A STUDY ON SCALABLE HIGH-PERFORMANCE VIRUS

[1] N.RAJA RAJAN. [2] L.DINESH
[1] Dept.of.Computer science and Engineering ,St'joseph College of Engineering,Elupatti,Thanjavur
[2] Asst.Professor,school of Computer science and Engineering ,St'joseph College of Engineering,Elupatti,Thanjavur

ABSTRACT--*In proxy Generation Wireless Networks (NGWN), lots of information, such as voice and video data, will be used in All-IP networks. It is important to note that in such environments, seamless service for users and handoff between heterogeneous networks must be taken into account. Therefore, in this paper, we propose a SePH (Seamless Proxy based Handoff) using PMIPv6-based proxy model, which is able to improve the performance of handoff in NGWN. The SePH can efficiently support seamless and IP-based mobility, by reducing the search process. The performance results show that our proposed scheme outperforms in terms of Quality of Service (QoS), such as throughput, handoff latency, packet loss, and signaling overhead, comparing to the existing schemes. These performance measures are used in an optimization problem that is formulated to determine the optimal number and optimal location of proxy servers problem. Based on a comparison with a genetic algorithm, it can be concluded that algorithm produces near-optimal to optimal results in a very short time.*

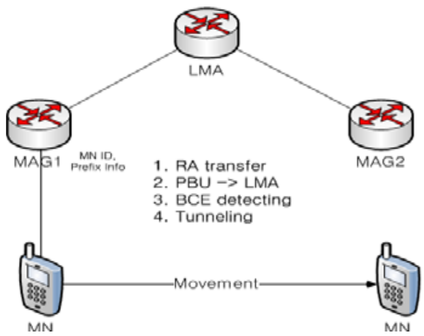
Keywords: ***Scalable,Wireless.***

I. INTRODUCTION

Overview of the current web client web proxy - web server mechanism and takes a deep look into one of its main disadvantages: the replication, in the proxy's cache, of web objects having different URL but the same content. This problem is known as the "Duplicate Transfer" problem and is mainly caused by the current mode of indexing web objects based on their URL, which is used as a primary key in the cache repository. We present in this paper a statistical analysis based on real traffic measurements, which shows that more than 10% of a proxy's cache consists of replicated objects, grabbed from the Internet in a useless manner and stored redundantly at least twice. These results urge the development of a scalable real life solution to the duplicate transfer problem: some solutions have been previously proposed, but never deployed on a large scale in Internet. The dominant traffic types in the current Internet fall in two main categories that take approximately equal shares of the global Internet traffic: the web traffic and the peer to peer (P2P) traffic. Although the latest one has taken a serious advantage beginning with the late 90s, more recently the web traffic also claimed its share and today it exceeds any other form of Internet traffic, including P2P traffic.

This come-back situation is mainly due to the following two aspects:
a) The legal battle against the illegal aspect of sharing multimedia content files transported over peer to peer protocols;
b) The migration of software development from desktop and standalone applications towards web-based applications.

Seamless Proxy based Handoff

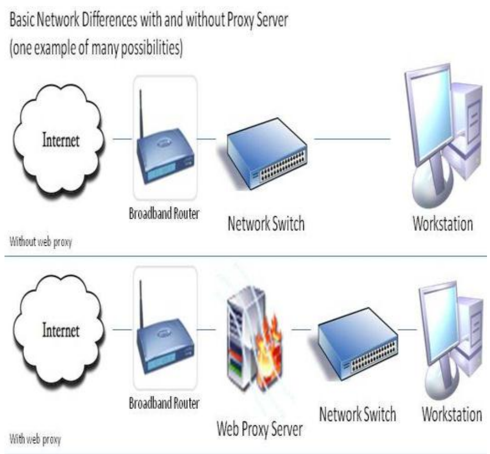


Once the web server completes processing a client's request, it responds to the client by sending the processed information back to the client and then it starts processing the next request. The client processes the piece of information that it received from the web server and when the processing is complete, the client makes the next request to the web server. This is the finite population queue mode Based on the current implementation of the HTTP protocol, we assume that all clients send only one request at a time and wait for a response before sending out the next request.

Duplicate Transfer Problem Proxy servers:

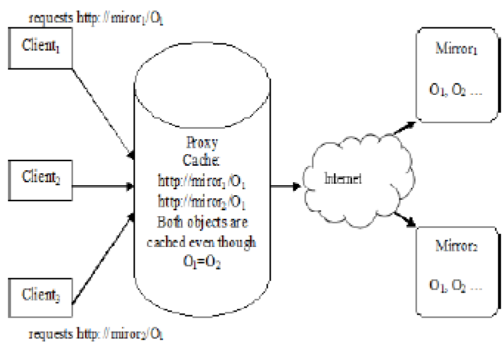
International Journal of Advanced Research in
Computer Science Engineering and Information Technology
Volume: 3 Issue: 1 26-Jun-2014,ISSN_NO: 2321-3337

A real scenario that covers the above formal description follows. Our Computer Science Department hosts a series of Linux workstations labs. Periodically, all these Linux workstations execute an automatic update process, which implies retrieval of any available updates from the Internet.



Performance criteria and scalability for a single web server

An update is a web object located in a specific web repository. For load balancing reasons, the updates repository is replicated in multiple mirrors, each mirror being accessed With a different URL. This fact implies that the same updates, located in repositories having different URLs, are treated by the proxy server as different objects, even if they
have the same content.



