International Journal of Research in Advent Technology (E-ISSN: 2321-9637) Special Issue
1st International Conference on Advent Trends in Engineering, Science and Technology
"ICATEST 2015", 08 March 2015

Resource Scheduling Techniques to Improve Cloud Performance Based on Client

S. Jayanthi *[1], Dr. P. Srinivasa Babu [2]
[1] *M.E Scholar, Department of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur,*
[2] *Professor, Department of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur,*

Abstract: Infrastructure as a Service (IaaS) in cloud provides ample scope for various high volume applications to be run on the servers across the WAN, availing a fair service to the end clients. Many effective schedulers have been designed to consider the contention of the computational and communicational resources, which provide a guaranteed effectiveness for resource sharing. However, the vast diversity of client devices in a cloud demand scheduling based on their features and capabilities. Mobile clients, workstations, laptops, PDAs and thin clients in the cloud vary in aspects such as processing power, screen size, battery life, geographical distance and many. Algorithms in the cloud, which are based on client capabilities result in many consumer benefits such as network load balancing, saving battery, reducing latency, efficient processing. In this paper, the authors propose a client aware credit scheduler for virtualized server setup in a cloud that schedules the client requests based on the client device features and capabilities. Rich Internet Applications (RIA) is proposed, in order for the server to realize the client device capabilities such as the type of client, the battery remaining and the location of the client. Results show that the client-aware credit scheduler is effective in terms of saving energy

Keywords – Virtual Machines, Cloud computing, schedulers, clients, Cloud Service Providers.

1. INTRODUCTION

Cloud computing is on its way to forming new trends in the IT industry. Cloud Computing uses the Internet to deliver services to the recipient. The recipients are charged on the basis of service rendered. Rather than buying and maintaining own servers, software, data centre space, and network equipment, the Infrastructure as a service (IaaS) provides users the abstract of computer infrastructures such as storage, processor, and networking as virtualized service over the cloud. Clients buy these resources as a fully outsourced service, which are maintained at another end of the WAN. Cloud services are resourcefully utilized only when virtualization takes its place on the real servers. In an enterprise cloud environment, the Cloud Service Providers (CSP) accepts various requests from the end clients in the form of application services. The end clients depend upon the CSP to compensate resource deficiencies and also to reduce the workload. This type of resource utilization brings in the latencies that could be incurred upon the client. In order to reduce the cost of ownership for deploying the number of physical machines to serve multiple service requests, CSP uses virtualization with multiple Virtual Machines (VMs) installed on servers. Hence, virtualization was proposed to consolidate servers that had various physical machines requiring multiple operating systems. These Virtual Machines, which run on the emulated software or hardware virtualization, are maintained by Virtual Machine Monitors (VMM), which runs through the server using independent configurations of operating systems, drivers, and software-- providing multiple services on a single physical host machine [1].

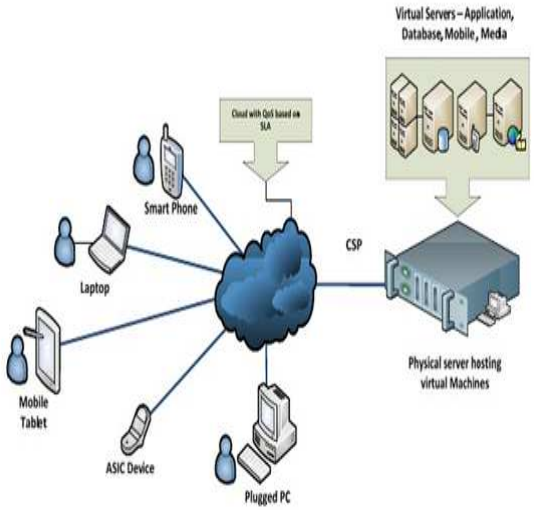

Figure 1.Cloud server

With virtualization in effect, VM's can be used create virtual entities of modules such as storage, network, software and hardware resources under each abstract partitions of VMs as shown in Figure 1. Some of the advantages that make virtualization very effective are listed below.

Consolidation: Workloads of multiple servers can be consolidated into a single physical machine, which guarantees cost effectiveness.

