Influence of Browser Type on HTTP Traffic Parameters

Vladimir Deart, Ivan Kozhuhov
Moscow Technical University of Communications and Informatics
Moscow, Russia
vdeart@mail.ru, iikorgii@gmail.com

Abstract

Following the changes from WEB 2.0 to WEB X.0 new Internet browsers provide scenarios of faster ways of getting information from servers. The scenarios are based on extension of number parallel connections and pipelining. These factors cause HTTP traffic as the bursts become bigger and the queues in routers grow. It is important to investigate the influence of browser type on HTTP traffic parameters and how it will affect on QoS because browser type defines mechanism of load optimization. For this purpose a simulation model is used based on NS2 and PackMIME. Here are discussed first simulation results.

Index Terms: WEB, Browser, HTTP 1.1, TCP-flow, burst, simulation, NS2, PackMIME.

I. Introduction

Web resource organization conception WEB2.0 was spread in modern networks. It helps users to get information access more quickly and use information put on a website by the other users. According mentioned conception it can be built portals with dynamic pages or pages with aggregated newsletters from different sites. Thus using certain WEB2.0 conceptions changes not only user’s relations to web access but HTTP-traffic structure in whole [1-3].

On the one hand, new browser’s development leads to HTTP traffic change: transfer traffic volume increasing and data loading acceleration at the expense of parallel sessions. These factors drastically change HTTP traffic parameters which we would like to estimate in our research.

A. Analysis of current situation

During last years we observe quick modernization of different browsers. One can find easily latest releases of Internet Explorer, Mozilla Firefox, Opera or Google Chrome. Among a lot of mechanisms for speeding up downloads from Internet: Cookie, HTTP Caching, HTTP-compression, DNS pre-resolution, Keep-Alive connection, HTTP pipelining, Smart parallel downloads/Multiple download threads, the last two are most popular for new browser versions. In the table 1 we can see how the number of parallel TCP-connections changes from one release to the next [4].

As it can be seen from the Table I the number of parallel sessions increases from version to version for the most types of the browsers. Also we should take into account the pipelining so both factors may enlarge bursts in HTTP flows.

TABLE I
THE TABLE OF CORRESPONDENCE BROWSER, ITS VERSION, AND THE NUMBER OF SIMULTANEOUS CONNECTIONS

Browser	Version	Max threads per host	Max threads total
Internet Explorer	6	2	34
	7	2	55
	8	6	35
	9	6	35
Mozilla FireFox	4	6	30
	5	6	30
Safari	5	6	30
	6	5	5
Opera	9	4	20
	10	8	30
	11	8	32
	Mini 5	11	30
Google Chrome	9	6	35
	10	6	35
	11	6	35

II. SIMULATION

A. Simulation model

HTTP-traffic simulation model was built with use of NS2 simulator and PackMIME module [5, 6]. To consider mechanisms for speeding up downloads used in modern WEB-browsers the PackMIME module was modified. In the source code of the module was added the algorithm, in accordance with which between two adjacent primary HTTP-sessions introduce secondary session.

The modeling network topology is presented on the fig. 1. The local network is presented by multitude clients simulating data exchange from Web servers. All users present in access network have equal channel throughput to router R1 and single service Web traffic priority. Every client simulates the user's work with Web browser. The Internet is simulates as Web-servers placed in the network. Every Web-server replies by HTTP response for the coming request independently which client had sent it.

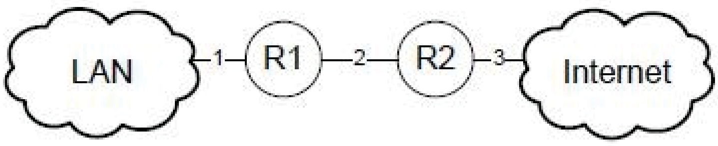

Fig. 1. The modeling network topology

Quantity of clients and servers in each network were independent and not defined beforehand. Client and server multitude in simulation model are made by PackMIME component assistance and NS2 makes use of them in TCP session organization. Thus, there are no relation between HTTP objects and pages in simulation model: all downloaded by user data are bind only with defined TCP session.

Quantitatively the client multitude in the LAN network is defined with TCP session intensity. Every session is made for appointment data volume and contains one or more GET requests. All appointment for transmission into session data by TCP protocol should be delivered from the server to the client. In case of data packet loss the retransmission occurs.

The analysis of Web traffic transmission throughput shows that the service quality is maximal influenced by the channel 2 capacity between Core (R2) and border (R1) router. Therefore the channel 1 throughput in simulation model which contact Web clients and border router was determined equal 1 Gbps (channel 1), that is much more than general channel 2 throughput. In the basic simulation model the throughput of channel 2 is 100 Mbps, and it changed so that it can be possible to obtain different utilization. The channel 3 bandwidth is also 1 Gbps.

HTTP traffic has an asymmetric character: to the Web servers direction users send the official packets only signalized about TCP session installation and about successful data packet reception. User transfer effective load to Web server direction consists of GET requests only with average sizes 650-700 byte [6]. Then for every GET request servers send HTTP response to user with size 8-10 KB [6]. It means that traffic from WEB servers to users is 10 times more than in opposite direction.

We originally focus on the traffic taking into account the burst arrival process. As it shown in [6] intensity of organization of TCP-sessions for the transfer HTTP-data is determined by two dependent distributions: exponential and uniform. The presence of the exponential distribution is explained by a generalization of the traffic from a large number of users, whereas, the presence of a uniform distribution caused by the work of a web browser.

The simulation is done for downstream HTTP traffic coming from servers to users. The size of buffers in Routers considered as 100 packets and FIFO queues was used. So IP packets will be dropped if overflow of buffer occurs.

