ISSN:0975-9646

Dynamic Congestion control for TCP in Peer - to - Peer Network

Dr D. Raghu, Thatha Reddy Karri ,Ch. Raja Jacob
Computer Science Dept, Nova College of Engineering Jangareddy Gudem

Abstract-The data transmission carries throughout the internet using TCP protocol, the internet traffic , performance depends on the Transmission control protocol. The congestion control and adaptive routing in isolation, much less attention has been paid to whether these two resource-allocation mechanisms work well *together* to optimize user performance. The characteristics of a particular version of TCP are defined by the congestion control algorithm. In the paper we propose analyze the interaction between congestion control and dynamic routing. Dynamic routing assume to reduces the network traffic by selecting alternative path. The proposed solutions focus on a variety of problems, starting with the basic problem of eliminating the phenomenon of congestion collapse, and also include the problems of effectively using the available network resources in different types of environments in peer- to – peer network. In distributed, and heterogeneous environment such as the Internet, effective network use depends not only on how well a single TCP based application can utilize the network capacity, but also on how well it cooperates with other applications transmitting data through the same network.
Keywords: Network utility maximization, congestion control, Dynamic routing, TCP/IP, packet reordering.

I. INTRODUCTION

In the internet the allocation of network resources to maximize user utility congestion control (in TCP) and routing (in IP). Congestion control allocates the limited capacity on each link to competing flows, while routing determines which flows pass through which links. The congestion control proposals for TCP that preserve its fundamental host-to-host principle, meaning they do not rely on any kind of explicit signaling from the network.1 The proposed algorithms introduce a wide variety of techniques that allow senders to detect loss events, congestion state, and route changes, as well as measure the loss rate, the RTT, the RTT variation, bottleneck buffer sizes, and congestion level with different levels of reliability. TCP congestion control implicitly solves network-utility maximization problems [4], but these studies assume a static mapping of traffic to network paths. The key feature of TCP is its ability to provide a reliable, bi-directional, virtual channel between any two hosts on the Internet. Since the protocol works over the IP network [3], which provides only best-effort service for delivering packets across the network, the TCP standard [1] specifies a sliding window based flow control. This flow control has several mechanisms. First, the sender buffers all data before the transmission, assigning a sequence number to each buffered byte. Continuous blocks of the buffered data are packetized into TCP packets that include a sequence number of the first data byte in the packet. Second, a portion (window) of the prepared packets is transmitted to the receiver using the IP protocol. As soon as the sender receives delivery confirmation for at least one data packet, it transmits a new portion of packets the sender holds responsibility for a data block until the receiver explicitly confirms delivery of the block. As a result, the sender may eventually decide that a particular unacknowledged data block has been lost and start recovery procedures. The acknowledge data delivery, the receiver forms an ACK packet that carries one sequence number and (optionally) several pairs of sequence numbers. The former, a *cumulative ACK*, indicates that all data blocks having smaller sequence numbers have already been delivered. The latter, a *selective ACK*, explicitly indicates the ranges of sequence numbers of delivered data packets. To be more precise, TCP does not have a separate ACK packet, but rather uses flags and option fields in the common TCP header for acknowledgment purposes. The interaction between congestion control and Dynamic routing based on centrally minimizing the maximum link utilization. Congestion control is not modeled analytically and the results are limited to networks with a single bottleneck. The only paper with a detailed analytic model of TCP/IP interaction is the recent work [14]. This study views the joint optimization problem as maximizing user utility with both the source rates and network paths as optimization variables. In particular, the interaction between TCP and IP is modeled as dynamic routing based on congestion on the links, where congestion price can be interpreted as link metrics like packet loss or queuing delay. However, the work in [14] assumes a particular separation of timescales: congestion control converges instantaneously, followed by one step of dynamic route optimization, and the process repeats. In reality, the joint system consists of two distributed control loops running concurrently.

