Outbound Load Balancing in BGP Using Online Simulation

Tao Ye, Hema Tahilramani Kaur, Shivkumar Kalyanaraman
Electrical Computer and Systems Engineering Department,
Rensselaer Polytechnic Institute, Troy, NY-12180.
{yet3,hema,shivkuma}@networks.ecse.rpi.edu

***Abstract*— In this paper, we investigate a specific inter-domain traffic engineering problem: the outbound load balancing in a multi-homed Autonomous System (AS). We present an optimization-based approach for this problem *without modifying BGP protocol* in any way. In this approach, the load balancing problem is generalized as a black-box optimization problem. The Online Simulation system [1], specifically designed for this class of problems, can be applied to adaptively adjust BGP configuration to achieve load balancing. This approach is also very flexible in that other optimization objectives can be used to achieve different TE objectives easily. The simulation results show that the proposed scheme substantially improves the load balancing or packet drop probability at the outbound links of a multi-homed AS.**

***Index Terms*— BGP, Load-balancing, Optimization, Online Simulation, Traffic Engineering**

I. Introduction

In the current Internet, IP traffic is mapped onto the network by standard routing protocols, such as, Open Shortest Path First (OSPF) for intra-domain traffic and Border Gateway Protocol (BGP) for inter-domain traffic. When routing the traffic, the routing algorithms used in these protocols normally select the shortest path without taking into account the traffic conditions and Quality of Service (QoS) constraints. These routing protocols are often called *topology-driven*. The routing generated by such algorithms tends to result in a highly uneven mapping of traffic. Some links may get very congested while the others may be consistently under-utilized. This phenomenon has been confirmed by many traffic measurements[2], [3] where a large variation in link utilizations is observed across the network. Traffic Engineering (TE) tries to eliminate this situation by adapting the network routing according to the prevailing traffic conditions.

Traffic engineering deals with mapping traffic to physical network topology in order to improve the performance and use resources efficiently. Inter-domain TE work has focused on multi-homed Autonomous Systems (AS), in-bound/out-bound load-balancing between adjacent ASes using BGP attributes (e.g. Multi Exit Discriminator (`MED`), `local_pref`, `as_path`, etc.) [4]. Ideally, a good traffic engineering solution can be obtained by using the knowledge of network-wide traffic demands. This is feasible in the case of intra-domain traffic engineering since all the routers are under the same administrative organization. Network-wide or Ingress/Egress monitoring can be used to estimate traffic demand statistics. Moreover, control over all the routers in the network is possible. However, in the case of inter-domain routing, with different administrative organizations, the traffic demand statistics are usually kept private and the control over routers outside the administrative organization is almost impossible. Therefore, in the inter-domain case, the traffic engineering can only be performed locally.

The ASes are increasingly becoming multi-homed [4]. The outbound traffic of an AS may be routed on any of the multiple links, depending on the decision made by the inter-AS routing algorithm, usually BGP. BGP routing decisions are made by a series of policy filters. Most ASes use the shortest AS path for most destinations. This may lead to unbalanced load even amongst the multiple outbound interfaces of an AS. In this paper, we consider the problem of load-balancing outbound traffic in BGP from the perspective of a single AS. We show that this is an NP-hard problem. We have formulated this problem as an optimization problem and propose to use the Online Simulation (OLS) tool to solve this problem. The advantage of using OLS is that it is very flexible. Using this approach, the service provider may easily use any other optimization criteria, such as minimizing the packet loss in the network, to achieve different traffic engineering goals.

This paper is organized as follows. In Section II we motivate this work. Section III describes the Online Simulation network management tool. The outbound load balancing problem in BGP environment is described and the optimization-based approach is presented in Section IV. Section V presents simulation results and Section VI ends this paper with concluding remarks.

II. Motivation

The main motivation for this work is the growing interest in the methods that achieve TE objectives without requiring modifications to the existing protocols. These methods are based on the assumption that there exist some parameters in network protocols such that the performance is sensitive to these parameter settings. These protocol parameters can then be tuned to achieve various TE objectives. For example, current work in the area of intra-domain TE [5] has focused on optimizing OSPF link weights for load-balancing. Changing link weights impacts the routes used by OSPF which in turn results in a different load offered to the link.

