ON THE IMPACT OF CONCURRENT DOWNLOADS

Yong Liu
Weibo Gong

Department of Electrical & Computer Engineering
University of Massachusetts
Amherst, MA 01003, U.S.A.

Prashant Shenoy

Department of Computer Science
University of Massachusetts
Amherst, MA 01003, U.S.A.

ABSTRACT

Concurrent downloads accelerate information access speed for individual web users. The speed-up comes from multiple connections launched for one subject download, which leads to unfairness at user level. In this paper, we study the impact of concurrent downloads on the network. Particularly, we investigate the fairness between users who employ different downloading concurrency. We also discuss how concurrent downloads affect the transient behavior of the network.

1 INTRODUCTION

The past decade has seen a dramatic growth in the number of users accessing the Internet. Web traffic accounts for a large portion of Internet traffic. Numerous studies have shown that the nonuniformity of web accesses results in hotspots of server and network load and increases the latency for information access. Several techniques have been developed to improve the latency of web accesses. Among them, concurrent downloads are end-host enhancements which open multiple concurrent TCP connections to download requested web objects simultaneously. By doing that, web clients can aggressively grab bandwidth from both network and web server, thus improve their own download latency. In the same time, the latency of other clients who utilize single connection will be degraded. Concurrent downloads also increase the chance of network congestion and server overload. In this paper, we limit ourselves to the impact of concurrent downloads on the network part. The fairness among users is investigated. We also discuss how concurrent TCP connections affect the congestion and transient behavior of the network.

The paper is organized as follows. In section 2, we briefly describe two commonly used techniques to increase download concurrency. In section 3, we show how concurrent downloads increase users' throughput in an under-utilized network. In section 4, fairness issue is investigated within an optimization framework, with the degree of the concurrency as the main parameter. In section 5, we discuss the impact of concurrent downloads on the loss and dynamics of the network. Conclusions and future work is presented in section 6.

2 HIGH DEGREE DOWNLOAD CONCURRENCY

There are two commonly used ways to increase web clients' download concurrency. As the first technique, the browser first downloads the requested HTML page from the server and then opens several simultaneous HTTP connections to download all remaining objects (e.g., images) embedded within the web page. Such parallel HTTP downloads improve the overall latency of accessing the web page and its constituent images. Most mainstream browsers use this technique, although the degree of concurrency employed can vary from one browser to another.

Recently a number of applications have been developed that claim to significantly speedup HTTP downloads (e.g., FlashGet (FlashGet 2001), Go!Zilla, ReGet, Download Accelerator, GetRight, GetSmart, Download Devil). These applications parallelize the download of each web object by opening multiple connections per object and downloading a different portion of the object on each connection (HTTP allows a byte range to be specified with each request; these programs exploit this feature of HTTP to parallelize downloads). Such applications can be integrated with browsers via plugins, making the entire process transparent to end users. This enables browsers to not only parallelize downloads of multiple objects but also parallelize the download of each individual object. The degree of concurrency employed by these applications depends on the network connectivity of end users: the faster a user's Internet connection, the larger is the concurrency employed for downloads.

3 UNDER-UTILIZED NETWORK

In this section, we are going to show how concurrent downloads improve users' throughput in a network where resource is not fully utilized. Network resource is shared among heterogeneous users. There is no explicit information about how much bandwidth is available for each individual user. As an end-to-end congestion control mechanism, TCP aims at probing and grabbing available network bandwidth and remains responsive to network congestion in the same time. But there are situations where TCP connections cannot utilize network resource efficiently. Two factors affect the efficiency of TCP: additive increase step size and maximum congestion window size. By design, TCP increases its congestion window by 1 every round trip time given there is no congestion indication from the network. If the network is under utilized, the congestion window will keep growing until it reaches the maximum congestion window size and stay there till the end of the connection or a reception of congestion indication.

