# Multihoming Route Control among a Group of Multihomed Stub Networks

Yong Liu, A. L. Narasimha Reddy

Department of Electrical and Computer Engineering
Texas A&M University
College Station, TX 77843
{yongliu,reddy}@ece.tamu.edu

**Abstract.** In recent years, commercial "Multihoming Route Control" devices are used by multihomed stub networks to optimize the routing of their Internet traffic. Previous studies have shown that "Multihoming Route Control" can improve the Internet communication performance of the multihomed stub networks significantly. In this work, we study "Multihoming Route Control" among a group of multihomed stub networks that may belong to an organization and exchange large amount of data regularly. Simulations show that greedy "Multihoming Routing Control" schemes could lead to oscillations. We propose a user-optimal routing based "Multihoming Routing Control" scheme that improves routing performance without causing oscillations. The proposed scheme is simpler to implement than optimal routing based approaches. We show through simulations that the proposed scheme achieves performance close to optimal routing and works well in various network conditions.

## 1 Introduction

Multihoming [1] has been traditionally used by stub networks for improving network availability. In recent years, Multihoming Route Control (MRC) technology [2] has been employed by multihomed stub networks to improve their Internet access performance. MRC devices choose the best ISP for Internet traffic of stub networks according to measured qualities of alternate paths via different ISPs. Measurement based analysis of the benefit of multihoming [3] shows that MRC may improve Internet access performance significantly for both enterprises and large data centers.

In this paper, we study the MRC of traffic among a group of multihomed stub networks. An example of such a group of stub networks is the networks of branches of an enterprise. The branch networks may multihome and exchange considerable traffic regularly among themselves. We call the traffic among such a group of stub networks as *inhouse* traffic and call the traffic between a stub network and networks not in the group

as *Internet* traffic. While our approach is mainly designed for MRC of in-house traffic, it can also be applied to Internet traffic to a small number of "top" Internet destination networks that account for a large portion of the total Internet traffic. In this work, we assume there are a few top Internet destinations for each stub network and we use the same MRC algorithm for these Internet destinations.

When the access links of the stub networks are not over-provisioned, traditional MRC schemes that use a single path at a time may cause oscillations as we sill show in Section 3. A global optimal routing based coordination method can be employed to avoid such possible oscillations [4]. In this work, we propose a distributed user-optimal routing based MRC scheme to solve the above problem. The basic idea is to use multiple paths provided by multihoming simultaneously and move traffic gradually during network environment changes. Specifically, our scheme calculates "user-optimal routing" [5] using the gradient projection method that is originally used in solving optimal routing problems. Calculating user optimal routing is simpler to implement and can achieve similar performance as global optimal routing for this problem as we will show in Section 5.

The rest of this paper is organized as follows: In Section 2, we give the network model of the MRC problem. In Section 3, we give an example of possible oscillations caused by greedy MRC. Our MRC scheme is introduced in Section 4. In Section 5, we compare the performance of our scheme with the optimal solution and show the dynamic characterization of our scheme using simulations. Conclusions are drew in Section 6.

## 2 Network model

In this work, we study the MRC of stub networks that employ BGP [6] based multihoming. We assume that a stub network advertises its IP address block(s) to all its ISPs because this approach achieves higher degree of survivability. In such a situation, the stub network can send outgoing traffic via either of its ISPs, but it cannot control which ISP the ingress traffic comes from. The route of ingress traffic is decided by the BGP relationships between ASes(Autonomous Systems) on the Internet. Therefore, the task of MRC for BGP based multihoming is to map egress traffic onto available paths provided through BGP based multihoming.

