Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium, Volume 22, No. 1, ISSN 1726-9679
ISBN 978-3-901509-83-4, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2011
Make Harmony between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals & Proceedings of DAAAM International 2011

QUALITY OF SERVICE IN MULTIMEDIA COMPUTER NETWORKS

MACURA, A[rijana]; MISSONI, E[duard] & MAKOVIC, B[ranko]

Abstract: *Implementing a specific algorithm for synchronization based on a selection of relevant QoS (Quality of Service) parameters requests their detailed analysis in order to enable realization of the desired multimedia applications. Ensuring of requested QoS presents one of the important elements of the design and implementation of multimedia networks. The paper analyzes the parameters of importance for multimedia communications. In recent years there has been a rapid increase in multimedia traffic in computer communication networks and the demand for quality of multimedia content transfer services for the users. In response to such requests numerous mechanisms have been developed that enable services for allocating priority of traffic in communication networks, which makes it possible to determine the quality of service.*
Key words: *algorithm, TCP/IP, reno, tahoe, SACK*

1.INTRODUCTION

Implementing a specific algorithm for synchronization based on a selection of relevant QoS (*Quality of Service*) parameters requests their detailed analysis in order to enable realization of the desired multimedia applications. Ensuring of requested QoS presents one of the important elements of design and implementation of multimedia networks. Quality of service is the ability of network to provide better service to certain types of network traffic. This ability can be achieved with different network technologies, as well as in the layers of different network protocols (usually TCP/IP protocols).TCP protocols makes possible a reliable transfer of connection type. Primary TCP would initiate connection by manifold transmitter segments pricking to the network up to the window size notified by the receiver. That is acceptable when two hosts are in the same LAN, but in the situations when there are routers and slower links between the transmitter and receiver problems can emerge. Particular mediating routers must create waiting files with packages in such a way that mentioned router remains without free space for the temporary location of the package. In order to avoid that problem, it is necessary to develop *slow-start* algorithm. Congestion can happen when manifold incoming flows arrive on router whose outgoing capacity is smaller than the sum of all entrances. For that reason algorithms are developed to avoid congestion. Quality of service in computer network traffic is the ability of communication and transmission devices to guarantee the minimum requirements, in other words, quality of traffic required by some network application. TCP is based on principle of package “preservation”, which means if connection works on available capacity of the permission size, in that case package is not thrown in net before the other package leaves the net. As distributed multimedia systems grow larger and faster, the associated problems of resource management become harder to manage. The traditional method of managing applications is that, prior to execution, an application determines its quality of service (QoS) requirements in terms of various parameters such as image and sound quality.

2. FUNDAMENTALS OF IMPROVING MULTIMEDIA DATA TRANSFER IN TCP/IP COMPUTER NETWORKS

Frequently used or suggested improvement solutions for multimedia data transmission quality over TCP/IP computer networks are divided into four main categories.