II. DYNAMIC ROUTING TO REDUCE OPTIMIZATION

Dynamic shortest path tree (SPT) algorithm for a router determines a new SPT for a root node in response to a link-state or other network topology change. The dynamic SPT algorithm determines the new SPT as an optimization problem in a linear programming based in an existing SPT in the router. The dynamic SPT algorithm emulates maximum decrement of selecting nodes of the existing SPT for consideration and update of parent node, child nodes, and distance attributes based on the maximum decrement. For the maximum decrement, a node in the existing SPT is selected by each iteration based on the greatest potential decrease (or least increase) in its distance attribute. The length of a string is defined by its link's weight. The set of strings connecting the nodes defines a path between the root node and a particular node. The shortest path is the path defined by the strings from a root node to a particular node that are tight. For the dynamic SPT algorithm, an increase (or decrease) in an edge weight in an existing SPT corresponds to a lengthening (or shortening) of a string. By sequentially pulling nodes away in a single direction from the node of the root node, the new SPT becomes defined by the node and tight strings. The routing processor of a router for routing packet network based on a shortest path tree (SPT) defining paths between nodes in the network, the routing processor comprising: an initialization module adapted to generate, for a current SPT, a temporary list of nodes in the network affected by a change in the weight of a link between two nodes in the network, wherein the change corresponds to the addition of a new link in the network or an increase or decrease in the weight of an existing link in the network; a node selection module adapted to select a node of the list in accordance with a maximum decrement criterion, the maximum decrement criterion identifying the node in the list having a most negative or a least positive associated distance change resulting from the change in the weight of the link; an update module adapted to update one or more paths in current SPT for nodes reachable from the selected node.

Network Routing

A network is modeled as a set of *L* uni-directional links with finite capacities *c* = (*cl; l* = 1*; : : : ;L*), shared by a set of *N* source-destination pairs, indexed by *i* (we will also refer to a source-destination pair simply as "source *i*"). There are a total of *Ki* acyclic paths for each source *i*, represented by a *L £ Ki* 0-1 matrix *Hi*, where *Hilj* =1*;* if path *j* of source *i* uses link *l* 0*;* otherwise. Let *Hi* be the set of all columns of *Hi* that represents all the available paths for *i*. Define the *L £ K* matrix *H* as *H* = [*H*1 *: : : HN*]*;* where *K* := P *i Ki*. *H* defines the topology of the network. Let *wi* be a *Ki* £ 1 vector where the *j*th entry represents the fraction of *i*'s flow on its *j*th path such that *wij* ¸ 0 *8j;* and 1*Twi* = 1*;* where 1 is a vector of an appropriate dimension with the value 1 in every entry. We allow *wij 2* [0*;* 1] for multipath routing. Collect the vectors *wi*, *i* = 1*; : : : ;N*, into a *K £ N* block-diagonal matrix *W*. Define the corresponding set *Wm* for multipath routing as *fWj W* = diag(*w*1*; : : : ;wN*) *2* [0*;* 1]*K£N;* 1*Twi* = 1 *g*. As mentioned above, *H* defines the set of acyclic paths available to each source, and represents the network topology. *W* defines how the sources load balance across these paths. Their product defines a *L £ N* routing matrix *R* = *HW* that specifies the fraction of *i*'s flow that traverses each link *l*.

TCP Model

We interpret the equilibria of various TCP congestion-control algorithms as solutions of a network utility maximization problem defined in [4]. Suppose each source *I* has a utility function *Ui*(*xi*) as a function of its total transmission rate *xi*. We assume *Ui* is increasing and strictly concave (as is the case for TCP algorithms [7]). The constrained utility maximization problem over *x* for a fixed *R* is maximize P*i Ui*(*xi*) subject to *Rx · c:* (1) The duality gap for the above optimization problem is zero. Zero duality gap means that the minimized objective value of the Lagrange dual problem is equal to the maximized total utility in the primal problem (1). We briefly review the solution to (1). First form the Lagrangian of (1): *L*(*x; p*) = X*iUi*(*xi*) +X*l pl* (*cl* ¡ *yl*) where *pl* ¸ 0 is the Lagrange multiplier (*i.e.*, congestion price) associated with the linear flow constraint on link *l*, and *yl* = P *i Rlixi* is the load on link *l*. It is important that the Lagrangian can be decomposed for each source: *L*(*x; p*) = X *i*" *Ui*(*xi*) ¡ Ã X *l Rlipl* ! *xi* # + X *l clpl* = X *I Li*(*xi; qi*) + X*l clpl* where *qi* =P *l Rlipl* is the end-to-end price for source *i*. The Lagrange dual function *g*(*p*) is defined as the maximized *L*(*x; p*) over *x* for a given *p*. This 'net utility' maximization can be conducted distributive by each source, as long as the aggregate link price *qi* is feedback to source *i*: *x*¤*I* (*qi*) = argmax *qi* [*Ui*(*xi*) ¡ *qixi*] *; 8i:* (2) The Lagrange dual problem of (1) is to minimize *g*(*p*) over *p* ¸ 0. An iterative gradient method can be used to update the dual variables *p* in parallel on each link to solve the dual problem.