Existing MRC schemes work as follows [2]: the MRC device passes modified BGP and routing information to the BGP router of local network to force it to select a particular ISP for egress traffic to an IP address prefix. Because BGP uses single route for an address prefix at any time,

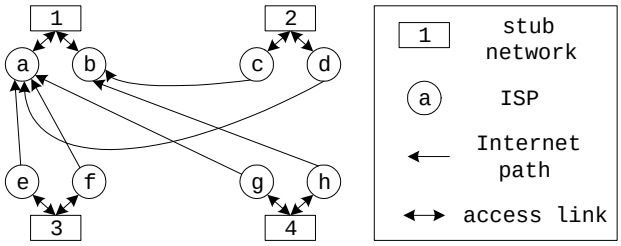



**Fig. 1.** Topology consisting of multihomed stub networks, the edge routers of their ISPs and the paths among them

MRC is also restricted to use a single route. It is essential for ISP networks to use single path for routing of inter-domain traffic in order to make BGP scale. However, for stub networks, it is not a problem to use multiple paths for egress traffic. Therefore, in this work, we assume stub networks can use multiple paths for egress traffic. To deploy our approach, multiple routes need to be assigned for interested destination networks and the forwarding engines uses our algorithm to decide the fraction of traffic to send on each route. Hashing based traffic splitting methods, like [7], or reordering robust TCP, such as [8, 9], can be used to avoid the packet reordering caused by multi-path routing.

We assume backbone links have large capacity compared to the access links of the stub networks. Thus, the route control of the stub networks does not significantly change the total load level on backbone links. Therefore, we can abstract a network path between the ISP edge routers of two stub networks as a virtual directed link with a given quality that may change overtime. We also abstract each stub network as a node. Therefore, a network we are studying consists of: (1) nodes representing stub networks in the group; (2) nodes representing ISP edge routers of the stub networks; (3) nodes representing a number of top Internet destinations of the stub networks; (4) links representing access links of the stub networks; (5) virtual directed links between ISP edge routers of different stub networks ; (6) virtual directed links between ISP edge routers of the stub networks and the Internet destinations. Fig. 1 shows the partial topology of a "4x2" network ("AxB" means the topology has $A$ stub networks and each stub network has $B$ ISPs) . The ISPs of different stub networks are different. For clarity, we only draw paths from stub network 2, 3, and 4 to stub network 1. Other paths among the stub networks and paths to and from Internet destinations are ignored.

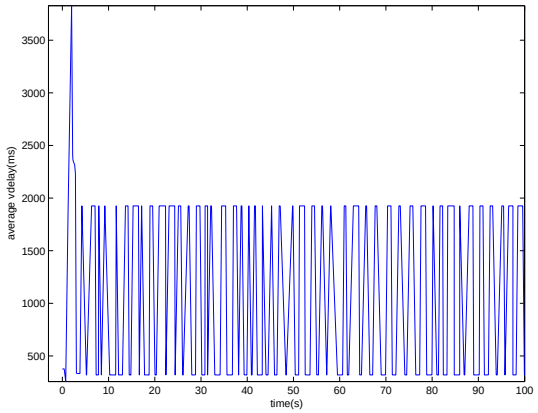



**Fig. 2.** Possible oscillations of greedy MRC

## 3 Possible oscillations of greedy MRC

In this section, we use simulations to show the possible oscillations of "greedy" MRC schemes for routing among a group of multihomed stub networks. The simulator, topology and traffic matrix generation methods are described in Section 5.2 where the dynamic characterization of our scheme is evaluated. However, here, we scale the traffic matrix to make the maximum link utilization 95% assuming traffic are equally split among alternate paths.

The greedy MRC scheme we illustrated here works as follows: (1) the gateway of a stub network keeps measuring the one way delay and loss rate of alternate paths to other stub networks and a few top Internet destinations; (2) the gateway decides whether to change the routing path for traffic to a destination after a random interval that is uniformly distributed between 0.5 second to 1.5 second; (3) the gateway changes the path for a destination when the "virtual delay" of currently used path is 20% larger than the minimum of alternate paths and is larger than 40 milliseconds (we define "virtual delay" as a metric of overall path quality, see Section 4.1 for details).

