Routing with Deceptive Information

C.Casetti, M. Mellia, M. Munafò, C. Racca
Dipartimento di Elettronica, Politecnico di Torino
Corso Duca degli Abruzzi, 24 – I-10129 Torino, Italy

I. INTRODUCTION

In the early years of packet networks, a hot research topic was the definition of efficient *dynamic* routing algorithms, aiming at an optimal exploitation of links by adapting packet routes to instantaneous traffic conditions. Dynamic routing algorithms can in principle offer significant advantages with respect to static routing, since they automatically react to congestion situations, therefore offering better performance and quality of service (QoS). However, the finding that dynamic routing algorithms may lead to route flapping (i.e., a periodic route changes which forces traffic to be routed through an underloaded set of paths, causing a sudden overload of new paths and under load of previous paths) and to the consequent performance degradations, has limited their diffusion. As a consequence, the dynamic features implemented today in routing protocols are mostly limited to automatic reactions to topology changes due to link failures or infrastructure updates.

Despite all the effort spent by the research community, no QoS routing has ever been deployed in the Internet. Several reasons are behind this choice, like traffic unpredictability, protocol complexity, and increased overhead. The last item is particular critical, since each dynamic QoS routing relies on the the network status information which must be diffused among nodes. Therefore this increases both network load (in term of signaling informations) and node power (in term of computational processing). Indeed, compared to static routing protocol, which relies on static information, dynamic algorithms rely on the knowledge of the current status of the network. In this paper we therefore investigate how update policies affect the QoS routing algorithm performance, quantifying the intuitive results that if route selection performed by nodes is based on stale information, the QoS routing algorithm may provide worst performance compared to traditional static routing. To overcome this limitation, we propose a novel yet very simple mechanism that is shown to improve performance of QoS routing algorithms when the state information is slowly updated.

II. PROBLEM OUTLINE

We extended ANCLES [1], a connection-level simulator that was previously developed at the Politecnico di Torino. ANCLES is a generic connection-level simulator, where traffic sources request connections and the network performs all the actions required to manage them. The reader interested in the simulation tool is referred to [2].

A. Modeling data connections

The traffic the simulator models is of elastic nature [3] (such as that generated by prominent connections). Each connection attempts to perform a bulk data transfer whose size is randomly chosen from an exponential distribution with average 2.5 MBytes (20 Mbit). Traffic is uniformly generated by all sources and evenly distributed among all possible destinations. Results are collected over different load situations. As performance metrics, we consider the average throughput of connections that complete the data transfer.

B. Routing Algorithms

As regards routing algorithms implemented in the simulator, beside the static, hop-count based algorithms, that are unable to cope with the variation of available bandwidth in the network, ANCLES implements several dynamic, traffic-driven routing algorithms, like those proposed in [4], [5], [6]. Here, we briefly describe the two algorithms used in the simple performance evaluation carried out in this work:

- *Shortest-Path* (*SP*): for each source-destination pair, the algorithm determines the path with the minimum hop count and routes flows over that path.
- *Minimum-Distance* (*MD*): for each source-destination pair, the path P is chosen which minimizes the quantity: $D(P) = \sum_{l \in P} \frac{1}{b_l}$ where b_l is the max-min fair bandwidth that is available to a new connection over link l belonging to path P [5].

The distribution among network routers of the updated QoS parameters, such as the currently available bandwidth, is assumed to occur every s seconds independently by each node. Updates are flooded to other nodes according to a simple model that accounts for both transmission and queueing delay. Each node maintains its own link state database based on the information received by other nodes.

The choice of the MD algorithm was made because it was found to provide good performance among traffic-driven routing algorithms [6].

III. PERFORMANCE EVALUATION

To compare the performances of the system, we considered a network scenario with a random generated 24-node topology with an average connectivity degree of 3. The values of the timer that triggers a new LSA flooding is shown near the MD with the notation *update* time, so 'MD - 10s' means an MD implementation with a *periodicUpdate* every 10 s. The 'MD-ideal') assumes that each node has a precise indication of the link load at each time.