2.1 Solving big bandwidth-delay product

Most solutions in this area have emerged because of the necessity to solve the problems of older TCP/IP implementations. In the older implementations there is a problem of solving big bandwidth-delay product because of a small congestion window parameter, a slow sliding window algorithm and also slow recovery time from traffic congestion. Also, the overall scalability of the transfer depending on the number of simultaneous data flows is one of the most important areas for improving the TCP/IP algorithm. Some proposed changes include increased initial TCP window size, changes in sliding window algorithm, and the introduction of *selective acknowledgment* (SACK) that allows proper adaptation of the TCP source of network traffic in case of loss of more network frames within one RTT (Round Trip Time) period (Fall & Floyd, 1996). Given that the increase of "congestion window" parameter is based on increase of the number of frames and not on frame length (in Bytes), therefore it is recommended to know beforehand the maximum frame size on any given segment or connection. Usage of the maximum permitted frame size affects the delay, but it also maximizes the usage of network bandwidth by reducing the impact of header - overhead. TCP flow and congestion control algorithms present the problem for the transmission of multimedia data due to the basic problem that the connection constantly increases network load until it reaches the level of congestion, after which the load rapidly decreases, and then the process repeats in iteration. Load (bandwidth) variation and the network congestion example can be shown with case of two flows of data that use the same portable algorithm (e.g. TCP Reno), with the same network delay and transmission initiation's beginning. Also, given that in this situation the total of network's bandwidth is is used, and the data flow with the bigger delay does not have enough resources for normal traffic, both TCP Reno flows will continue to compete for the available bandwidth, steadily bringing the network into a state of congestion (Casseti & Meo, 2001). This behavior of TCP Reno implementation shows considerable intolerance toward connections with bigger delays in the network (Gao & Zheng, 2001). Figure 1. shows the comparison of the scalability of the main implementation of the TCP protocol. The effect of such behavior on multimedia communications can be extremely negative, resulting in a very low quality of video and audio streams in a network that has sufficient capacity to support them (in terms of bandwidth, delay, or variation in delay). Therefore it is necessary to find a better algorithm for detection and control of congestion inside protocol. Given that the framework for the protocol allows the

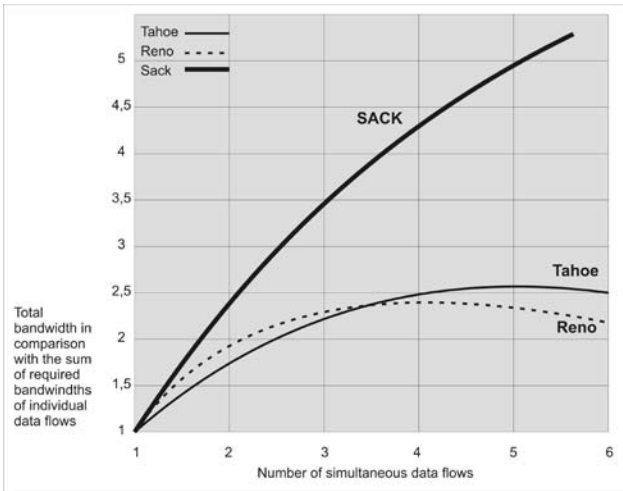

Fig. 1. Scalability of different TCP implementations that depend on the number of concurrent data flows

introduction of different algorithms for conduct in a state of congestion , it is possible to increase efficiency and intelligence of TCP by implementing new algorithms with adaptive behaviors. TCP Tahoe algorithm for avoiding congestion utilizes "additional increase and manifold decrease" (AIMD). Loss of package is considered as a sign of congestion and Tahoe records half of temporary window as the limit value, and then starts *slow- start* till the limit value is reached. After that, increments line-ray till the loss of package happens. In such a way the window size slowly increases till it comes nearer to the full size capacity. Disadvantage of Tahoe algorithm is the need of complete interval of time limitation to detect loss of a package. TCP Reno is keeping back the basic principles of Tahoe algorithm, such as *slow- start*, but detects earlier lost packages and flow structure is discharged each time a package is lost. Reno requires getting confirmation as soon as segment comes to destination. Reno suggests algorithm named *fast-retransmit* which decreases transmitter waiting time before lost segment retransmission. Reno behaves well when there are small packages loses. But, when we have manifold packages loses in the same window, its performances are approximately the same at those of TCP algorithm at the same conditions.TCP SACK with "selective confirmations" is widening in relation to TCP Reno and works on problems which TCP Reno meets, especially on detection of manifold lost packages, as well as retransmission of more than one lost package through RTT.