We randomly assign a path for traffic to a destination at the beginning of simulation. The result for a "8x3" topology is shown in Fig. 2. We can see the average virtual delay of all the traffic in the network keeps fluctuating that reflects the oscillations of the "greedy" MRC scheme. Similar oscillations are easily observed for simulations with different random seeds. These observations motivated our approach.

## 4 User-optimal routing based MRC

"User-optimal routing" [5] is optimal from the point of view of each user. It is also called "selfish routing". Like previous work on "user-optimal routing" [10], we assume traffic consists of a lot of "infinitesimal flows" and each user controls such an infinitesimal flow. At equilibrium of user-optimal routing, each flow is routed along a path with minimum end to end delay. Thus no user can reduce the delay of its traffic by changing the routing of its own traffic unilaterally. Previous work [10] showed that "selfish routing" can achieve similar performance as "optimal routing" [11] for intra-domain routing. As we will show in Section 5, for MRC, the performance of user-optimal routing based approach is also close to optimal routing based approach. One of the advantages of user-optimal routing is that it is distributed in nature and easier to implement.

In this section, we first introduce the concept of optimal routing and the cost function we used in this work; then we give the algorithm of our scheme.

### 4.1 Optimal routing and cost function

In a general optimal routing problem [11], routing traffic on a link incurs some cost and the optimal solution minimizes the overall cost. A commonly used cost function is the delay on a link weighted by the traffic volume on the link. Thus, the objective is to minimize the average delay of all traffic. It is assumed that the queue length of each link is infinite and there is a solution satisfying the capacity constraints.

In real networks, the queue length is limited; when load approaches or exceeds link capacity, there are packet losses. To take packet losses into account, we replace the delay function with a *virtual delay function*, which is a function of delay and loss rate on that link. Specifically, we use half of the expected TCP handshake round trip time [12], as the virtual one way delay function, i.e. $delay + T_s \frac{loss\_rate}{1-2loss\_rate} (loss\_rate < 0.5)$, where $T_s$ is the TCP SYN timeout, initially three seconds [13]. TCP handshake round trip time is used in comparing qualities of alternate paths by previous studies on Multihoming, e.g. [3]. Using this virtual function, the goal of optimal routing is to avoid congestion(packet losses) and to minimize the delay of traffic.

### 4.2 Measurement based user-optimal routing algorithm

Because of the similarity of the characterization between optimal routing and user-optimal routing, user-optimal routing can be calculated using

optimal routing algorithms with a specific cost function [5, 14]. The specific cost function of a link is the integral of the delay function of the link. An important class of algorithms for solving optimal routing problems are the gradient projection methods [11] that are also suitable for distributed implementation.

Our MRC scheme uses a measurement based distributed asynchronous gradient projection algorithm for calculating the user-optimal routing solution. The algorithm for each node is shown as Algorithm 1. The following notation is used in the algorithm: (1) $N$ is the set of stub networks; (2) $M_n$ is the set of Internet destinations of $n \in N$; (3) $P_{ij}$ is the set of valid paths from $i$ to $j$, where $i, j \in N, i \neq j$ or the set of valid paths from $i \in N$ to its Internet destination $j \in M_i$; (4) $l_{ijw}$ is the end to end virtual delay of path $w \in P_{ij}$, where $i, j \in N, i \neq j$ or $i \in N, j \in M_i$; (5) $x_{ijw}$ is the fraction of traffic from $i$ to $j$ routed along path $w \in P_{ij}$, where $i, j \in N, i \neq j$ or $i \in N, j \in M_i$.

**Algorithm 1** User-optimal routing based MRC (for traffic from $i$ to $j$, where $i \in N$; $j \in N, j \neq i$ or $j \in M_i$.)