It is proved in Chiu and Jain (1989) that congestion control schemes can drive the network to a fair state as long as they implement additive increase and multiplicative decrease (AIMD). The linear increase step size α corresponds to the aggressiveness of congestion control scheme in grabbing available bandwidth; the multiplicative decrease factor β corresponds to the responsiveness when facing congestion (for TCP, $\alpha = 1$ and $\beta = \frac{1}{2}$). Analysis on General AIMD (GAIMD) in Yang and Lam (2000) shows a congestion control scheme can remain to be TCP-friendly (Mahdavi and Floyd 1997) as long as α and β satisfy an equation. By using N TCP connections for one download, the user effectively sets his α to N, which enables him to grab available bandwidth more aggressively. When there is congestion in the network, this user is not "friendly" to other users who use just one TCP connection for their download. We will discuss this fairness issue in next section.

Another important factor which affects TCP performance is maximum congestion window size M. In an ideal under-utilized network where there is no packet loss, the TCP throughput of bulk data transfer is approximately $\frac{M}{RTT}$, where RTT is the round trip time of the connection. For a connection with large delay bandwidth product, M is an potential throughput degrading factor if it is not set right. For example, the default value of M in windows operating system is set to be 8KB, which means the maximum achievable throughput of a TCP connection with RTT of 1 second is only 8KB/sec, which is not good enough for download of big files. There is an extension to TCP (RFC1323 1992) addressing this issue by allowing window bigger than 64KB. Several operating systems, including Windows 2000, have incorporated this feature, by default it is turned off. Multiple concurrent TCP connections can resolve this problem easily. As long as the network remains to be under-utilized, using multiple concurrent TCP connections can increase the user's transmission rate linearly, thereby shorten the download latency perceived and improve the network resource utilization.

4 FAIRNESS

When the network resource is fully utilized, each user should back off upon receiving congestion indications from the network to avoid congestion collapse. HTTP employs TCP as its underlying transport protocol which tries to achieve fairness among competing connections. If we consider fairness at the HTTP level, users who uses a larger degree of concurrency for downloading will get a bigger share of bandwidth. To further illustrate, we discuss this fairness issue under both homogeneous and heterogeneous network setting.

4.1 Homogeneous Network

In a homogeneous network, i.e., each connection shares the same bottle-neck link, the same round trip time and the same TCP parameters, it is proved (Chiu and Jain 1989) that in steady state each connection get its fair share of the link bandwidth. It follows that an application with multiple concurrent connections traversing a bottleneck link will receive a proportionately larger share of the link bandwidth.

Assume n existing TCP flows on a congested link, two users download an object over this link, one with m concurrent TCP connections and the other with a single connection. Then the fraction of the link bandwidth available to these users is $\frac{m}{n+m+1}$ and $\frac{1}{n+m+1}$, respectively (assuming the flows are long lived and reach steady state). In the scenario where each user uses only one connection per object, their bandwidth shares would have been $\frac{1}{n+2}$ each. Hence, the bandwidth share of the former user increases from $\frac{1}{n+2}$ to $\frac{m}{n+m+1}$, whereas that of latter user decreases from $\frac{1}{n+2}$ to $\frac{1}{n+m+1}$ as a result of the concurrent download. We observe that such unfairness can result even among users using concurrent downloads. In general, any user who uses a larger degree of concurrency for downloading web objects worsens the performance for users using a smaller degree of concurrency.

4.2 Heterogeneous Network

For a network with heterogeneous users, it is no longer true that each TCP connection gets equal share of

bandwidth. We can adopt the model used in (Vojnovic, Boudec, and Boutremans 2000) and (Hurley, Boudec, and Thiran 1999) to discuss the fairness issue among different users. Let $\mathcal{L}$ be the set of links in the network, S the set of TCP connections competing for the network bandwidth and x the vector of transmission rates of all the connections. The routing matrix $A = (A_{l,i}, l \in \mathcal{L}, i \in \mathcal{S})$, such that $A_{l,i} = 1$, if connection i traverses link l, and $A_{l,i} = 0$, otherwise. For TCP connection i, the round-tip time is τ_i, additive increase step size is α_i, multiplicative decrease factor is β_i. It is proved in Vojnovic, Boudec, and Boutremans (2000) that the rates x are distributed such that x maximizes