Heterogeneity: Each VM has its own operating system and they can be functional at one time on a single physical machine, which increases efficiency and also implicates compatibility with different applications.

Migration: Redundancy and load balancing is possible because VM can migrate from one physical machine to another using the concept of mobile software compartment.

Isolation: Each VM keeps its resources to itself behaving like a standalone server.

Server virtualization is categorized based on hardware built as Full Virtualization, Partialn Virtualization, and Para Virtualization. They differ in the simulations and interactions of the operating system with the physical resources on the server.

The cloud encompasses a vast diversity of client devices. Client devices such as mobiles phones, workstations, laptops, PDAs, thin clients, iPad, differ in attributes such as processing sensitive workloads, Power, battery life, screen size, geographical distances and so on. It also becomes important to ensure that the requests from the clients are scheduled based on their attributes for fairness. For example, a client that has low battery remaining calls for higher priority in scheduling when compared to a client that has enough battery life. Rich Internet Applications (RIA) has enabled servers to realize the capabilities of client devices and has driven the deployment of client aware technologies. With RIAs, application codes download to the client device and execute on it using a RIA software framework, which is typically based on HTML5. RIAs may be used to infer the hardware properties of the clients such as processing power, battery life, network bandwidth and other properties [2].

Several enhancements have been proposed for the cloud, which improves the response time for latency However, not many schedulers are proposed for the cloud, which take into account the client device attributes and capabilities at servers, when serving the client requests. Client aware scheduling is very important for the design of cost effective cloud architecture because of the wide variety of clients available and their limitations in various features. Especially because requests in a cloud travel over a WAN, it is important not only to ensure a low latency of service but also to schedule the requests based on the client capabilities.

In order to address the above challenges and issues, the authors in this research propose a Client Aware Credit Scheduler (CACS) for virtual servers in a cloud. The proposed scheduler is a variant of the default credit scheduler, in which the type of client, battery life and distance of the client from the server attributes are used for scheduling the requests.

2. LITERATURE SURVEY

One of the Cloud Computing forums describes Mobile Cloud Computing (MCC) as an infrastructure in which both data processing and the date storage move the computing power and data away from the mobile platform to the cloud with servers capable of servicing large numbers of subscribers [19].

In [20] Hoang T. Dinh et al studied the importance of MCC and explained how battery consumption on a mobile device is reduced by moving data processing from mobile or to high performance servers. Similar findings were done by A. Rudenko et al in [21] specifically on laptops. Based on evaluations, both studies show that 45% of battery consumption is reduced by opting to use remote process execution.

A. Carroll in [22] also performed a detailed analysis on power consumption by concentrating in the area of Smart phones. Component-wise breakdown has been done to measure the power consumption and has been tested with micro-benchmarks, which are built realistic scenarios. The author showed how these different usage scenarios translate into overall power consumption and consumption of battery life.

Y. Liu and co in [23] performed an empirical evaluation on the battery power consumption of the mobile devices for the date streaming applications. The authors analyzed the most frequently used streaming protocols used in the internet for streaming data. They show that chunk-based streaming was more power efficient because of its adoption to PSM traffic shaping technique. They also show that P2P streaming is not efficient since it adds load by re-uploading and transmitting control traffic.

The authors in [24] enhanced the client-aware technologies and apply them to desktop virtualization and cloud by inferring main advantages from the end-point device capabilities.They approach the objectives by featuring a private cloud that has the ability to infer the device attributes and capabilities and to adapt services accordingly. They emphasize on creating a balance between client-hosted virtualization and server-hosted virtualization by implementing client-aware technologies like workload shifting, multimedia redirection, command remoting, reverse seamless technology, and rich internet applications.

D. Ongaru et al in [15] have extensively studied the Credit Scheduler of the Xen hypervisor and proposed few enhancements, which effectively improved the scheduling performance of the domains. The improvements include sorting the run queue by boosting the I/O handling domains giving priority over the other. They also propose a mechanism to reduce the pre-emption of the event channel notifier improving the overall latency and performance aspects.

G. Liao et al in [25] have demonstrated some software techniques to improve the virtualized I/O performance in multi-core systems. They state that the extra driver domain for I/O requests processing and the scheduler for the VMM are responsible for the degradation of the I/O performance. To reduce these overhead they propose to solutions – a cache aware scheduling to reduce the inter domain communication latency and a mechanism to steal the credits from other domains so they can allocate them to the I/O intensive VCPUs.