**loop**
  measure all one-way end to end virtual delays, $l_{ijw} : w \in P_{ij}$
  find the minimum virtual delay to $j$, $l_{ij\bar{w}}$
  **for all** $w \in P_{ij}, w \neq \bar{w}$ **do**
    $x_{ijw}^{t+1} = max\{0, \quad x_{ijw}^{t} - \alpha^t (l_{ijw} - l_{ij\bar{w}})/l_{ijw}\}$
  **end for**
  $x_{ij\bar{w}}^{t+1} = 1 - \sum_{w \in P_{ij}, w \neq \bar{w}} x_{ijw}$
  wait for random time $T \in [0.5T_0, 1.5T_0)$;
**end loop**

Our scheme works as follows:

(1) The gateway of a stub network keeps measuring the end to end delay and packet loss rate of alternate paths from it to remote stub networks and some top Internet destinations. The virtual delay is then calculated according to the delay and loss rate. Methods like ICMP timestamp request can be used to measure the one way delay and loss rate. To filter measurement noise, the delay and loss rate is calculated from a number of samples. We leave the detailed measurement protocol and filtering method as future work and assume the measurement is accurate in this paper. Because our algorithm is based on the difference of delays of alternate paths, it does not require clock synchronization of different nodes.

(2) Every $t$ seconds, the gateway updates the routing vector $x$ for each destination according to the gradient projection algorithm. $t$ is uniformly distributed in $[0.5T_0, 1.5T_0]$ to avoid update synchronization, where $T_0$ is a constant that we set to 1 in this work. The gateway first calculates a new routing vector along the steepest descent direction with regard to the specific overall cost function (this step is mathematically expressed as $x^t_{ijw} - \alpha^t(l_{ijw} - l_{ij\bar{w}})/l_{ijw}$); then projects the routing vectors onto the non-negative range (this step is mathematically expressed as $max\{0, \dots\}$). $\alpha^t$ is the step size, for distributed implementation, it is usually a constant. Previous work [15] shows that the algorithm converges given $\alpha$ is small enough. In this work, we experimentally choose a constant $\alpha$ that works for all the simulations.

In this work, we assume that the traffic demand and the path qualities do not change too frequently compared to convergence speed of the algorithm.

## 5 Evaluation

We evaluate our scheme using simulations. The evaluation consists of two parts: (1) Since our scheme is based on user-optimal routing, it is important to ensure it will not cause network wide performance degradation. We perform a number of simulations to compare the performance of user-optimal routing based MRC with the optimal solution. (2) We study the dynamic behavior of our algorithm in various dynamic network environments.

We wrote a flow level simulator for the simulations. The delay and loss rate on each access link are calculated according to a queuing model. To get the delay and loss rate for different link utilizations, we use piecewise linear approximation models built from samples of ns-2 [16] simulations. The two models are Poisson queuing model, M/M/1, and a Pareto ($\beta = 1.5$) queuing model, P/M/1. The parameters of the ns-2 simulation are as follows: The average packet length is 558 bytes(calculated from a backbone trace); The link capacity is 100 Mbps; The buffer size of each link is equal to the product of 250 milliseconds and link speed.

We generate traffic matrices using a simplified version of the Gravity model [17]. To generate inhouse traffic between nodes (stub networks), we assign two uniformly distributed random numbers to each node $i$, $O_i, D_i \in [0, 1]$. Then the traffic demand from node $i$ to node $j$ is calculated as $\alpha O_i D_j$, where $\alpha$ is a parameter, $O_i$ and $D_i$ model how active node $i$ is as a sender and as a receiver. Similarly, to generate Internet traffic,