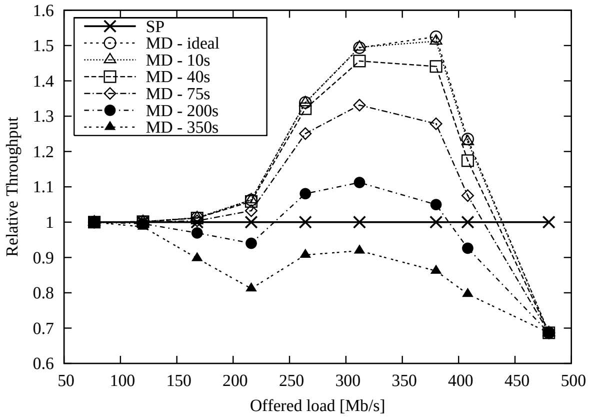

Fig. 1. Average per-connection throughput gain with respect to SP

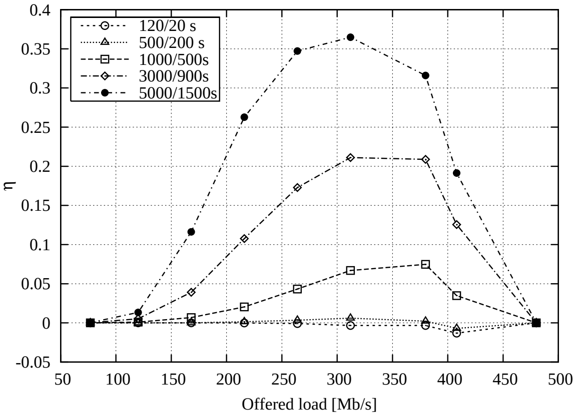

Fig. 2. Throughput gain for the DIE algorithm

Figure 1 presents the average per-connection throughput, as a function of the offered load, while the same results. This results provide a way of gouging the gains of the different strategies used for the MD algorithm over the "standard" Shortest Path. As expected, the ideal case performs best; however, by allowing as many as 40 seconds of Periodic Update time, the performance remains quite close to the ideal case, and thus fully acceptable.

Even if these results are quite positive in the comparison between the MD and the SP algorithms, the MD algorithm performance depends strongly on the values of the update timers: large timer values can reduce the per-connection user throughput below the equivalent SP levels.

IV. THE DIE PROTOCOL: DECEPTIVE INFORMATION EXCHANGE

From the results presented in the previous section, we can foresee at least two solutions to the performance decrease experience by the MD algorithm in presence of inaccurate information: i) increase the frequency of updates when a link is congested; ii) disseminate "doctored" LSA information with the purpose of decreasing congestion on overloaded links.

The first solution, though viable, would entail periodical LSA floods that increase the network and node overhead and might even worsen the problem. The second solution, which we are pursuing in this paper, is based on a simple line of reasoning. If a link is congested, it is sensible to inform all nodes, so that new connections are routed away from that link. If a link is not congested, advertising its state might trigger a gold rush to that link by a great number of incoming connections; furthermore, if the link update period is larger than the arrival rate of new connections, the link might be easily overloaded in a short time.

We propose a modification to the LSA distribution mechanism, called DIE (Deceptive Information Exchange). According to DIE, the link state information of a generic link l that a router must advertise is altered by a threshold mechanism. If the number of connections crossing l is higher than a threshold, T, the LSA carries correct information on the link load; otherwise, the LSA carries a *deceptive* link load, higher than the actual one.

As threshold T_l each node computes a running average of c_l for every link l. The *deceptive* load information L' is defined as

$$L' = (1-\beta)c_l[n] + \beta T_l[n]$$

depending on the value of $\beta \in [0,1]$, L' advertises the actual c_l, or its average value T_l, as determined from recent samples.

A. Performance evaluation of the DIE protocol

In Figure 2 we compare the performance of the DIE algorithm plotting the relative gain $\eta = \frac{thr_{DIE} - thr_{old}}{thr_{old}}$, being thr_{old} the throughput in the unmodified version of the algorithm. The plots show an increase in the gain at the rarefaction of the LSAs, especially for medium/high load. At higher load, as usual, the network status is such that no algorithm can perform much better than SP (all MD perform the same). The curves in the plot are for very high timer values (larger than 120s), especially if compared with the average connection duration 20 s, showing that the algorithm is performing well even in high (and unrealistic) under-sampling scenarios.

REFERENCES

[1] M.Ajmone Marsan, A.Bianco, C.Casetti, C.F.Chiasserini, A.Francini, R.Lo Cigno, M.Mellia, M.Munafò, "An Integrated Software Environment for the Simulation of ATM Networks", *SCSC'97, Summer Computer Simulation Conference*, Arlington, Virginia, USA, July 1997

[2] TNG-Polito *ANCLES home page*, `http://www.tlc-networks.polito.it/ancles/`

[3] C.Casetti, R.Lo Cigno, M.Mellia, M.Munafò, Z.Zsoka, *A Realistic Model to Evaluate Routing Algorithms in the Internet*, Globecom 2001, San Antonio, Texas, USA, November 25-29, 2001.

[4] Z.Wang, J.Crowcroft, "QoS Routing for Supporting Multimedia Applications," *IEEE JSAC*, Vol.14, N. 7, September 1996, pp 1228-1234.

[5] Q.Ma, P.Steenkiste, H.Zhang, "Routing High-Bandwidth Traffic in Max-Min Fair Share Networks," *ACM SIGCOMM 1996*, Stanford, CA, August 1996.

[6] C. Casetti, G. Favalessa, M. Mellia, M. Munafò, "An Adaptive Routing Algorithm for Best-effort Traffic in Integrated-Services Networks", *16th International Teletraffic Congress (ITC-16)*, Edinburgh, UK, June 1999