Saransh Mittal in [6] has studied the application patterns in the virtualized environment and proposed a mechanism to do quantitative based network scheduling. The author combined the working of the CPU credit scheduler in Xen and DWRR scheduler used in routers to develop a mechanism and he implemented on the network backend drivers. Evaluations show that QoS requirements are fulfilled because of the corresponding bandwidths allocated for respective application.

In [26] S. Govindan et al have developed an algorithm to improve the performance of the virtualized based hosting platforms to enable satisfactory application performance even under conditions of high consolidation, while still adhering to the high-level resource provisioning goals in a fair manner. They develop the algorithm to schedule the CPU of VMM that considers the I/O behavior of the VMs— creating an unfair advantage to communication intensive applications over CPU intensive ones.

3. PROBLEM DESCRIPTION

Although the default credit scheduler provides fairness of CPU to its domains, it is not suitable for client-aware scheduling. In [15], the authors added the BOOST state so that an I/O domain can start executing immediately without having to wait for other domains to execute. This prioritizes the I/O requests when compared to processor intensive requests. In an IaaS and PaaS cloud, most of the requests will be I/O and will need to access data from the disk arrays. In such a case, the BOOST state will not be efficient.

In a cloud, since the requests traverse through the WAN, certain client device capabilities like battery remaining, distance from the server, and processor utilization need to be taken into account. The default credit scheduler is more focused on providing fair CPU resources to the virtual servers and it does not prioritize the requests.

Hence, the authors propose a client-aware credit scheduler that schedules the domains according to their priorities, which is based on the device capabilities. The task of priority assignment is offloaded to Domain 0, which is the driver domain. From there, the driver domain decides the priority of a request according to the client device capabilities. It must be noted that priorities are assigned to requests present in the same queue as they were added by the cloud middleware. Therefore, requests within the High, Medium, and Low priority queues are assigned to their respective queues and those queues are serviced separately. This ensures that the services and the QoS according to the SLA are still met.

4. PROPOSED SOLUTION

4.1Client Aware Scheduling

The proposed client-aware credit scheduler schedules the client requests based on the client device features and capabilities. The server is made aware of the client features and capabilities by using the Rich Internet Application (RIA) and the way it is done is explained below.

4.1.1Need for Client Aware Credit Scheduler

The cloud encompasses a variety of client devices. A lot of industries and organizations these days are resorting to the cloud and requesting software, infrastructure, and platforms as services. Mobile clients, PDAs, workstations, laptops, tablets, ASIC devices and smart phones are some of the client devices to name that request services from the cloud. These devices vary in a wide variety of attributes such as screen size, screen resolution, battery usage, location, processing power and so on. Workstations are usually plugged to a power outlet while mobile clients need to be power aware. Laptops and workstations have a higher processing power compared to mobiles clients and smart phones. Certain clients are located closer to the servers than the other clients. Mobile clients and smart phones have built in GPS capabilities. In order to achieve a balance in providing fair service [27] to the mobile clients and plugged in clients, the client-aware credit scheduler is proposed and it has many advantages.

4.1.2 Advantages in terms of Energy Consumption

In order to be beneficial to the clients, certain features and capabilities may be taken into account to schedule the requests from the clients in a cloud. Battery is one of the main concerns for mobile devices [27, 28]. The energy consumption of a mobile device is affected by the complete end-to-end event chain from the client sending request to the server and server responding back. In web applications, the server response times can have a significant impact on the energy consumption of mobile clients [27, 29]. Hence, by reducing the server response time, the power consumed by the mobile client may be reduced.

According to [27], the virtual server response time (Tr) is a function of scheduling time (Schedd) and seek time (Seekd). By reducing either the scheduling time or the seek time, the response time of a virtual server may be reduced. For local virtual server architectures, the WAN is an important entity, the total response time (Tr) depends on the network latency as well.

4.1.3 Advantages in Terms of Reducing Latency

Most of the network latency comes from the distance between the client and the server and also from the processing latency at the routers. The proposed client-aware credit scheduler services the requests from far away clients faster than nearer clients, in order to alleviate the response time. In order for the server to realize these client device capabilities, the authors propose is to use the Rich Internet Applications.