| | | Poisson | | | | Pareto | | | |
|---|---|---|---|---|---|---|---|---|---|
| opt obj($10^5$) | elb | 0.170 | 0.269 | 0.704 | 2.022 | 0.170 | 0.269 | 0.740 | 2.084 |
| | uopt | 0.158 | 0.252 | 0.482 | 1.391 | 0.158 | 0.252 | 0.503 | 1.446 |
| | gopt | 0.158 | 0.250 | 0.475 | 1.382 | 0.158 | 0.250 | 0.496 | 1.437 |
| loss rate(%) | elb | 0.000 | 0.000 | **0.908** | **2.912** | 0.000 | 0.000 | **0.985** | **3.012** |
| | uopt | 0.000 | 0.005 | **0.431** | **1.842** | 0.000 | 0.007 | **0.476** | **1.938** |
| | gopt | 0.000 | 0.000 | **0.414** | **1.822** | 0.000 | 0.000 | **0.458** | **1.918** |
| delay(ms) | elb | 23.94 | 23.94 | **24.00** | **24.03** | 23.94 | 23.94 | **24.00** | **24.03** |
| | uopt | 22.09 | 22.11 | **22.22** | **22.37** | 22.09 | 22.11 | **22.23** | **22.37** |
| | gopt | 22.09 | 22.10 | **22.21** | **22.36** | 22.09 | 22.11 | **22.23** | **22.38** |
| max util | elb | 0.600 | 0.950 | 1.100 | 1.250 | 0.600 | 0.950 | 1.100 | 1.250 |
| | uopt | 0.673 | 0.974 | 1.035 | 1.149 | 0.673 | 0.963 | 1.031 | 1.148 |
| | gopt | 0.673 | 0.966 | 1.032 | 1.147 | 0.673 | 0.946 | 1.025 | 1.147 |
| mean util | elb | 0.306 | 0.485 | 0.563 | 0.638 | 0.306 | 0.485 | 0.563 | 0.638 |
| | uopt | 0.306 | 0.485 | 0.563 | 0.638 | 0.306 | 0.485 | 0.563 | 0.638 |
| | gopt | 0.306 | 0.485 | 0.563 | 0.638 | 0.306 | 0.485 | 0.563 | 0.638 |

**Table 1.** Performance of optimal(gopt), user-optimal(uopt) and static load-balancing(elb) MRC for 8x2 topologies

we assign two uniformly distributed random numbers to each node $i$, $O'_i, D'_i \in [0, 1]$. The egress and ingress Internet traffic of the node are $\beta O'_i$ and $\beta D'_i$, where $\beta$ is a parameter, $O'_i$ and $D'_i$ model how active node $i$ is in sending and receiving Internet traffic. In our simulations we choose the $\alpha$ and $\beta$ to make the expected volume of inhouse traffic as 50% of the total traffic. The egress Internet traffic is randomly distributed to 5 Internet destinations(each has a random weight uniformly distributed in [0,1]). The ingress Internet traffic is randomly distributed on all ingress access links(each has a random weight uniformly distributed in [0,1]).

We generate a network topology in two steps: (1) We map stub networks onto some of 17 major cities of the United States; (2) We generate the path delay(i.e. propagation delay) of alternate paths by multiplying the measured one way delay between the two cities on AT&T backbone [18] with a random number uniformly distributed in [1, 1.6].

### 5.1 Performance compared to optimal routing

We calculate the global optimal routing("gopt"), user-optimal routing("uopt") and static load-balancing("elb") MRC solutions for randomly generated topologies and traffic matrices. The "static load-balancing" here is to split traffic evenly among all alternate paths. The topologies are of size "4x2", "4x3", "8x2" and "8x3". For each size, we generate 7 topologies and 5 traffic matrices. We scale the traffic matrices to make the maximum

link utilization 60%, 95%, 110% and 125% assuming basic load-balancing routing is used. Because the results for different sizes of topologies we simulated are similar, only data for topologies of 8x2 is presented, as shown in Table 1.