SACK keeps *slow- start* and *fast- retransmit* parts of TCP Reno. It has the same long lasting time limitations as Tahoe, in case that the package loss has not been detected by the modified algorithm. TCP SACK required segments not to be confirmed cumulatively, but selectively. The biggest problem with SACK is that momentary selective confirmations are not supported by transmitter. SACK implementation or better selective confirmations is not an easy task, but based on simulation results can be concluded that TCP SACK shows the best results. No one of the proposed versions realize the best results in all circumstances. (Ceco & all., 2010). SACK behaves like TCP Reno when the transfer takes place normally, or when there is a loss of one network frame within one transfer window (Floyd & all., 2000). However, when it comes to losing more than one network frame, SACK allows faster recovery and greater achieved bandwidth. SACK achieves better performance in all cases of transfer, with a score of over 95% bandwidth utilization with a small amount of losses. Also, in case of losing more than one network frame within one transfer window, SACK implementation shows a faster recovery and better bandwidth than Reno implementation. TCP SACK algorithm made minimal changes to TCP Reno implementation. The key improvement is achieved by allowing selective acknowledgement of received frames, and variable setting, by which it tracks the remaining number of frames in transit through the network. In this way allowing the algorithm to be undisturbed even more frames are lost in the network within one transfer window. Dependence on measuring network frame's round trip time (RTT) as the main parameter for increase of data flow bandwidth in Reno and SACK implementations is responsible for the preference of data flows with smaller delay, in other words, RTT time.

TCP Tahoe algorithm's specific improvements are related to the mechanism of detection and estimated network frame's round-trip time through the network Also, *fast-retransmit* algorithm built-in the Tahoe implementation was later modified in modern implementations. If you disable this function, TCP Tahoe transfer algorithm will take considerable time to detect network frame loss and then send it back to the network, which means that there will be decrease in transmission quality if only one network frame gets lost. When frame loss gets detected, transmission will continue with reactivation of *slow-start* algorithm, which means that there will be a gradual increase in transmission bandwidth.

3. CONCLUSION

Troubleshooting QoS management in multimedia computer networks is based on discovering the efficient allocation of network resources, with realization of QoS requirements and a high degree of resource utilization. Different applications have different requirements for their data transmission means. Applications will generally expect that the network can transmit data at the speed at which they are generated. In addition, applications show various sensitivity to many other transmission parameters, such as delay and delay variation in the transmission. Also, some applications are more sensitive than others to the problem of packet loss, in other words, data fragments in the transmission. These network applications' characteristics are expressed through the parameters of capacity, delay, jitter and loss rate. Priority and service quality allocation mechanism must be ensured to meet the communications requirements of some applications, to reserve the free capacity of computer network. The future researches will continue in direction to discoveries of new mechanisms and creation of new version of TCP protocol aiming in combination of existing versions of TCP protocols in joint implementation, which would have superior performances.

4. REFERENCES

Casetti, C. & Meo, M. (2001). Modeling the Stationery Behavior of TCP Reno Connections, *Proceedings of the International Workshop on Quality of Service in Multiservice IP Networks*, ISBN:3-540-41512-2, pp. 141-156, January 2001.

Ceco, A.; Bradic, K.& Nosovic, N. (2010). Performance comparison of different TCP versions, *Available from*: http:// www.telfor.rs/files/radovi/TELFOR 2010-10-23.pdf *Accessed*: 2011/03/07

Fall, K. & Floyd, S. (1996.) Simulation – based Comparisons of Tahoe, Reno and SACK TCP. *ACM SIGCOMM Computer Communication Review*, Volume 26, Issue 3, July 1996., ISSN:0146-4833, ACM New York, USA

Floyd, S.; Maldavi, J.; Mathis, M. & Podolsky, M. (2000). An Extension to the Selective Acknowledgement (SACK) Option for TCP, *Available from*: http://www.icir.org/floyd/talks/dsack-nov99.pdf *Accesed*: 2011/03/11

Gao, C.; Li, M. & Zheng, F. (2001). An Analytic Throughput Model for TCP Reno, *Proceedings of the 2001 International Conference on Computer Networks and Mobile Computing*, ISBN 0-7695-1381-6, pp. 111, October 16-19, 2001