Therefore, if the total response time is reduced, the energy wastage by a mobile client can be minimized and the latency for a distant client can be minimized.

4.2 Client Aware Credit Scheduler

When the clients in a cloud place a service request, the middleware in the cloud assigns the requests to a high, medium or a low priority queue based on the SLA [31]. Each of these queues are sent to the servers separately for getting serviced. In server virtualized environments, a single physical server hosts multiple virtual servers and each of the virtual servers accept the appropriate requests to service.

The client-aware priority scheduler proposed in this research takes into account the client device capabilities inside the high, low, and medium priority queues and schedules the requests inside each queue based on the client capabilities that placed the requests. This ensures that the original priority and SLA of the requests is still maintained, yet scheduling is client-aware inside each of the priority queues.

The three main device capabilities considered in this research include the type of client (mobile or plugged in), the battery remaining, and the distance of the client from the server. There may be other important parameters that can be considered, such as bandwidth, storage capacity, and so on. For example, client bandwidth information may be used to dynamically adjust data delivery and may be used for load balancing. However, in this research, the authors use only device type, battery capacity, and distance metrics for scheduling purposes.

4.2.1 Ordering the Run Queue

The domains ready for scheduling are sent to the VMM. The virtual servers intending to execute are added to a run queue and the domain at the head of the run queue executes before the next domain. In the default credit scheduler, the domains in the run queue are arranged in the First Come First Serve (FCFS) order. In [25, 34], the research done by the authors reveals that an FCFS scheduling is not suitable for latency sensitive applications. The authors in [25, 32, 33] also state that the default credit scheduler does not take the task information in each VM into account. Hence, they implement a mechanism using shared variables in which every virtual server reveals the priority and status of its tasks to the VMM. Based on this information, the priority of a guest domain is taken as the highest priority of tasks in the run queue of the guest domain. The algorithm proposed by the authors in [32] takes into account both the run queue and wait queue of a guest domain to choose the next VM to be scheduled.

In the client-aware credit scheduler, the authors propose some modifications to the algorithm proposed in [15] in order to make the credit scheduler client aware. In the client- aware credit scheduler algorithm, the authors propose that using the same mechanism proposed in [2], the guest domains reveal the device type, battery remaining, and distance of the client attributes of the requests in their run queues to the VMM. Every guest domain knows the client capabilities, whose requests are being serviced using the RIA downloaded on the client. Using this information, the VMM rearranges the domains in its run queue based on the battery life and distance from the client. The following section explains the scheduling of domains by the VMM. Unlike the default credit scheduler, in the client-aware credit scheduler, two run queues are proposed at the VMM. The first run queue holds the requests from mobile clients and the second run queue holds requests from plugged in clients. In the proposed ordering of the domains in the run queues, first a virtual server that services request from a mobile client is pushed to the front end of the first run queue. When two or more

domains are servicing requests from mobile clients, the battery remaining of the clients is checked. The virtual server servicing the request of a mobile client with the lowest battery life is present near the head of the queue for earlier execution. For example, if VM1, VM2, and VM3 are servicing the requests from mobile clients M1, M2, and M3 with battery remaining 65%, 55% and 80% respectively, then VM2 is run first, VM1 is run next, and VM3 is run last. This type of ordering of run queue ensures that clients having low battery are serviced before they can run out of charge. The domains in the run queue that service requests from power plugged clients are ordered in the second run queue based on their distances from the server. A client that is farthest from the server is serviced before the client that is nearest to the server. This type of ordering ensures that a client farthest from the server does not incur high latency in waiting for the request to be serviced. The same domain might be servicing requests from both a mobile client and a plugged in client. In such a case, the domain will be present in both the run queues. The two run queues that hold the mobile and plugged-in client requests are proposed to be assigned to separate cores on the physical server for parallel execution