$$F_A^h(x) = \sum_{i \in \mathcal{S}} \frac{1}{\tau_i} \log \frac{x_i}{\alpha_i + \beta_i x_i} \tag{1}$$

subject to the constraints

$$\sum_{j \in \mathcal{S}} A_{l,j} x_j \leq c_l, \ \forall l \in \mathcal{L} \tag{2}$$

This is called F_A^h fairness among competing TCP connections.

Asymptotically, if $x_i \gg \frac{\alpha_i}{\beta_i}$, i.e., small additive-increase/multiplicative-decrease ratio relatively to connection throughput, the distribution of rates x maximizes

$$F_A^{h+}(x) = -\sum_{i \in \mathcal{S}} \frac{\alpha_i}{\tau_i \beta_i x_i} \tag{3}$$

subject to (2); if $x_i << \alpha_i/\beta_i$, then the objective function is

$$F_A^{h-}(x) = -\sum_{i \in \mathcal{S}} \frac{1}{\tau_i} \log x_i \tag{4}$$

From objective functions (1,3,4), we can see the bias of TCP against connections with large round-trip time.

If every user uses just one connection for download, the fairness among users are the same as the fairness among their connections. For users use multiple connections for one download, the throughput achieved by a user is the summation of rates of all his connections. Let $\mathcal{U}$ be the set of users, with size $|\mathcal{U}| = m$. Suppose user j launches n_j concurrent connections for his download, let x_j^i be the transmission rate of its ith connection. The aggregate rate for user j is $y_j = \sum_{i=1}^{n_j} x_j^i$. Also assume all connections of one user follow the same route, define users' routing matrix $B = (B_{l,j}, l \in \mathcal{L}, j \in \mathcal{U})$, such that $B_{l,j} = 1$, if user j traverses link l, and $B_{l,j} = 0$, otherwise. Applying notations τ_j, α_j, β_j for user j, we can rewrite (1) and (2) as

$$F_A^h(x) = \sum_{j \in \mathcal{U}} \frac{1}{\tau_j} \sum_{i=1}^{n_j} \log \frac{x_j^i}{\alpha_j + \beta_j x_j^i} \tag{5}$$

subject to the constraints

$$\sum_{j \in \mathcal{U}} B_{l,j} \sum_{i=1}^{n_j} x_j^i \leq c_l, \ \forall l \in \mathcal{L} \tag{6}$$

Because of the homogeneity among connections of the same user, the optimal solution of (5) and (6) should satisfy $x_j^1 = \cdots = x_j^{n_j} = \frac{y_j}{n_j}$. Given the uniqueness of solution of (5) and (6), it is equivalent to solve the optimization problem for $y = \{y_j; j \in \mathcal{U}\}$,

$$\max_y F_A^u(y) = \sum_{j \in \mathcal{U}} \frac{n_j}{\tau_j} \log \frac{y_j}{n_j \alpha_j + \beta_j y_j} \tag{7}$$

subject to the constraints

$$\sum_{j \in \mathcal{U}} B_{l,j} y_j \leq c_l, \ \forall l \in \mathcal{L} \tag{8}$$

Comparing equation (7) with equation (1) with α_j replaced by $n_j \alpha_j$, we can see user j using n_j concurrent connections is even more aggressive then a user using one connection with increase step size $n_j \alpha_j$. This is because whenever there is one congestion indication received by the user, only one of the user's connection will back off. The loss rate for a connection is proportional to its transmission rate, so the loss rate perceived by a user is also proportional to its aggregate rate.