BGP provides only some simple capabilities for TE between AS neighbors. The `MED` attribute can be used by an AS to in-

This work was supported by the DARPA Contract No. F30602-00-2-0537

form its neighbor of a preferred connection (among multiple physical connections) for inbound traffic to a particular address prefix. Usually it is used by the service providers on the request of their multi-homed customers. Lately, it is also being used between the service providers. The `local_pref` attribute is used locally within the AS to prefer an outbound direction for a chosen destination prefix, AS or exit router. Recent work [6] notice that it is possible to automatically tune the `local_pref` parameters of "hot-prefixes" to control outbound traffic subject to a range of policy constraints. However, they do not provide any mechanism to do this. The `as_path` attribute has also been used to achieve TE objectives. `as_path` is "stuffed" or "padded" with additional instances of the same AS number to increase its length and expect lower amount of inbound traffic from the neighbor AS to whom it is announced. However, this may lead to a large overhead if done too often. Another way used to achieve some TE is to subvert the BGP-CIDR address aggregation process. In particular an AS may extract more-specifics, or de-aggregate it and re-advertise the more-specifics to other ASes. The longest-prefix match rule in IP forwarding will lead to a different route for the more specific address. However, this is achieved at the expense of larger number of entries in forwarding tables. This is an indirect and undesirable way to achieve inbound load-balancing. One way to avoid subverting CIDR aggregation (shown in our recent work [7]), in the case of multi-homed *stub AS*, is by mapping the inbound load-balancing problem to an address management problem. Alternatively, AS neighbors may agree on BGP community attributes [8] (that are not re-advertised) to specify traffic engineering. We notice that inbound load-balancing is considerably complex and requires re-advertisements or support from neighboring ASes. However, outbound load-balancing is simpler, and can be achieved by impacting local policy changes.

III. Online Simulation (OLS) Network Management Tool

We have developed a general network management tool which relies on the assumption that network protocols have parameters that impact the performance and a good parameter setting depends on the network traffic and topology. However, due to the complex nature of interactions between the parameter setting and the performance measure, it is rarely possible to find optimal parameter settings *analytically*. The Online Simulation [1] formulates the performance improvement as a *black-box* optimization problem and uses simulation to evaluate the objective function (also referred to as the performance metric). The network is considered as a black-box with the parameters as its input and the performance metric as its output. The advantage of this approach is that it makes the OLS a very *flexible* system whose use is *not restricted in one specific protocol or one performance objective*.

Figure 1 shows the functional block diagram of the (OLS) scheme and its interaction with the network. In the context of network optimization, a highly efficient search algorithm is needed to find "good" OSPF link weight setting since the network is a dynamic system and network conditions may change significantly from time to time. Furthermore the search algorithm should be *scalable to high-dimensional problems* since there may be hundreds of parameters in a network. Another issue that needs to be considered is that network simulation only provides an approximate estimation of network performance. This means that the objective function is superimposed with small random noises due to inaccuracies in network modelling, simulation, etc. To address these issues, we have developed a Recursive Random Search (RRS) scheme that is fast, scalable to high-dimensional problems and robust to noise in the objective function. The RRS is based on the high-efficiency feature of random sampling in initial steps and uses a recursive shrink and re-align procedure to search a high-dimensional parameter space for optima. The reader may refer the technical report [9] for details and performance study of RRS.

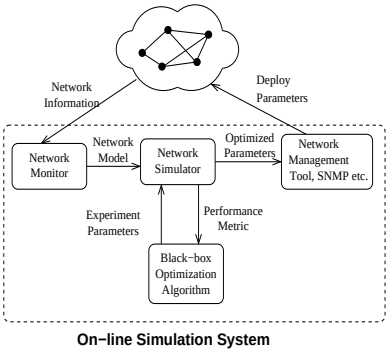

Fig. 1. On-line simulation framework for optimization of network protocols

In our previous work, we have used OLS for minimizing packet loss in an OSPF network by optimizing link weights [10] and adaptive tuning of RED [11]. In this paper, we consider the load balancing problem in BGP environment. In this paper, we consider a specific load balancing problem, the outbound load balancing problem of a single AS. We solve this problem using the OLS scheme.