B. Simulation results

For investigation of the influence of multiple TCP connections we used the model [6]. There was a research of influence on mean packet delay from different ratio secondary flow: 1/1 and ½ [7]. Therefore ratio ½ means that for each second main session opens one secondary session.

In our work we used the modified PackMIME module to introduce different number of parallel sessions: 2, 4, 6 and 8. Therefore 2 parallel sessions means that for each main session browser opens two secondary sessions.

We assume that initial requests are made by user clicks and have an exponential distribution of intervals between clicks. Each click initiates first TCP connection with the server. When browser had received first reply from the server it starts parallel connections with very small intervals between them. In general, such procedure may increases bursts in TCP flow.

On fig. 2 mean packet delays are shown according utilization and number of parallel connections (ratio of the secondary flow). As it can be clearly seen from figure 2 the increasing of number of parallel TCP connections provides decreasing of mean packet delays.

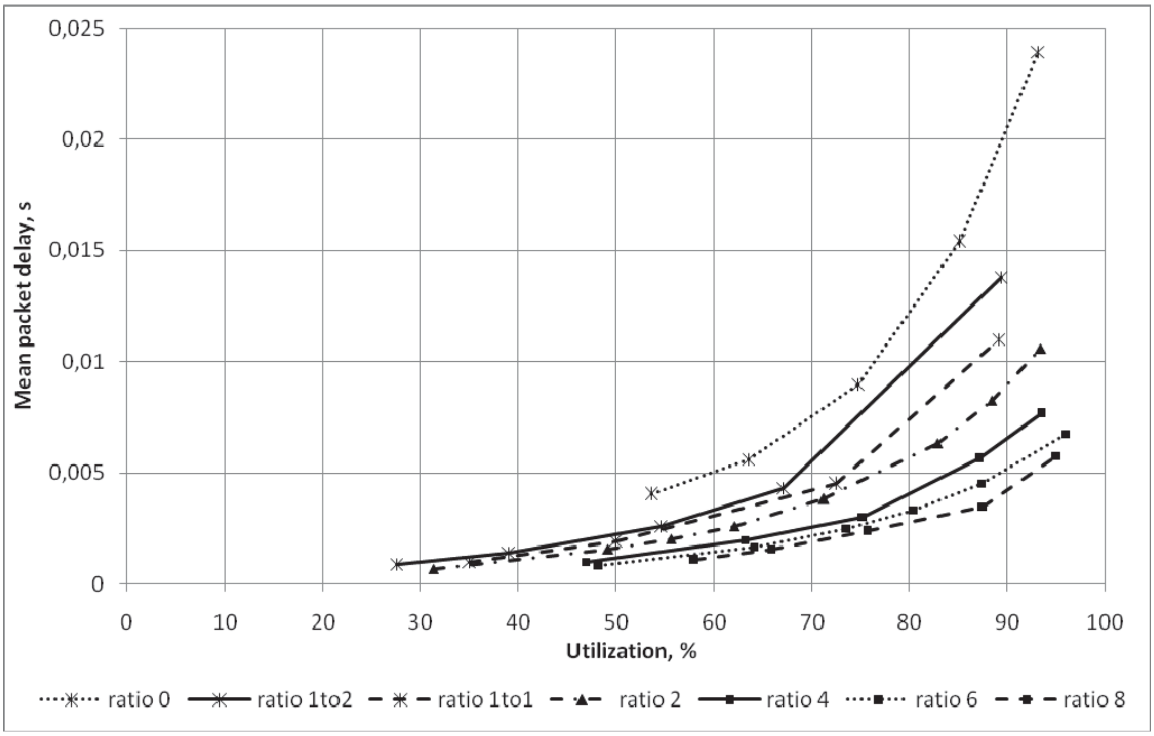

Fig. 2. Mean packet delay according number of parallel TCP connections

III. Conclusion

All new browsers use multiple parallel TCP connections to one server (host). The number of parallel connections is about from 4 (IE) up to 11 (Opera). The parallel connections may have a great influence on HTTP traffic parameters making packet's bursts much bigger. However from the simulation results can be seen that for utilization ($\rho > 80\%$) the increase of multiple connections leads to some decrease mean packet delay. The investigation should be proceeded for more detailed results.

References

[1] J. Padhye, V. Firoiu, D. Towsley, J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation // In proceedings ACM SIGCOMM Computer Communication Review, Volume 28, Issue 4. 1998. – P. 303-314.

[2] K. Fall, K. Varadham. The ns Manual // The VINT Project documents. – Berkeley, USA, 2009. 430 p. Website: http://www.isi.edu/nsnam/ns/doc/index.html.

[3] L. Shuai, G. Xie, J. Yang. Characterization of HTTP Behavior on Access Networks in WEB2.0 // In proceedings ICT 2008. June, 2008. – P. 1-6.

[4] http://www.browserscope.org/.

[5] M. Wiegle. PackMime-HTTP: Web Traffic Generation in NS2 // http://www.isi.edu/nsnam/ns/doc/node552.html. 2007.

[6] V.U. Deart, V.A. Mankov, A.V. Pilugin. HTTP Traffic Measurements on Access Networks. Analysis of results and Simulation // In proceedings of The 9th International Conference on Next Generation Wired/Wireless Advanced Networking. – Berlin, Germany.: Verlag, 2009. – Vol.5764. – P.180-190.

[7] A.V. Pilugin, V.U. Deart. HTTP Traffic Model for Web2.0 and Future WebX.0 // International Journal of Wireless Networks and Broadband Technologies, 1(1), January-March 2011 -La Trobe University, Australia, IGI Global. - P.50-55.