For the asymptotic limits (3) and (4), the optimization problems for users' rates y are

$$\max_y F_A^{u+}(y) = -\sum_{j \in \mathcal{U}} \frac{n_j \alpha_j}{\tau_j \frac{\beta_j}{n_j} y_j} \tag{9}$$

and

$$\max_y F_A^{u-}(y) = -\sum_{j \in \mathcal{U}} \frac{n_j}{\tau_j} \log \frac{y_j}{n_j} \tag{10}$$

subject to constraint (8)

It is clear from (9) that a user using n_j concurrent connections is equivalent to a user using one connection with increase and decrease element set to be $n_j \alpha_j$ and $\frac{\beta_j}{n_j}$ respectively. Thus users using more connections can take advantage of users who use less connections by more aggressively increasing their rates in congestion avoidance stage and less conservatively decreasing their rates upon receiving a congestion indication.

If all users use the same number of connections for their downloads, n_j is no longer a critical number in the overall objective function (5). Then multiple connections is not a concern of fairness anymore. But everybody uses concurrent download can increase the total number of connections within the network linearly, which in turn changes the dynamics of the network dramatically.

5 IMPACT ON NETWORK BEHAVIOR

Concurrent downloads will change characteristics of offered traffic to the network, thus change the network behavior. When a user switches from one connection sequential downloading to multiple connections concurrent downloading, it increase the number of TCP connections within the network. Notice that each connection of concurrent downloading only downloads a portion of the requested data. Consequently, the life time of each connection should be shorter the duration of one connection downloading. On the other hand, TCP is a closed loop transfer protocol. When there are many connections within the network, the transfer rate for each connection will be throttled down, as a consequence, the duration of each concurrent downloading connection will be prolonged. The download latency corresponds to the maximum duration of all the connections belonging to the download. In the extreme case, when every user uses concurrent downloading with same degree, for each connection the reduction in its transmission rate will cancel out the reduction in the amount of data transfer. Thus, no user can get improvement in his downloading latency. To make things worse, concurrent downloading may lead to big loss rate, large oscillation and poor link utilization, which in turn worsen users' performance.

5.1 Loss rate

Given a loss rate of p, a TCP connection can achieve throughput $T \propto \frac{1}{\sqrt{p}}$ (Mahdavi and Floyd 1997). Consider a simple homogeneous network, there are m users competing for total bandwidth C at a bottle-neck link. Each user launches n TCP connections. At steady state, let η be the achieved bandwidth utilization, then we have the simple equation to decide the average packet loss rate:

$$n \times m \times \frac{K}{\sqrt{p}} = \eta C \tag{11}$$

Then packet loss rate is proportional to n^2. (Note: The TCP throughput formula is only good for reasonably small p; it won't work when p is big, which is the case when there are too many connections in the network. The equation above just tries to demonstrate the impact of concurrent downloads on the packet loss, cannot be directly used to calculate change of the loss probability when users switch from one connection downloading to multiple connections downloading.)

One consequence of high loss rate is that lost packets have to be retransmitted, which degrades the good-put of each connection and the overall download latency of data blocks which consist of those lost packets. Another consequence is that when loss rate is high, TCP connections will be dragged into time out stage and shrink their congestion window down to 1 and takes much longer to recover. It causes big oscillations and poor network resource utilization.

5.2 Transient Behavior

Other than fairness in steady state, transient behavior of the network is also very important. For congestion control schemes, desirable properties include smoothness in sending rate, aggressiveness in grabbing available resource and responsiveness when facing congestion (Yang, Kim, and Lam 2001). From the network standpoint, it is desirable to make the whole system evolves in such a way that it can achieve high link utilization, small queueing delay, and small queueing delay jitter, etc. Different Active Queue Management (AQM) schemes, such as RED (Floyd and Jacobson 1993), REM (Athuraliya, et al. 2001), have been proposed to improve the network transient behavior. The parameters tuning for those AQM schemes is difficult. One of the difficulties comes from the uncertainty of network load level. Multiple connections downloads will increase the number of concurrent connections within the network. It will have big impact on network transient behavior.