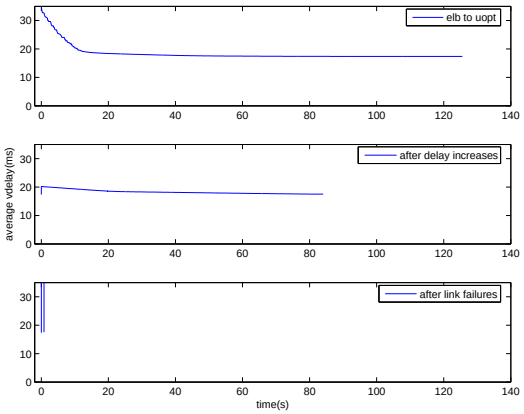
From Table 1, we have following observations: (1) Both optimal routing and user-optimal routing can reduce the overall routing cost significantly at higher link utilizations; (2) The performance of optimal routing and user-optimal routing are quite similar; (3) The two approaches reduce the overall cost by reducing the maximum link utilization and average loss rate, in other words, by avoiding congestion in the network; (4) The above observations hold for both Pareto queuing model and Poisson queuing model.

### 5.2 Dynamic behavior

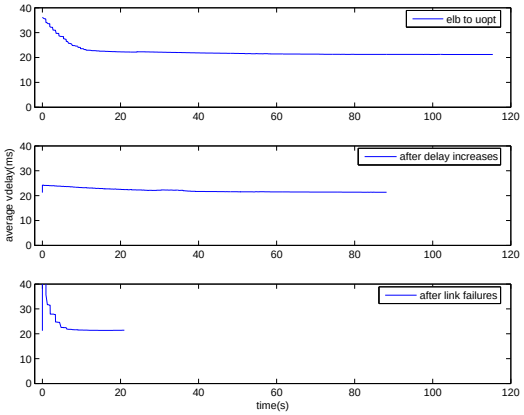
We study the convergence of our algorithm under following three types of scenarios: (1) We generate a random 8x3 topology and a random traffic matrix, assign initial routing allocation according to the "static load balancing" algorithm and let the algorithm converge to user-optimal equilibrium; (2) At the equilibrium point, we select 30% of paths among the stub networks and increase the delay of them by 50 ms and let the algorithm converge; (3) At the equilibrium point, we select 30% of paths among the stub networks and mark the path as disconnected and let the algorithm converge. Here, we define convergence as the state when the maximum difference of virtual delays of alternate paths used by traffic to one destination are not larger than 5 milliseconds. To converge to a state where the maximum difference is not larger than 1 millisecond or less, it takes more time but gets similar overall performance. In the simulations, the traffic matrix is scaled such that the maximum utilization for the "elb" routing approach is 110%.

Results of three sets of simulations are shown in Fig. 3. The different subfigures in each figure correspond to the three scenarios (1), (2) and (3) explained above. We can see that the algorithm converges fast to a near-equilibrium point, in a few seconds. The convergence time for the "link failure" scenario is shorter than other scenarios because the algorithm responses to large virtual delay difference faster. The link failures cause traffic to be switched to other paths immediately and cause high virtual delay on some other paths.

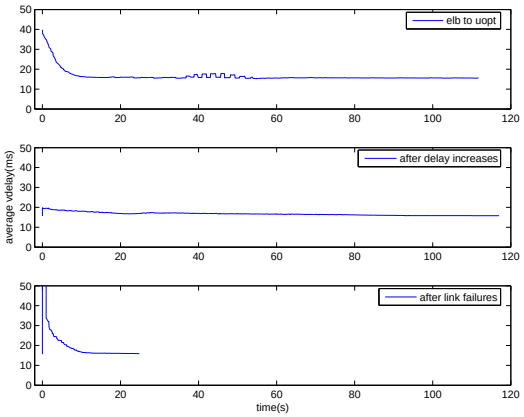
We also study the affect of different step sizes on the convergence of the algorithm. The convergence from "static load balancing" to "user-optimal equilibrium" of one previous simulation (as shown in Fig. 3(a))




