locally at each router in the RAN, we thought we could do better than no traffic engineering but worse than full global knowledge traffic engineering. Our results agreed with our initial hypothesis. We showed that LSFSB can be effective as a traffic engineering architecture and algorithm. Our simulations showed that LSFSB improved QoS traffic packet loss, delay, and throughput in a RAN with a point load and performed no worse than standard IP routing in a RAN with a uniform load. This was an expected result, since networks with large uniformly distributed loads have no place to re-route traffic for improved performance. We also showed that LSFSB, a relatively simple algorithm, could perform nearly as well as a much more difficult to achieve global state algorithm, Global-WSP. This result supports the possible use of LSFSB in RANs where complex traffic engineering solutions are impractical and undesirable.

REFERENCES

[1] C. Perkins, "IP mobility support," *IETF RFC 2002*, Oct. 1996.

[2] H. Soliman, C. Castellucia, K. El-Malki, and L. Bellier, "Hierarchical MIPv6 mobility management," *IETF Internet Draft*, Sept. 2000, work in progress.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," *IETF RFC 2475*, Dec. 1998.

[4] G. Apostolopoulos and S. Tripathi, "On the effectiveness of path precomputation in reducing the processing cost of on-demand QoS path computation," in *Proc. IEEE Symp. Computers Commun.*, 1998, pp. 42–46.

[5] Q. Ma and P. Steenkiste, "On path selection for traffic with bandwidth guarantees," in *Proc. IEEE Int. Conf. Network Protocols*, 1997, pp. 191–202.

[6] R. Guerin, A. Orda, and D. Williams, "QoS routing mechanisms and OSPF extensions," in *Proc. IEEE GLOBECOM'97*, vol. 3, 1997, pp. 1903–1908.

[7] A. Shaikh, J. Rexford, and K. G. Shin, "Evaluating the impact of stale link state on quality of service routing," *IEEE/ACM Trans. Networking*, vol. 19, no. 2, pp. 162–176, Apr. 2001.

[8] S. Nelakuditi and Z.-L. Zhang, "A localized adaptive proportioning approach to QoS routing," *IEEE Commun. Mag.*, vol. 40, no.6 pp. 66–71, June 2002.

[9] S. Krasser, H. Owen, J. Grimminger, H.-P. Huth, and J. Sokol, "Online traffic engineering and connection admission control based on path queue states," in *Proc. IEEE SoutheastCon 2004*, Mar. 2004, pp. 255–260.

[10] G. Ahn and W. Chun, "Overview of MPLS network simulator: Design and implementation," Department of Computer Engineering, Chungnam National University, Korea, 2000.

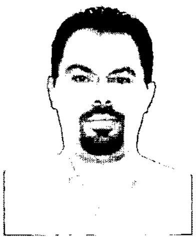
[11] S. Murphy, "The ns/MPLS/DiffServ patch," Dublin City University, Ireland, 2000.

[12] C. Jedrzycki and V. Leung, "Probability distribution of channel holding

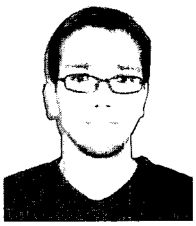
time in cellular telephony systems," in *Proc. IEEE VTC'96*, vol. 1, 1996, pp. 247–251.

[13] D. Barlow, "Router-based traffic engineering in MPLS/DiffServ/HMIP radio access networks," Ph.D. dissertation, Georgia Institute of Technology, Atlanta, Georgia, USA, Apr. 2002.

David A. Barlow is an Assistant Professor in the Department of Electrical Engineering and Computer Science, United States Military Academy, West Point, New York. His current research interests include mechatronics and networking robotic systems. He received his Ph.D. from the Georgia Institute of Technology, Atlanta, Georgia, and M.S. and B.S. degrees from Worcester Polytechnic Institute, Worcester, Massachusetts.