To study the transient behavior, we start from a non-linear model of TCP dynamics which is derived in Misra, Gong, and Towsley (2000). Let $W(t)$ be the expected congestion window size of a TCP connection, $R(t)$ be the round trip time, which consists of queueing delay and propagation delay, $p(t)$ be the packet drop probability. Then the non-linear differential equation describing the evolution of $W(t)$ is:

$$\frac{dW(t)}{dt} = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))}p(t-R(t)) \tag{12}$$

Jumps in the sample path of individual congestion window are smoothed out by taking the expectation. When there are a large number of homogeneous TCP connections, $W(t)$ is a good approximation of the sample mean size of all congestion windows. At the steady state, we will have $W_0^2 p_0 = 2$.

In order to complete the analysis of transient behavior of the network, we need to model the dynamics of both queueing behavior and queue management schemes. For a single bottle-neck network, the queue dynamics is simply

$$\frac{dq(t)}{dt} = \frac{W(t)}{R(t)}N(t) - C \tag{13}$$

where N(t) is the number of connections at time t and C is the capacity of the bottle-neck link. To stabilize

queue length, we need to match the rate, i.e., let $W_0 = \frac{R_0 C}{N}$. By carrying out linearization (Hollot et al. 2001) of the system around its operating point $\{W_0, p_0, R_0\}$, one can obtain the control block diagram of the system with queue management scheme as the controller (see figure 1)

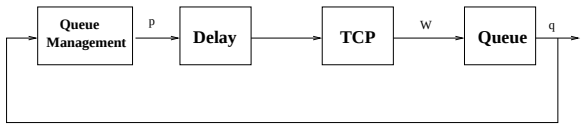

Figure 1: Control loop of the network

The transfer function of TCP is

$$P_{tcp}(s) = \frac{\frac{R_0 C^2}{2N^2}}{s + \frac{2N}{R_0^2 C}} \tag{14}$$

The transfer function of the bottle-neck queue is

$$P_{queue}(s) = \frac{\frac{N}{R_0}}{s + \frac{1}{R_0}} \tag{15}$$

From equation (14) and (15), the number of connections N is an important parameter for both the loop gain and the time constant of the open control loop, which basically decide the transient behavior of the whole system. The main topic in Hollot et al. (2001) is to give out some design guidelines of RED routers which are robust against N and R.

One counter intuitive result there is that the more connections in the system the more stable the system is. (See remark 3(5) in Hollot et al. 2001) The reason is that at steady state the congestion window size of each connection is inversely proportional to N. The impact of N on the sensitivity of TCP window size toward change of packet loss probability p at router is two fold. First, the loss event arrival rate at each connection is proportional to its current window size; Secondly, the size of back off when receiving a loss is proportional to its window size. On the other hand, the sensitivity of queue length toward expected TCP window size is only proportional to N. The overall sensitivity of the control plant $P_{tcp}(s)P_{queue}(s)$ is inversely proportional to N. The more connections in the network, the bigger the stability margin, and the slower the transient response. For designing of RED parameters, reserving a big gain margin will provide good robustness against load change.

Bear in mind, all the analysis is carried out on the linearized model, which assumes small perturbation around the system operating point and ignores intrinsic non-linearity of congestion control schemes, queue behavior and even queue management schemes. When the load level within the network has a dramatic change, for example, every user begins to use multiple sessions for their download, the system operating point will change dramatically. It won't be a surprise for those non-linearity come into play. Particularly, if the queue management at the bottle-neck link is drop-tail, because of its "famous" synchronization effect, we can't avoid non-linear analysis. In that case, if there are a large number of concurrent connections within the network, when there is no congestion, each connection increases its congestion window size by 1 each round trip time. Then queue will quickly build up at bottle-neck links. Packets will be dropped in batch. Each connection begins to back off and repeat the same cycle in high frequency. If we also take the feedback delay into account, the large number of connections will result in large over(under) shoot of the aggregate transmission rate. The network utiliation will degrade substantially.