5. SIMULATIONS AND RESULTS

Java Socket Programming is used to write the network application programs [36]. To run the network applications, client and server codes are written and together they create client and server environment. Java is simple and it is helpful when writing neat and understandable code of network applications. As a profound approach to support the theoretical calculations, simulations have been performed using the Java Socket Programming. Based on the CAS algorithm, a java program has been written which nearly simulates the proposed model. The client program generates 100 packets by creating 100 threads which equivalent to 100 clients sending packets at time TS. The packets contain the battery percentages and the distance as the payload. The server program, which listens for the traffic receives the packets in a first-come first-serve basis and are all put into a buffer space. Then each packet is read from the buffer and pushed into a new buffer where packets are sorted for every clock cycle of 5 milliseconds. To add the processing delay, a random generator function generates random processing delays TP f. The program has to process the packet and return a reply to respective client. In order to show the processing time, the program is made to stay idle for TP amount of time for every packet. It is clearly evident that the queuing delay increases with the increasing battery percentages and compared to the random times of the unsorted version. The energy consumed by the clients while waiting for the requests

6. CONCLUSION AND FUTURE WORK

This research analyzes the performance of the credit scheduler in a cloud. The default credit scheduler follows a fair allocation policy to all the VMs and does not consider the priorities of requests. This mechanism is unsuitable in a cloud where some requests require preferential treatment over the others. The authors proposed two enhancements to the credit scheduler, segregating the requests into two run queues based on the type of request and then sorting the run queues according to the client capabilities. These optimizations decrease the latencies suffered by high priority clients which require a faster service on a cloud environment. The information about client capabilities such as battery level, battery dissipation rate, distance from the server, and many applicable parameters such as battery percentages, distances from the client are delivered to the hypervisor by the RIA technology. The condition with empty credits after sorting has to be addressed and is left for future work; a method to get the solution using credit stealing is possible. The future work includes the deployment of the application on a real hardware which involves the altering the kernel of the Xen hypervisor and developing the workable Credit Aware Scheduler for it.

REFERENCES

[1] J. E. Smith and N. Ravi, "The architecture of virtual machines," Computer, vol. 38, pp.32-38, 2005.
[2] Applying Client-aware Technologies for Desktop Virtualization and Cloud Services. White Paper, Intel, 2011.
[3] Moving to Virtualization: A Guide to What's Possible from VMware., White paper, VMWare, 2010.
[4] Rosenblum, M.; Garfinkel, T., "Virtual machine monitors: current technology and future trends," Computer , vol.38, no.5, pp. 39- 47, May 2005, doi: 10.1109/MC.2005.176
[5] Goldberg, Robert P., "Architectural Principles for Virtual Computer Systems". Harvard University. pp. 22–26. Retrieved 2010-04-12.
[6] Saransh Mittal,. "Differentiated Network QoS in Xen". IIT Bombay, July 2009. Understanding Full Virtualization, Paravirtualization and Hardware Assist., White paper, VMWare, Nov 2007.
[7] Oliver Garraux, Information Technology Fundamentals, How Virtualization Works and its Effects on IT, Oct 2007.
[8] David Chisnall, The Definitive Guide to the Xen Hypervisor, Prentice Hall PTR, Nov 2007.
[9] M. Bourguiba, K. Haddadou, and G. Pujolle, "Evaluating and Enhancing Xen-Based Virtual Routers to Support Real-Time Applications," in

Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE, 2010, pp. 1-5.
[10] David E. Williams, Juan R. García, Virtualization with Xen: Including XenEnterprise, XenServer, and XenExpress, Elsevier Science, 2007.
[11] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, et al., "Xen and the art of virtualization," SIGOPS Oper. Syst. Rev., vol. 37, pp. 164-177, 2003.
[12] K. J. Duda and D. R. Cheriton, "Borrowed-virtual-time (BVT) scheduling: supporting latency-sensitive threads in a general-purpose scheduler," SIGOPS Oper. Syst. Rev., vol. 33, pp. 261-276, 1999.
[13] L. Cherkasova, D. Gupta, and A. Vahdat, "Comparison of the three CPU schedulers in Xen," SIGMETRICS Perform. Eval. Rev., vol. 35, pp. 42-51, 2007.
[14] D. Ongaro, A. L. Cox, and S. Rixner, "Scheduling I/O in virtual machine monitors," presented at the Proceedings of the fourth ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, Seattle, WA, USA, 2008.
[15] Moblin Platform Awareness Service, http://software.intel.com/en-us/articles/moblin-platform-awareness-service/., browsed 04/25/2012 at 16.25PM.
[16] Laszlo: An Open Source Framework for Rich InternetApplications,http://today.java.net/pub/a/today/2005/03/22/laszlo.html, browsed on 06/21/2012 at 09:20AM
[17]Allaire, J.Macromedia Flash MX - A next-generation rich client. Macromedia, 2002
[18]Mobile Cloud Computing Forum, http://www.cloudcomputinglive.com/events/170-mobile-cloud-computing-forum.html, 2010, browsed on 08/21/2012 at 9:50 PM.
[19]Hoang T. Dinh, Chonho Lee, Dusit Niyato, and Ping Wang., "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches". Wireless Communications and Mobile Computing, doi:10.1002/wcm.1203.
[20]A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, "Saving portable computer battery power through remote process execution," SIGMOBILE Mob. Comput. Commun. Rev., vol. 2, pp. 19-26, 1998.
[21] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," presented at the Proceedings of the 2010 USENIX conference on USENIX annual technical conference, Boston, MA, 2010.
[22]Y. Liu, L. Guo, F. Li, and S. Chen, "An empirical evaluation of battery power consumption for streaming data transmission to mobile devices," presented at the Proceedings of the 19th ACM international conference on Multimedia, Scottsdale, Arizona, USA, 2011.
[23]Enabling Emerging Enterprise Usages with Client-Aware Technologies, White Paper, Intel, Feb 2012.
[24] G. Liao, D. Guo, L. Bhuyan, and S. R. King, "Software techniques to improve virtualized I/O performance on multi-core systems," presented at the Proceedings of the 4thACM/IEEE Symposium on Architectures for Networking and Communications Systems,San Jose, California, 2008.
[25]S. Govindan, A. R. Nath, A. Das, B. Urgaonkar, and A. Sivasubramaniam, "Xen and co.: communication-aware CPU scheduling for consolidated xen-based hosting platforms," presented at the Proceedings of the 3rd international conference on Virtual execution environments, San Diego, California, USA, 2007.
[26]Y. Wang, X. Wang, M. Chen, and X. Zhu, "Power-Efficient Response Time Guarantees for Virtualized Enterprise Servers," presented at the Proceedings of the 2008 Real-Time Systems Symposium, 2008.
[27]C. Krintz, Y. Wen, and R. Wolski, "Application-level prediction of battery dissipation," presented at the Proceedings of the 2004 international symposium on Low power electronics and design, Newport Beach, California, USA, 2004.
[29]N. Thiagarajan, G. Aggarwal, A. Nicoara, D. Boneh, and J. P. Singh, "Who killed my battery?: analyzing mobile browser energy consumption," presented at the Proceedings of the 21st international conference on World Wide Web, Lyon, France, 2012.
[30]Robert B. Leighton, Richard Feynman, The Feynman Lectures on Physics. Vol I, 1963 Chapter 13, § 3, pp 13-2,3
[31]M. Dakshayini, H. S. Guruprasad. "An Optimal Model for Priority based ServiceScheduling Policy for Cloud Computing Environment". In International Journal of Computer Applications. '11, 23-29,
[32]S. Xi, J. Wilson, C. Lu, and C. Gill, "RT-Xen: towards real-time hypervisor scheduling in xen," presented at the Proceedings of the ninth ACM international conference on Embedded software, Taipei, Taiwan, 2011.
[33]C. Xu, S. Gamage, P. N. Rao, A. Kangarlou, R. R. Kompella, and D. Xu, "vSlicer: latency-aware virtual machine scheduling via differentiated-frequency CPU slicing," presented at the Proceedings of the 21st international symposium on High-Performance
[34] Parallel and Distributed Computing, Delft, The Netherlands, 2012.
[35]M. Bourguiba, K. Haddadou, and G. Pujolle, "Evaluating and enhancing xen-based virtual routers to support real-time applications," presented at the Proceedings of the 7th IEEE conference on Consumer communications and networking conference, Las Vegas, Nevada, USA, 2010.

[36] Sarah Vallieres, "Understanding EdgeSight Network Data", White Paper, Citrix, 2010.
[37] James F. Kurose, Keith W. Ross, Computer Networking, 5th edition, 2009.