Behaviour of Maximum-Throughput Dynamic Routing in Loopy Networks*

Venkataramana Badarla and Douglas J. Leith

Hamilton Institute, NUI Maynooth, Ireland

Abstract. Maximizing network throughput via back-pressure routing is the subject of a considerable body of literature. However, the associated optimal throughput guarantees come at the cost of excessively high delays. Though there exists some proposals in literature for improving the delay performance, the impact of these proposals on the delay performance is very little in loopy networks (i.e., networks with routing loops). In this demonstration, with series of experiments over different topologies, we show that the basic back-pressure algorithm and two of its variants for improving delay performance induce extensive routing loops with associated high packet delay even in simple network topologies.

1 Introduction

Throughput maximization is an important issue in current Internet and it would continue to be an issue in future Internet. Using dynamic multi-path routing, the state of art back-pressure routing[1] offers considerable gains in throughput over conventional single-path routing algorithms. Indeed, it guarantees to achieve the network capacity and so its throughput performance cannot be bettered by any algorithm. While maximizing network throughput, back-pressure routing comes with no guarantee on network delay. Indeed certain features of the back-pressure routing algorithm suggest that long delays will be common. Firstly, back-pressure routing uses queue buildup at nodes to create a "gradient" within the network that guides routing. However, this may come at the cost of increased queuing delay. Secondly, back-pressure routing tends to explore all paths in a network, including paths with loops and "dead-end" paths that cannot lead to the desired destination. Hence, packets generally may not take the shortest path to their destination, thereby leading to additional delay.

The basic back-pressure algorithm is detailed in Algorithm 1. Each forwarding node in the network uses per flow FIFO queuing and we let $q_n^f(t)$ denote the number of packets queued for flow f at node n at time t. Roughly speaking,

* This material is based upon works supported by the Science Foundation Ireland under Grant No. 07/IN.1/I901.

whenever it has a transmission opportunity each node n forwards a packet from the flow f^* to the next hop $m^*(f^*)$ that jointly maximizes the utility

$$U^f_{n,m}(t) = \left(q^f_n(t) - q^f_m(t)\right) R_{n,m}$$

where $R_{n,m}$ is the mean transmit rate of the link from node n to node m. Observe

Algorithm 1 Basic back-pressure algorithm

1: For each flow f and neighbor node m, node n computes utility $U^f_{n,m}(t) = \left(q^f_n(t) - q^f_m(t)\right) R_{n,m}$, $m^*(f) = \arg\max_m U^f_{n,m}$, $f^* = \arg\max_f \left(U^f_{n,m^*(f)}\right)$ (ties broken arbitrarily).
2: If $U^{f^*}_{n,m^*(f^*)} > 0$, then node n schedules flow f^* and forwards $\min(q^{f^*}_n(t), R_{n,m^*(f^*)})$ packets to neighbor $m^*(f^*)$.
3: Otherwise node n takes no action at time t.

that this back-pressure algorithm tends to transmit packets to the neighbor(s) with the smallest queue and highest link rate and intuitively the queue backlogs provide a "gradient" down which packets are routed. However, it commonly occurs that $q^f_n(t)$ and $q^f_m(t)$ differ by only a small amount. The routing "gradient" is then both small and rapidly fluctuating, in which case routing loops can readily be induced. Furthermore, observe that even when a neighbor has no connectivity to the destination of a flow, packets will still be forwarded to this neighbor until such time as a sufficiently large queue backlog has developed to prevent further packets stop being forwarded in that direction. However, the packets already sent in that direction will never reach the flow destination.

Back-pressure With Distance Weighting: In [2] it is noted that the utility function $U^f_{n,m}$ can by modified by summing with a finite constant without affecting the stability properties of the basic back-pressure routing, and in particular, that selecting a constant based on the shortest-path distances from nodes n and m to the destination of flow f is a possible choice. We therefore consider a modified utility function of the form

$$U^f_{n,m}(t) = \left(q^f_n(t) - q^f_m(t)\right) R_{n,m} + \frac{M}{D^f_m}$$

where the intuition is that since the utility is inversely proportional to the distance D^f_m, shortest paths are favored when making forwarding decisions. The design parameter M allows the relative weight of the distance term to be adjusted. We refer to use of this utility function as *distance aided routing*.

2 Implementation, Demo Setup and Snapshot of Results

We implemented the back-pressure algorithms as a *DynamicRouter* element within the Click router framework in Linux. The Click router framework provides a modular architecture that lies within the Linux queuing layer (*i.e*, between the network layer and the device driver layer) and interacts with the network layer and the interface device driver via the Click elements ToHost/FromHost and ToDevice/FromDevice, respectively.

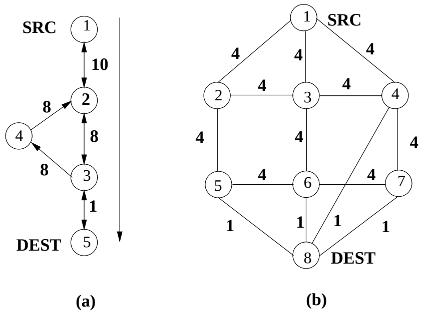

Fig. 1. Network Topologies.

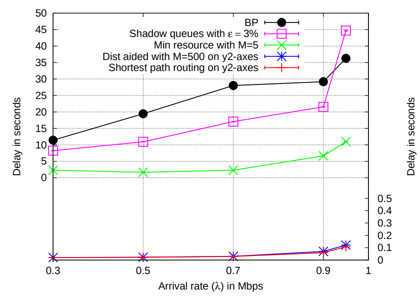

Fig. 2. Delay performance over the topology in Fig. 1(a).

Table 1. Demonstration setup for the Demo Session

Item	Description
Soekris computers (quantity 10)	Single-board computers with 433MHz CPU, 256MB RAM and four 100Mbps ethernet ports
N/w topologies	Shown in Figure 1
Performance metrics	a) packet delay b) throughput c) Buffer size at intermediate nodes d) reassembly buffer at receiver
Traffic type	a) TCP using iperf and scp b) UDP using mgen

Details of the demonstration setup, such as type of computers will be used, network topologies will be considered etc are given in Table 1. Fig 2 presents snapshot of the experimental results conducted over the topology in Fig 1(a). In Fig. 2(a), which shows measured delay performance, we can observe that the delay for back-pressure algorithm and its variants (shadow-queue and min-resource algorithms [3]) is excessively high (ranges from 3s to 45s) at all offered loads. Also can be observed that the proposed distance aided approach shows the lowest delays which are at par with the shortest path routing. More detailed results over the second topology will be shown during the live demonstration session at the conference venue.

References

1. A. L. Stolyar, "Maximizing Queueing Network Utility subject to Stability: Greedy-Primal Dual Algorithm," *Queueing Systems*, vol. 50, no.4, pp. 401-457, 2005.
2. L. Georgiadis et al, *Resource Allocation and Cross Layer Control in Wireless Networks* 2006.
3. L. Bui et al, "Novel Architectures and Algorithms for Delay Reduction in Back-pressure Scheduling and Routing", *in Proc. IEEE Infocom Mini-Conference* 2009.