IV. Outbound Load Balancing in BGP

The BGP attribute `local_pref` holds the highest priority in the policy filter hierarchy, i.e. the BGP will choose the path with highest `local_pref` over any other policy attribute. Therefore, if we know the desired routing to meet the traffic engineering objective, we can use the `local_pref` to over-ride the default routing. Based on this idea, we propose a load balancing scheme that calculates the ideal routing based on current traffic conditions and deploys the preferred routing using the `local_pref` attribute. In the following sub-sections, we describe the details of this scheme.

A. Traffic Demands

Given a certain outbound traffic demand, load balancing aims to split this traffic demand and distribute them evenly among different links. Usually, the traffic demand is composed of a number of traffic flows. In the finest granularity, a traffic flow is determined by the source and destination IP addresses and port numbers. In a coarse granularity, a traffic flow can be identified

by the source and destination AS-pair. Internet measurements have shown that the traffic aggregate based on destination prefixes is suitable for load balancing [2]. The destination prefixes are relatively stable through the day and on per-hour time scales. We have used this granularity for defining a flow in our load balancing scheme. In other words, the traffic demand is split based on the per destination-prefix level of flows. One may use many passive measurement methods, such as monitoring SNMP management information base (MIBs), packet-sampling etc., to obtain the traffic demand statistics [12].

A typical BGP routing table consists of thousands of entries for various destination prefixes. It will be very complex and undesirable to work with such a large number of traffic flows. Many measurements [2], [3] have demonstrated that the Internet traffic exhibits the so-called elephant and mice phenomenon. A small number of traffic streams, known as *elephants*, generate a large portion of total traffic whereas a large number of steams, the *mice*, generate a small portion of total traffic. Another observation is that the elephant traffic flows are usually very stable over time. Therefore, elephant flows are suitable to be re-routed for load-balancing purpose. It has been found that the top 9% of flows between ASes account for 86.7% of the packets or 90.7% of the bytes transmitted [3]. Therefore, it is not necessary to consider all the traffic flows in the load balancing scheme. In this paper, we only attempt to adjust the routing of approximately top 10% destination prefixes in the routing table based on their traffic demands, i.e., a few hundred of flows [1].

B. Optimal Routing Calculation for Load Balancing

In addition to traffic demands, network topology should also be known to calculate optimal routing for traffic engineering objectives. In the case of outbound load balancing, only the number of outbound links and their bandwidths need to be known. Given the knowledge of traffic demand and outbound link information, the optimal routing for load balancing can be calculated.

Let m be the number of outbound links in the AS under consideration. Let l_i and c_i, $i = 1 \ldots m$ denote respectively the i^{th} outbound link of the AS under consideration and its capacity (or bandwidth). All the outbound traffic of this AS will be routed on these links. If s_i, $i = 1 \ldots m$ denotes the share of output traffic on i^{th} link (or the total traffic carried by the i^{th} link), then the utilization of link l_i is given by s_i/c_i. The objective of load balancing is to minimize the maximum link utilization among all the outbound links, i.e.,

$$\text{minimize} \max_{i=1\ldots m} \frac{s_i}{c_i} \tag{1}$$

Let n denote the number of selected destination prefixes and d_j, $j = 1 \ldots n$, denote the average offered load for these destinations. Our load balancing scheme attempts to adjust the routing of these n prefixes in order to minimize the objective function in Equation (1). Let $\mathcal{D}_i$ denote the subset of the n prefixes that are routed on link l_i under adjusted routing. If f_i denotes the load on link l_i generated by the other 90% traffic flows, which is sent on this link by the default BGP routing, then Equation (1) becomes

$$\text{minimize } \Phi = \max_{i=1\ldots m} \Big(\sum_{j\in\mathcal{D}_i} \frac{d_j}{c_i}\Big) + \frac{f_i}{c_i} \tag{2}$$

where, the first term represents the percentage load due to the selected 10% flows and the second term represents the percentage load generated by the other 90% flows on link l_i. This problem can also be written as the following integer programming problem.