6 CONCLUSIONS AND FUTURE WORKS

We studied the impact of concurrent downloads on the network. Users who employ higher downloading concurrency can take more bandwidth from the network, which causes unfairness at the user level. Fairness has been formulated as an optimization problem with downloading concurrency as an important parameters. Concurrent downloads increase number of concurrent connections within the network, which in turn increase the loss rate within the network and degrade the good-put of each connection. They also change the network dynamics dramatically, thus pose more challenges for network resource management.

Since concurrent downloads can improve response times, there is strong incentive for users to employ such techniques. Moreover, there is an incentive for servers to provide concurrent downloading capabilities (since users perceive better "service"), although too many such concurrent sessions will eventually harm the server performance. This problem will become more and more important. As future works, more modeling and analysis is needed for network transient behavior when there are large number of concurrent downloads. It will also be very important to investigate the impact of concurrent downloads on web servers.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation grants CCR-9984030 and ANI9980552.

REFERENCES

S. Athuraliya, V. H. Li, S. H. Low and Q. Yin. 2001. "REM: Active Queue Management", *Proceedings of the 17th International Teletraffic Congress*, September 24-28, 2001.

D. M. Chiu, R. Jain. 1989 "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks", *Computer Networks and ISDN Systems*, Vol. 17, June 1989.

FlashGet Web Page, 2001 *http://www.flashget.com*,

S. Floyd and V. Jacobson. 1993. "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. on Networking*, vol. 1, no. 4, pp. 397–413, August 1993

C.V. Hollot, V. Misra, D. Towsley, W. B. Gong. 2001 "A Control Theoretic Analysis of RED", *Proceedings of IEEE/INFOCOM*, 2001.

P. Hurley, J.Y. L. Boudec, and P. Thiran, 1999 "A note on the fairness of additive increase and multiplicative decrease", *Proceedings of ITC16*, June 1999.

J. Mahdavi, S. Floyd. 1997. "TCPfriendly unicast rate-based flow control" *Note sent to the end2endinterest mailing list*, 1997.

V. Misra, W. B. Gong, and D. Towsley, 2000. "Fluid-based Analysis of a Networkof AQM Routers Supporting TCP Flows with an Application to RED" *Proceedings of ACM/SIGCOMM*, 2000.

RFC1323. 1992 "TCP Extensions for High Performance", *IETF RFC 1323*, 1992.

M. Vojnovic, J.Y. L. Boudec, and C. Boutremans. 2000 "Global fairness of additiveincrease and multiplicativedecrease with heterogeneous roundtrip times" *Proceedings of IEEE/INFOCOM*, 2000.

Y. R. Yang, M. S. Kim, S. S. Lam. 2001 "Transient Behaviors of TCPfriendly Congestion Control Protocols". *Proceedings of IEEE/INFOCOM*, 2001.

Y. Richard Yang, Simon S. Lam. 2000. "General AIMD Congestion Control". *Technical Report TR-00-09, Department of Computer Sciences, UT Austin*, May 9, 2000.

AUTHOR BIOGRAPHIES

YONG LIU is currently a PHD student in Department of Electrical and Computer Engineering, University of Massachusetts, Amherst. He received his master degree from University of Science & Technology of China in 1997. His email address is <yonliu@ecs.umass.edu>.

WEIBO GONG received his Ph.D degree in 1987 from Harvard University. He is a Professor in the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst. He is also an adjunct professor at the Computer Science Department of the same university. He is a recipient of the IEEE Control Systems Society 1997 George Axelby Outstanding Paper Award and an IEEE Fellow. His email address is <gong@ecs.umass.edu>.

PRASHANT SHENOY received his Ph.D. in Computer Sciences from the University of Texas at Austin in 1998. He is currently an Assistant Professor of Computer Science at the University of Massachusetts, Amherst. His research interests are multimedia file systems, operating systems, computer networks and distributed systems. He is a member of the ACM and the IEEE. His email address is <shenoy@cs.umass.edu>.