Vasos Vassiliou holds a Ph.D. from Georgia Institute of Technology, Atlanta, Georgia, USA in the subject of Telecommunications and Mobile Network protocols. Since January 2004, he is a visiting lecturer at the Computer Science Department of the University of Cyprus. Prior to that he has been with the Computer Science Department of Intercollege, Cyprus. He is a graduate of the Higher Technical Institute (HND 1993), the University of South Florida (B.Sc. in Electrical Engineering, 1997), and the Georgia Institute of Technology (M.Sc., 1999 and Ph.D., 2002 in Electrical and Computer Engineering with Minor in Computer Science). He has held internship positions with BellSouth Mobility DCS and Nortel Networks during his graduate studies. He is a member of IEEE, IEE, ACM, and the Cyprus Technical Chamber. He has co-authored a book chapter and several papers in international conferences. His research interests focus on high speed network architectures (MPLS), mobile computing (mobile IP, ad-hoc networks), wireless communications (protocol enhancements for 3G/4G cellular wireless networks), and traffic engineering for computer networks.

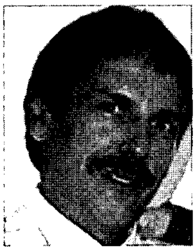


Sven Krasser received his Vordiplom in Electrical Engineering from the Technische Universität Carolo-Wilhelmina zu Braunschweig in Braunschweig, Germany in 2000. In 2001, he joined the School of Electrical and Computer Engineering at the Georgia Institute of Technology in Atlanta, Georgia from which he received a M.S. in 2002 and a Ph.D. in 2004. Currently, he works as a postdoctoral fellow at the Georgia Institute of Technology. His research interests include network security, communication network performance analysis, and traffic engineering.

Henry L. Owen is a Professor in the School of Electrical and Computer Engineering. He received his BSEE, MSEE, and Ph.D. in Electrical Engineering from the Georgia Institute of Technology in 1980, 1983, and 1989, respectively. He joined the faculty of the Georgia Tech Research Institute in 1980 as a Research Engineer I and was promoted to Research Engineer II in 1984. After completing his Ph.D. in 1989, he joined the academic faculty as an Assistant Professor. In 1990, he took a one-year leave of absence to conduct on site research for Alcatel in Stuttgart, Germany. In 2000, he was involved in an Internet security start up on a half-time basis. Dr. Owen's research interests include Internet security and internetworking.

Jochen Grimminger received the diploma (Dipl.-Phys. Univ.) degree as a Physicist at the Ludwig-Maximilian-University of Munich, Germany in 1997. He started to work then as a leading developing engineer at former Siemens OEN after inventing the SDX platform now with Juniper-Networks. In 1999, he joint Siemens Corporate Technology focusing on communication architectures related to e.g., 3GPP, IETF, IEEE802, IP-based-GSM/UMTS/4G providing intellectual properties, consulting, research, and prototyping for various aspects regarding IP-mobility, QoS, IPv6, and GMPLS. Recent research activities cover mainly technology scouting, Internet evolution, service and process oriented architectures (SoA, PoA) and web services. He is involved in Internet standardization and has published several IETF-drafts and papers.

Hans-Peter Huth received the diploma (Dipl.-Ing. Univ.) degree as an Electrical Engineer at the University of Munich, Germany in 1989. Since then he worked as a research engineer at Siemens Corporate Technology with various topics related to e.g., ATM, Ethernet, WLAN, and IP-based GSM/UMTS. He is currently heading a team doing consulting, research, and prototyping for various aspects regarding IP layer mobility, quality of service, policy control, IPv6, MPLS, and network monitoring. Recent research activities cover self-organizing access network architectures for WLAN hotspots, peer-to-peer style network monitoring, network support for adaptive applications, and ad hoc networks. He was involved in Internet standardization and has published several papers.



Joachim Sokol holds a diploma in Computer Science from the Fachhochschule Ulm, Germany. End of 1984, he joined Siemens. Since 1993, he is working as a project leader for Siemens Corporate Technology in several internal, national, and international projects with the research line focusing on quality of service in IP and ATM-based networks. Numerous scientific publications were published in these research areas. From 2000 to 2001, he was heading the "Network and Services Application" group at the International Computer Science Institute in Berkeley, USA. His current work focuses on content delivery networks, IP-based wireless networks, and their applicability and impacts to multimedia applications.