$$\begin{aligned} &\text{minimize} \quad t \\ &\text{subject to} \quad \sum_{j=1}^{n} x_{ij}\frac{d_j}{c_i} + \frac{f_i}{c_i} \le t,\ i = 1 \ldots m \\ &\qquad\qquad \sum_{i=1}^{m} x_{ij} = 1,\ j = 1 \ldots n \\ &\qquad\qquad x_{ij} \in \{0,1\},\ i = 1 \ldots m,\ j = 1 \ldots n \end{aligned} \tag{3}$$

where x_{ij} is a binary number and $x_{ij} = 1$ means flow d_j is output on link l_i, otherwise $x_{ij} = 0$. Note that traffic flow d_j may not have all outbound links as its alternative paths. One can assume an arbitrarily large d_j/c_i for those links. The problem represented by Equation (3) is actually a classical task scheduling problem with unrelated parallel machines [13], where a number of tasks with different sizes are assigned to a set of parallel machines. The processing time of each task is different on different machines and the objective there is to minimize the completion time of all tasks by carefully distributing these tasks onto the parallel machines. This problem is NP-hard and approximation algorithms can be used to obtain approximate solutions. For example, in [14] a linear programming technique is first used to obtain a basic solution where there are at most $m-1$ non-integral x_{ij}. Then for these non-integral x_{ij}, an exhaustive enumeration is performed to find the optimal scheduling. Combining the solutions of these two steps can produce an approximate solution with a upper bound of $2t^*$, where t^* denotes the value of t produced by the optimal solution. The time complexity of this method is exponential in the value of m.

Our approach to the problem represented by Equation (2) is to generalize it as an black-box optimization problem and use a general optimization algorithm, such as genetic algorithm and simulated annealing, to search for the near-optimal solutions. The optimization algorithms have been used for other practical problems and are very effective to deliver near-optimal solutions [5], [15].

The advantage of our approach is that flexible optimization objectives can be used without modifying the underlying optimization scheme. In the case of outbound inter-domain TE, we can also minimize the average packet loss instead of maximum link utilization in Equation (1). Using our approach, it is also possible to optimize for multiple objectives by formulating appropriate objective functions and combining these objectives using multi-objective optimization techniques. Consider the case when, in addition to load-balancing, the service

[1] The fraction of optimized destination prefixes can be kept fixed or increased in the event of increase in routing tables. In future, a smaller fraction of destination prefixes may be used if 10% gives a very large number.

provider also prefers to use the shortest paths. It is possible to formulate a multi-objective optimization problem and obtain a solution, using OLS, that meets both load-balancing and shortest path criteria.

We have developed a fast, scalable Recursive Random Search (RRS) algorithm that is used by OLS to find a good solution to the load-balancing problem (Equation (2)). RRS has been specifically designed for the requirements of adaptive networking optimization problems, where the efficiency is very important, i.e. finding a good result within a short time is more important than finding an optimal solution in a very long time. This is different from most of existing optimization algorithms, such as genetic algorithm and simulated annealing, which give more importance to full optimization. The RRS algorithm has been demonstrated to perform very efficiently for engineering problems where high-dimensional and noisy objective functions are quite common. RRS has been shown to be significantly faster than other local search algorithms with a high probability. Another unique advantage of RRS is to be able to automatically exclude negligible parameters from the optimization process. The reader may refer to [9] for the details and performance results of RRS.

C. The Scheme

The complete procedure can be summarized as follows:

Step 1 Extract top 10% destination prefixes, with traffic demands $d_j,\ j = 1 \ldots n$, from the routing table;

Step 2 Calculate d_j/c_i and $f_i/c_i,\ i = 1 \ldots m,\ j = 1 \ldots n$ according to the traffic demand for each prefix and the capacity of each outbound link.

Step 3 Each destination prefix may be reachable by all or some of the outbound links. This information can be obtained from Adj-RIBs-In at a BGP router. Assign a very large value of d_j/c_i for the infeasible routes, so the solution (minimization) will not result in an infeasible solution.

Step 4 Measure or compute the value of Φ for default routing using Equation (2) denoted by Φ^0.

Step 5 Run RRS till a stopping criteria is reached. A stopping criteria can be a limit on time, number of iterations etc.. Let Φ^*, $\overline{r}^*$ denote the value of objective function and corresponding routing at the end of optimization.

Step 6 If $|\frac{\Phi^0 - \Phi^*}{\Phi^0}| \geq \Delta$, where Δ is the threshold, deploy $\overline{r}^*$ by setting a high `local_pref` of desired links for appropriate destination prefixes.

Network conditions keep changing and traffic demands also vary from time to time. Whenever it is found that previous load balancing is deteriorated by these changes, the above procedure can be repeated with new traffic demands to achieve new load balancing. In this way, adaptive load balancing can be accomplished. Network conditions can be directly monitored to detect significant changes and start the optimization process accordingly. The maximum link utilization can also be monitored and when it increases beyond a certain level, the optimization can be performed. This adaptive BGP load balancing procedure is implemented in the on-line simulation framework.