III. PROBLEM EVOLUTION

The TCP/IP optimization problem and motivate the usage of congestion control. Then we define three models, comparing and contrasting their timescale assumptions.

Joint Optimization

What kind of TCP/IP interactions would work together to maximize end-user utilities over both rate allocation *x* and routing matrix *R*, solving the following problem: maximize P *i Ui*(*xi*) subject to *Rx · c; x* ¸ 0 *R 2 R;* where both *R* and *x* are both variables optimizing the Lagrangian *L*(*p; x;R*): min *p*¸0 X*i* max *xi*¸0 Ã *Ui*(*xi*) ¡ *xi* min *R2R* X *l Rlipl* ! +X *lclpl:* (5) It hints that dynamic shortest-path routing min*R* P *l Rlipl*, where link cost is based on congestion prices *p*, may be designed to jointly maximize network utility with TCP. This possibility was first investigated in [14], which shows that, under a particular timescale separation, TCP/IP would jointly solve (4) if an equilibrium exists. Such an equilibrium exists if multipath routing is allowed, but it can be unstable. It can be stabilized by adding a static component to link weight, but at the expense of a reduced utility at equilibrium. Before giving the detailed description of the models, we highlight the following idea: TCP adjusts *x*, IP adjusts *R*, each affected by the other through the congestion-price vector *p*(*x;R*), which is clearly a function of both *x* and *R,* and jointly determining the objective of P *i Ui*(*xi*). Since the timescale of TCP is affected by the round-trip time and that of IP determined by routing protocols and operational practices, there can be four different models of the above interaction. Given that IP rarely operates faster than TCP convergence, we have three System Models, including the one in [14] as a special case, described.

Ring Topology and Traffic

One of the goals of this paper is to derive closed-form solutions for the stability conditions of TCP/IP interactions. When link cost is a combination of both congestion price and a static component, analytic solution or even proof of the existence of an equilibrium is an open problem[14]. We thus focus on purely dynamic routing where the link cost is the congestion price. According to the KKT optimality condition (4), congestion price has to be zero when link load is strictly less than link capacity. Therefore, to avoid the case of random routing due to zero link costs, we need a topology and traffic model that can avoid zero congestion. Consider a ring topology with N nodes, each of which being a source with a destination being the clockwise neighbor node. Note that we can interchange l and I indices in this case. Each source has two possible paths: a one hop path and an (N ¡ 1)-hop path. For the problem defined by (1) at optimality, the KKT conditions (4) allows for the constraint Rx · c being satisfied to be a potential solution. If R is invertible, then the constraint would be satisfied with equality and the source rates would be x = R¡1c. In addition, congestion prices would be non-zero and p = R¡1q, where qi = $U0$ i (xi). There is one case when R is not invertible, namely when traffic is split evenly for each source. But this routing configuration would be changed in the next TCP/IP round: there would be at least one link with zero congestion price and the routing adaptation will change the routing matrix to take advantage of the zero-congestion-price link.

IV. SYSTEM ANALYSIS

Recall from earlier definition that System Model Three is similar to the regular TCP model algorithm, the only difference is that the routing matrix can change at any step instead of remaining constant throughout the TCP loop. For the ring topology and traffic model in TCP/IP System Model Three converges to minimum hop routing if the capacity of one link in the ring is sufficiently large and the step size is sufficiently small. It could also converge to the one-source-splitting configuration for other capacity distributions.