(a) Simulation 1




(b) Simulation 2




(c) Simulation 3

**Fig. 3.** Convergence of user-optimal routing based MRC in different scenarios (step size = 0.02)

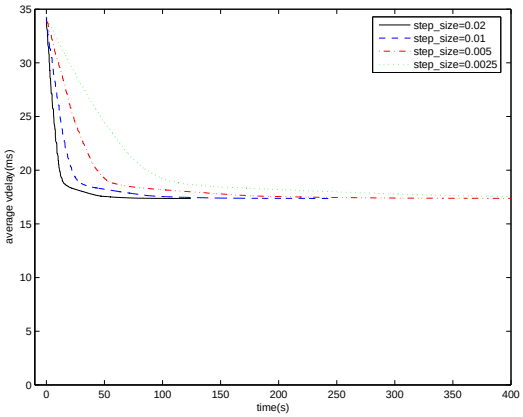



**Fig. 4.** Effect of different step sizes

is shown in Fig. 4. We can see that the algorithm converges quickly for several step sizes, converging faster with larger step sizes.

## 6 Conclusions

In this paper, we have studied Multihoming Route Control(MRC) among a group of multihomed stub networks. We showed that simple greedy MRC approaches could cause oscillations and degrade the performance of the data communication among the group of stub networks. We proposed a user-optimal routing based distributed Multihoming Route Control scheme that is simple to implement. We have shown through extensive simulations that the proposed MRC scheme improves performance in various network conditions without any oscillations. We have also shown that the user-optimal routing algorithm converges reasonably fast and achieves performance close to that of global optimal routing.

## References

1. Smith, P.: BGP multihoming techniques. NANOG 23 (2001)
2. Passmore, D.: Multihoming route optimizers. Bussiness Communications Review (2001)
3. Akella, A., Maggs, B., Seshan, S., Shaikh, A., Sitaraman, R.: A measurement-based analysis of multihoming. In: Proc. of ACM SIGCOMM. (2003) 353–364
4. Liu, Y., Reddy, A.L.N.: Route optimization among a group of multihomed stub networks. In: Proc. of IEEE GLOBECOM. (2005)
5. Wardrop, J.: Some theoretical aspects of road traffic research. In: Proc. of the Institute of Civil Engineers. Volume 1. (1952) 325–378

6. Rekhter, Y., Li, T.: A Border Gateway Protocol 4 (BGP-4). RFC 1771 (1995)
7. Sinha, S., Kandula, S., Katabi, D.: Harnessing TCPs Burstiness using Flowlet Switching. In: 3rd ACM SIGCOMM Workshop on Hot Topics in Networks. (2004)
8. Bhandarkar, S., Reddy, A.L.N.: TCP-DCR: Making TCP robust to noncongestion events. Proc. of Networking (2004)
9. M.Zhang, B. Karp, S.F., Peterson, L.: Reordering robust tcp with dsack. Proc. of IEEE ICNP (2003)
10. Qiu, L., Yang, Y.R., Zhang, Y., Shenker, S.: On selfish routing in Internet-like environments. In: Proc. of ACM SIGCOMM. (2003) 151–162
11. Bertsekas, D.P., Gallager, R.: Data Networks. 2nd edn. Prentice-Hall (1992)
12. Cardwell, N., Savage, S., Anderson, T.E.: Modeling TCP latency. In: Proc. of IEEE INFOCOM. (2000) 1742–1751
13. Braden, R.: Requirements for Internet hosts - communication layers. RFC 1122 (1989)
14. Florian, M., Hearn, D.: 6. Network equilibrium models and algorithms. In: Network Routing. Elsevier Science (1995)
15. Tsitsiklis, J.N., Bertsekas, D.P.: Distributed asynchronous optimal routing in data networks. Automatic Control, IEEE Transactions on **31**(4) (1986) 325 – 332
16. Network Simulator - ns2, http://www.isi.edu/nsnam/ns/.
17. Kowalski, J., Warfield, B.: Modeling traffic demand between nodes in a telecommunications network. In: ATNAC. (1995)
18. AT&T U.S. Network Latency, Feb. 2005, http://ipnetwork.bgtmo.ip.att.net/pws/network_delay.html.