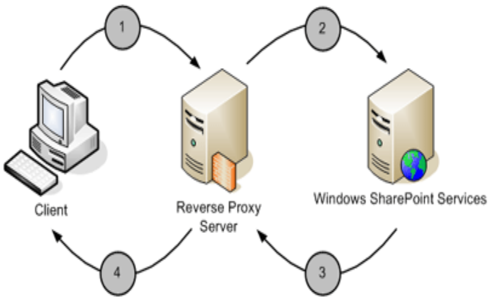
Disadvantages:

The disadvantages of this behavior are obvious. First of all, any web client, starting with the second one, that accesses a certain update, might wait for the update being retrieved from the Internet even if the update is already in the proxy server's cache, located in a location that is faster to access. Secondly, multiple retrievals of the same objects lead to wasted precious network resources such as bandwidth. And finally, there is a waste of storage space at the proxy server level, because objects having the same content are cached more than once. Consider an organization or company that would like to guarantee QoS in terms of throughput and delay for all its users accessing the company website.In order to do so, this organization would have to set up (based on the demand forecast) several web servers across the Internet.The optimization problem that we consider essentially determines whether or not to install a web server at each of the potential locations. We first explain the problem setting and then follow up with some useful notation before stating the optimization problem.

Solutions to the Duplicate Transfer Problem

The solution of the above problem is identifying and indexing web objects at the level of a proxy server by their MD5 checksum. This mode of identifying and indexing web objects is not to replace the classical one where web objects are identified and indexed by their source URL. It may be used as an alternative method for maintaining web objects at the level of a proxy server, especially web objects of a considerable size.

The delivery of the MD5 checksum to the client, if it is available at the level of the web server, might be an implicit or an explicit process In order to deliver the checksum the interested clients, this checksum must be available at the web server's level. There are multiple approaches for storing and computing the checksum for a web object: thesis is run-once process, because the checksum information can be stored in the web server memory cache and may be subsequently delivered to any other client that might request that web object again.



CONCLUSIONS

This paper proposes SePH, proxy-based seamless handoff technology, for mobility management protocol in a heterogeneous network, in order to provide better network performance in the PMIPv6-based heterogeneous wireless environment. The proposed SePH protocol uses the IHDA algorithm, based on score functions. The purpose of IHDA is to ensure seamless roaming, service continuity, and reduce service disruption for handoff required in 3G/4G/NGWN. The outcome of the proposed protocol's performance evaluation reveals that SePH has greater performance than existing protocols in terms of handoff delay time, throughput, packet loss rate, and other costs. we considered the problem of optimally allocating web servers on a rectangular grid. The problem is formulated as a mathematical program with the objective of minimizing the cost of setting up and maintaining the web servers subject to satisfying demand and QoS constraints developed an algorithm called DEJAVU to solve the optimization problem. We compared the DEJAVU algorithm against a genetic algorithm. We showed that for 80% pf the problem instances, the gap between the DEJAVU and genetic algorithm was less than 5% and the average gap over all the problem instances was only 3.68%. Since the DEJAVU algorithm takes less than a minute to solve (as opposed to several days for the genetic algorithm), it would be a suitable option to solve the optimization problem.

REFERENCES

[1] Chieh-Jen Cheng, Chao-Ching Wang, Wei-Chun Ku, Tien-Fu Chen , and Jinn-Shyan Wang, "Scalable High-Performance Virus Detection Processor Against a Large Pattern Set for Embedded Network Security" Commun. vol. 51, pp. 62–70,2011.

[2] O. Villa, D. P. Scarpazza, and F. Petrini, "Accelerating real-time string searching with multicore processors," Computer, vol. 41, pp. 42–50,2008.

[3] D. P. Scarpazza, O. Villa, and F. Petrini, "High-speed string searching against large dictionaries on the Cell/B.E. processor," in Proc. IEEE Int. Symp. Parallel Distrib. Process., 2008, pp. 1–8.

[4] D. P. Scarpazza, O. Villa, and F. Petrini, "Peak-performance DFA based string matching on the Cell processor," in Proc. IEEE Int. Symp. Parallel Distrib. Process., 2007, pp. 1–8.

[5] L. Tan and T. Sherwood, "A high throughput string matching architecture for intrusion detection and prevention,"in Proc. 32nd Annu. Int. Symp. Comput. Arch., 2005, pp. 112–122.

[6] S. Dharmapurikar, P. Krishnamurthy, and T. S. Sproull, "Deep packet inspection using parallel bloom filters," IEEE Micro, vol. 24, no. 1, pp.52–61, Jan. 2004.

[7] R.-T. Liu, N.-F. Huang, C.-N. Kao, and C.-H. Chen, "A fast string matching algorithm for network processor-based intrusion detection system," ACMTrans. Embed. Comput. Syst., vol. 3, pp. 614–633, 2004.

[8] F. Yu, R. H. Katz, and T. V. Lakshman, "Gigabit rate packet pattern matching using TCAM," in Proc. 12th IEEE Int. Conf. Netw. Protocols, 2004, pp. 174–178.intrusion detection system," ACMTrans. Embed. Comput. Syst., vol. 3, pp. 614–633, 2004.

[9] R. S. Boyer and J. S. Moore, "A fast string searching algorithm,"Commun. ACM, vol. 20, pp. 762–772, 1977.

[10] V. Aho and M. J. Corasick, "Efficient string matching: An aid to bibliographic search," Commun. ACM, vol. 18, pp. 333–340, 1975