Proof*:* System Model Three can only converge to (x¤;R¤) if the future congestion prices can guarantee to maintain the same R. If the R is constant, System Model Three reduces to a TCP loop, and will converge to the optimal x¤ for the given R. Without a loss of generality, we may assume after a number of iterations, at least one link becomes congested, then, following directly from the analysis of System Model One, there is at most one source splitting or going on the longer-hop path. Let the potentially splitting source be 1 and let 0 · a · 1; a = w1 1 parameterize all possible R after shortest path routing. It follows

y1(t) = a ap1(t)+(1¡a) PN2pl(t); yl(t) =1¡aap1(t)+(1¡a)PN2pl(t)+ 1pl(t) ; l 6= 1. There are three cases: 1) One source taking longer-hop path (a = 0): p1(t + 1) = [p1(t) ¡ ®c1]+, this is a monotonically decreasing function, so after a number of iterations, p1 > PN 2 pl will no longer hold and R will change for sure.

V. CONCLUSION

In this paper congestion control is a natural choice of link weights for dynamic routing, it is prone to oscillations if deployed in practice. The key insights of this paper are: faster timescale and more homogeneous link capacity configuration help attaining optimality, while smaller step size enhances stability. In particular, for System Models, it is possible to chose an appropriate step size in order to achieve convergence. The dependency on step size, however, argues against using congestion control as the feedback algorithm. It would be difficult for an operator to determine the appropriate step size for his network topology, this also needs to change as links are added or removed. As future work, we will study the interaction between congestion control and routing with routing models based on traffic-engineering practices. In particular, the routing would be trying to *centrally* minimize a penalty function of link utilization based on a network-wide view of the current offered traffic [11]. Another direction would be to examine heuristics for improving system stability or optimality. In the early ARPANET work [9], it was noted that by adding a static component to the network, or averaging traffic measurement over a longer period of time, stability of the routing improved. This idea can also be applied to our system to improve stability. Another possible heuristic is to send traffic along multiple paths, where the traffic on each path would be inversely proportional to link weight of that path.

REFERENCES

[1] R. J. La and V. Anantharam, "Utility-based rate control in the Internet for elastic traffic," in *IEEE/ACM Trans. Networking*, vol. 10, no. 2, pp. 272-286, April 2002.

[2] S. H. Low, L. Peterson, and L. Wang, "Understanding Vegas: A duality model," *J. of ACM*, 49(2):207–235, March 2002.

[3] S. H. Low. "A duality model of TCP and queue management algorithms," *IEEE/ACM Trans. Networking*, 11(4):525–536, August 2003.

[4] MultiProtocol Label Switching (MPLS)html.charters/mpls-charter.html.

[5] J. M. McQuillan and D. C. Walden, "The ARPA network design decision," *Computer Networks*, 1(5):243-389, August 1977.

[6] Open Shortest Path First (OSPF). http://www.ietf.org/html. charters/ospf-charter.html.

[7] J. Rexford. "Route Optimization in IP Networks," to appear in *Handbook of Optimization in Telecommunications*, Kluwer Academic Publishers, 2005.

[8] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhauser, 2004.

[9] Z. Wang and J. Crowcroft. "Analysis of shortest-path routing algorithm in dynamic network environment," *ACM SIGCOMM Computer Communication Review*, 22(2):63-71, April 1992.

[10] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 756–769, December 1997.

[11] K.-C. Leung, V. Li, and D. Yang, "An overview of packet reordering in transmission control protocol (TCP): problems, solutions, and challenges," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 4, pp. 522–535, April 2007.

[12] S. Low, F. Paganini, and J. Doyle, "Internet congestion control," *IEEE Control Syst. Mag.*, vol. 22, no. 1, pp. 28–43, February 2002.

[13] G. Hasegawa and M. Murata, "Survey on fairness issues in TCP congestion control mechanisms," *IEICE Trans. Commun. (Special Issue on New Developments on QoS Technologies for Information Networks)*, vol. E84-B, no. 6, pp. 1461–1472, June 2001.

[14] M. Gerla and L. Kleinrock, "Flow control: a comparative survey," *IEEE Trans. Commun.*, vol. 28, no. 4, pp. 553–574, April 1980.

[15] J. Nagle, "RFC896—Congestion control in IP/TCP internetworks," *RFC*, 1984.

[16] C. A. Kent and J. C. Mogul, "Fragmentation considered harmful," in *Proceedings of the ACM workshop on Frontiers in computer communications technology (SIGCOMM)*, Stowe, Vermont, August 1987, pp.