V. Simulation Results

In this section, we present the simulation results for the optimization scheme described in this paper. We have used two optimization objectives, load-balancing objective given by Equation (1) and the minimization of packets dropped, to illustrate our scheme.

A. Optimizing for Load Balancing

In an arbitrary network topology, assume that there are 8 outbound links for the concerned AS. These links have normalized capacities of 100, 100, 100, 100, 45, 45, 45, 12, respectively. We assume the number of top 10% destination prefixes which generate most of the traffic is 148, i.e., there are about 1480 prefixes in the routing table. Note this number is chosen somewhat arbitrarily only for the sake of illustration. In the simulation, we generate only 148 traffic flows instead of all the traffic flows since the actual effect of all the other 90% flows on the simulation is only to reduce the capacity of the links by a certain amount. Therefore, ignoring these flows will not compromise the validity of the simulation results in any way. We assign each destination prefix a certain load such that the total offered load is the 30% of the total capacity of all the links. In the beginning of the simulation, the routing of outbound traffic is decided by the default BGP routing. Then in the simulation, we apply the proposed load balancing scheme to the network. The maximum link utilization (given by Equation 1) before and after optimization is shown is Table I and the link utilizations of outbound links are compared in Figure 3.

	Before optimization	After optimization
Max. Link Utilization	91%	35%

TABLE I

Maximum link utilization before and after optimization

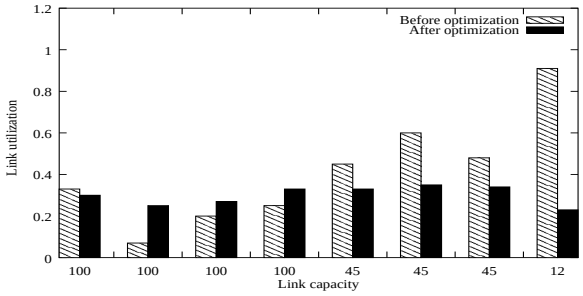

Fig. 2. Link utilization of different outbound links before and after optimization

Figure 3 shows that the default BGP routing leads to an uneven distribution of load across the outbound links, for example, one link is greatly under-utilized with a utilization of 7% while one other link is approaching full utilization with a utilization of 0.91%. After optimization with the proposed scheme, the load distribution become much more even over the outbound links and the utilization of all the links is very close to the ideal value, i.e., the average utilization 30%. This simulation demonstrates that using the proposed optimization approach, the load

balancing on the outbound links can be effectively achieved. Note that the optimization result shown in the table is obtained with an optimization process of only 1500 function evaluations. In a 450MHz Pentium-class PC, it only takes around 1 minute to finish the optimization process. Therefore, this optimization process can be used for on-line tuning of the load balancing objective.

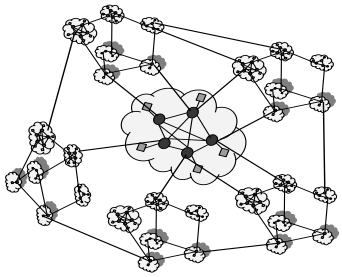

Fig. 3. Network topology used for simulation results

B. Minimizing Packet Loss

In this section, we present the simulation results by using the packet loss objective. We have used the $SSFNet$ [16] simulator to simulate the AS topology shown in Figure 2. This network consists of 31 ASes, 90 routers and 90 hosts. The bandwidth of all the links is assumed to be 10Mbps. In the simulation, the ASes did not have any policies (apart from the `local_pref` used to deploy the optimized routing). We optimize the performance for the AS 0 (shown in the center of graph).

During the simulation, the number of packets dropped on the outbound links is collected and the overall packet drop probability is calculated by the total packet number arrived on all outbound links. The optimization objective here is to minimize the overall packet drop probability. Table II shows the average packet drop probability before and optimization optimization and Figure 4 compares the number of packets dropped on each outbound link. We see that the average packet drop probability of the original routing is very high because of the uneven load distribution. After optimization, the average packet drop probability reduces dramatically since no link is over utilized.

