Volume-3, Issue-11, Nov-2014 • ISSN No 2277 - 8160

Research Paper | Technology

Novel Architecture for Delay Reduction in the Back Pressure Scheduling Algorithm

Anjaneyulu Mittapalli (M.Tech), JNTUK, Kakinada

M.Srilekha (M.Tech), JNTUH, Hyderabad

ABSTRACT

Back-pressure-based adaptive routing algorithms where each packet is routed along a possibly different path have been extensively studied in the literature. However, such algorithms typically result in poor delay performance and involve high implementation complexity. In this paper, we develop a new adaptive routing algorithm built upon the widely studied back-pressure algorithm. We decouple the routing and scheduling components of the algorithm by designing a probabilistic routing table that is used to route packets to per-destination queues. The scheduling decisions in the case of wireless networks are made using counters called shadow queues. The results are also extended to the case of networks that employ simple forms of network coding. In that case, our algorithm provides a low-complexity solution to optimally exploit the routing–coding tradeoff.

KEYWORDS :

SYSTEM ANALYSIS

EXISTING SYSTEM:

The back-pressure algorithm introduced has been widely studied in the literature. While the ideas behind scheduling using the weights suggested in that paper have been successful in practice in base stations and routers, the adaptive routing algorithm is rarely used. The main reason for this is that the routing algorithm can lead to poor delay performance due to routing loops. Additionally, the implementation of the back-pressure algorithm requires each node to maintain per-destination queues that can be burdensome for a wire line or wireless router.

DISADVANTAGES OF EXISTING SYSTEM:

In an existing algorithms typically result in poor delay performance and involve high implementation complexity.

PROPOSED SYSTEM:

The main purpose of this paper is to study if the shadow queue approach extends to the case of scheduling and routing. The first contribution is to come up with a formulation where the number of hops is minimized. It is interesting to contrast this contribution. The formulation has the same objective as ours, but their solution involves per-hop queues, which dramatically increases the number of queues, even compared to the back-pressure algorithm. Our solution is significantly different: We use the same number of shadow queues as the back-pressure algorithm, but the number of real queues is very small (per neighbor). The new idea here is to perform routing via probabilistic splitting, which allows the dramatic reduction in the number of real queues. Finally, an important observation in this paper, not found is that the partial "decoupling" of shadow back-pressure and real packet transmission allows us to activate more links than a regular back-pressure algorithm would. This idea appears to be essential to reduce delays in the routing case, as shown in the simulations.

ADVANTAGES OF PROPOSED SYSTEM:

√ Our adaptive routing algorithm can be modified to automatically realize this tradeoff with good delay performance.

√ The routing algorithm is designed to minimize the average number of hops used by packets in the network. This idea, along with the scheduling/routing decoupling, leads to delay reduction compared with the traditional back-pressure algorithm.

SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb.

SOFTWARE REQUIREMENTS:

- Operating system : - Windows XP.
- Coding Language : J2EE
- Data Base : MYSQL

IMPLEMENTATION MODULES:

- Exponential Averaging
- Token Bucket Algorithm
- Extra Link Activation
- Choice of the Parameter

MODULE DESCRIPTION:

Exponential Averaging:

In this module, using the concept of shadow queues, we partially decouple routing and scheduling. A shadow network is used to up-date a probabilistic routing table that packets use upon arrival at a node. The same shadow network, with back-pres-sure algorithm, is used to activate transmissions between nodes. However, first, actual transmissions send packets from first-in–first-out (FIFO) per-link queues, and second, potentially more links are activated, in addition to those activated by the shadow algorithm

Token Bucket Algorithm:

In this module, computing the average shadow rate and generating random numbers for routing packets may impose a computational overhead of routers, which should be avoided if possible. Thus, as an alternative, we suggest the following simple algorithm. At each node, for each next-hop neighbor and each destination, maintain a token bucket.

Extra Link Activation:

Under the shadow back-pressure algorithm, only links with back-pressure greater than or equal to M can be activated. The stability theory ensures that this is sufficient to render the real queues. On the other hand, the delay performance can still be unacceptable. Recall that the parameter M was introduced to discourage the use of unnecessarily long paths. However, under light and moderate traffic loads, the shadow back-pressure at a link may be frequently less than M, and thus, packets at such links may have to wait a long time before they are processed. One way to remedy the situation is to activate additional links beyond those activated by the shadow back-pressure algorithm.

Choice of the Parameter:

From basic queuing theory, we expect the delay at each link to be inversely proportional to the mean capacity minus the arrival rate at the link. In a wireless network, the capacity at a link is determined by the shadow scheduling algorithm. This capacity is guaranteed to be at least equal to the shadow arrival rate. The arrival rate of real packets is of course smaller. Thus, the difference between the link capacity and arrival rate could be proportional to epsilon. Thus, epsilon should be sufficiently large to ensure small delays while it should be sufficiently small to ensure that the capacity region is not diminished significantly

CONCLUSION

The back-pressure algorithm, while being throughput-optimal, is not useful in practice for adaptive routing since the delay performance can be really bad. In this paper, we have presented an algorithm that routes packets on shortest hops when possible and decouples routing and scheduling using a probabilistic splitting algorithm built on the concept of shadow queues introduced. By maintaining a probabilistic routing table that changes slowly over time, real packets do not have to explore long paths to improve throughput; this functionality is performed by the shadow "packets." Our algorithm also allows extra link activation to reduce delays. The algorithm has also been shown to reduce the queuing complexity at each node and can be extended to optimally tradeoff between routing and network coding.