	Before optimization	After optimization
% Packet Drop	41.07%	6.74%

TABLE II

COMPARISON OF AVERAGE PACKET LOSS PROBABILITY AT OUTBOUND LINKS BEFORE AND AFTER OPTIMIZATION

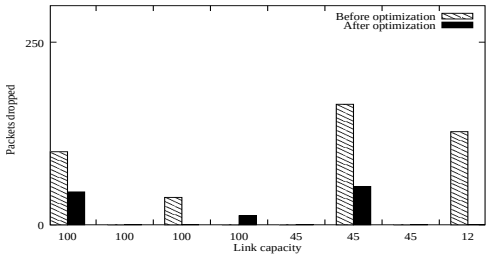

Fig. 4. Packets dropped on various outbound interfaces before and after optimization

VI. CONCLUSIONS

In this paper, we have demonstrated outbound load balancing in BGP using Online Simulation. Basically, this approach formulates the load balancing problem in BGP as a black-box optimization problem and uses an optimization algorithm search for near-optimal solutions. The online simulation scheme, a general adaptive network configuration scheme, has been used to realize the adaptive load balancing in BGP environment. The simulation results demonstrate substantial improvements after optimization using the proposed approach.

The OLS optimization approach presented in this paper is not limited to the objective of load balancing. In fact, one important advantage of this approach is its flexibility. It can be easily adapted to achieve other optimization objectives, such as, minimization of the link packet loss. The flexibility of the OLS approach is also demonstrated in that it can be applied any other network protocol besides BGP, as long as the performance of the protocol is sensitive to its parameter setting. The OLS scheme has also be successfully used for the adaptive configuration of other network protocols, such as OSPF [10] and RED [11].

REFERENCES

[1] Tao Ye and et al. Traffic management and network control using collaborative on-line simulation. In *Proc. of IEEE ICC'01*, Helsinki, Finland, 2001.
[2] S. Bhattacharyya, C. Diot, J. Jetcheva, and N. Taft. Pop-level and access-link-level traffic dynamics in a tier-1 pop. In *ACM SIGCOMM Internet Measurement Workshop*, November 2001.
[3] Wenjia Fang and Larry Peterson. Inter-as traffic patterns and their implications. In *Proceedings of Global Internet 99*, Rio, Brazil, 1999.
[4] G. Huston. Commentary on inter-domain routing in the internet. RFC 3221, December 2001.
[5] Bernard Fortz and Mikkel Thorup. Internet traffic engineering by optimizing ospf weights. In *Proceedings of the INFOCOM 2000*, pages 519–528, 2000.
[6] Jay Borkenhagen Nick Feamster and Jennifer Rexford. Controlling the impact of bgp policy changes on ip traffic. NANOG 25, June 2002.
[7] T. Ye S. Yadav M. Doshi A. Gandhi S. Kalyanaraman, H. T. Kaur. Load balancing in bgp environment using online simulation and dynamic nat. ISMA Workshop, by CAIDA,, December 2001.
[8] J. Stewart III. *BGP-4 Inter-domain routing in the Internet.* Addison-Wesley, 1999.
[9] Tao Ye and Shivkumar Kalyanaraman. A recursive random search for optimizing network protocol parameters. Technical report, ECSE Department, Rensslaer Polytechnique Institute, Dec 2001.
[10] Shivkumar Kalyanaraman Hema T. Kaur, Tao Ye. Minimizing packet loss by optimizing ospf weights using online simulation. submitted.
[11] Tao Ye and Shivkumar Kalyanaraman. Adaptive tuning of red using online simulation. In *Proceedings of IEEE GLOBECOM'2002*, Taibei, Taiwan, Nov. 2002.
[12] M. Grossglauser and J. Rexford. Passive traffic measurement for ip operations. In *The Internet As A Large-scale Complex System*. Oxford University Press, 2002.
[13] L. A. Hall. *Approximation Algorithms for NP-hard Problems*, chapter Approximation Algorithms for Scheduling. PWS Publishing Company, 1995.
[14] C. N. Potts. Analyssis of a linear programming heuristic for scheduling unrelated parallel machines. *Discrete Appl. Math.*, 10:155–164, 1985.
[15] Zelda B. Zabinsky. Stochastic methods for practical global optimization. *Journal of Global Optimization*, 13:433–444, 1998.
[16] SSFNET. network simulator. http://www.ssfnet.org.