Journal of Theoretical and Applied Information Technology
10th June 2014. Vol. 64 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN EFFECTIVE CROSS RATE PACKET AGGREGATION SCHEME FOR VIRTUALIZED NETWORK CLOUD COMPUTING

A. KAMALESWARI [1] AND P. THANGARAJ [2]

[1]Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, Erode, Tamilnadu, India
[2] Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, Erode, Tamilnadu, India.

ABSTRACT

Virtualization enables the cloud computing. The bottleneck of transferring packets from driver domain to Virtual Machines (VMs) in I/O channel was eliminated by Basic Aggregation and Destination based Aggregation scheme, but it is limited to aggregating packets for a particular destination and missing out more effective packet combinations. This can cause significant loss in capacity as packets on high quality links get demoted to low transmission rates. In this paper, using a new packet aggregation scheme, the network performance is improved. The proposed scheme aggregate the packet based on data rate dynamically on the I/O channel characteristic and it achieves better communication in cloud network, it balance the data rate between wired and wireless. The proposed model named as Cross Data Rate based Aggregation (CDRA) divides packet in MAC queue into groups based on the data rate, packets are to be transmitted at driver domain to VM. The algorithm aggregates packets in the same group and broadcasts the aggregated frame at the data rate of that group. CDRA aggregates packets for all links that have varying data rate and able to transmit packets out of order. The experimental evaluation shows that selectively demoting packets can further improve performance.
Keywords: *Aggregation, Cloud computing, Cross-Data Rate, Driver domain, Virtualization.*

1. INTRODUCTION

Cloud computing has to developed the computation's performance while reducing its cost for users [1]. A common data center for cloud computing consists of tens to hundreds of thousands of servers and comprises hundreds to thousands of hierarchically connected switches. By distribution of computing resources through services such as software as a service (SaaS), users can reduce the cost of hardware and software. In the SaaS model [11], cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. Cloud users do not manage the cloud infrastructure and platform. Where the application runs. This eliminates the need to install and run the application on the cloud user's own computers, which simplifies maintenance and support. Cloud applications are different from other applications in their scalability which can be achieved by cloning tasks onto multiple virtual machines at run-time to meet changing work demand. Virtual Machines are generally employed to offer the services and their migration crosswise the physical host's results in higher resource utilization.

A cloud platform basically consists in multiple data centers with a web portal, linked through a WAN. The data center is a collected of various physical nodes linked over a LAN. Inner of the data center, the infrastructure can be virtualized in which case each physical machine supports multiple isolated virtual machines. These virtual machines share the same hardware and storage.

Virtualization: In computing, refers the act of creating a virtual (rather than actual)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

version of something, including but not limited to a virtual hardware platform, operating system (OS), storage device, or network resources.

Network Virtualization: In computing, network virtualization is the process of combining hardware and software network resources and network functionality into a single, software-based administrative entity, a virtual network.

Aggregation: The act of gathering something together in cloud.

Packet Aggregation: In a packet-based communications network, packet aggregation is the process of joining multiple packets together into a single transmission unit, in order to reduce the overhead associated with each transmission.

Cloud computing [1] one of the key components is resource virtualization, it provides computing and storage services. Even though most of the readers should be familiar with distribution of task to CPU and use the storage efficiently, the partition and distribution of network I/O resources hasn't been expressed as well. This article offer network I/O virtualization technologies, explain the challenges, and presents new trends in obtain scalable data-center networking for cloud computing. When coupled with virtualization, the cloud computing model even enables higher utilization rates while reducing dedicated hardware costs.

Virtualization is an old technology that gained renewed interest recently. It basically offers a partitioning technique to run multiple and isolated virtual machines on a single physical machine. Thus, virtualization optimizes hardware usage through resource reuse and multiplexing which decreases the cost of power, hardware and network bandwidth. Furthermore, through on demand virtual machine creation and migration and dynamic resource allocation it enables flexible, scalable and cost effective virtualized data centers deployment.

A Virtual Machine Monitor (VMM) [2] is a software layer that manages resource sharing among the concurrent virtual machines (VMs) and ensures that diverse and different applications run in isolated environments. The driver domain is a special virtual machine that is in charge of managing the shared access to the devices, especially the network interface card (NIC). The driver domain handles networking by multiplexing outgoing and demultiplexing incoming traffic. This additional layer in the packets path obviously incurs an additional overhead. The I/O mechanism of the VMM consists in copying the packets to the shared memory between the driver domain and the virtual machine.

The physical resources and cache allocation impact on the VM networking performance based on multi-core processors. We proposed packet aggregation for networking performance enhancement. Our work goes deeper in the bottleneck analysis and extends the study to the delay and loss rate.

Initial proposals for aggregation, which we called as Basic Aggregation (BA) [5], aggregate all packets in the MAC queue, irrespective of destination, and broadcast the aggregated frame. However, BA does not consider the presence of multi-rate MAC protocols. In an environment where different destinations experience different link quality, the AP must broadcast the aggregated frame at a low enough data rate to ensure that it is received by all destinations. This can cause a significant loss in capacity as packets on high quality links get demoted to low transmission rates. This impact is not seen in previous studies because the evaluations assume that all links transmit at a fixed rate, the highest rate supported by the radio. In practice, applying aggregation without consideration to the packet data rates can cause excessive packet demotion, often leading to less efficient operation.

The Destination based Aggregation (DA) [5] was proposed where packets that have the same destination are aggregated and unicast to that destination. One can consider two naive mechanisms for aggregation: end-to-end and hop-by hop. In

Journal of Theoretical and Applied Information Technology
10th June 2014. Vol. 64 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the end-to-end scheme, the aggregation is done only at the ingress node for all flows routed for a common destination. A forced delay is added at the ingress to perform packet aggregation. However, its applicability is limited to availability of enough packets for aggregation under the forced delay budget. In the hop-by-hop scheme, packets are aggregated and de-aggregated at every hop by adding a forced delay at every hop. This scheme is oblivious to source or final destination of a given packet and can result in adding cumulative delay to all packets over multiple hops. The hop-by-hop scheme also suffers from extra complexity at each node for executing aggregation/deaggregation on every incoming packet. Furthermore, like BA, DA assumes that packets are transmitted in order, simplifying aggregation, but in general, missing out on more effective packet combinations.

Data Rate based Aggregation (DRA) [5], aggregates packets for all links that have the same data rate and is able to transmit packets out of order. These two properties allows DRA to have a large number of opportunities to aggregate packets, and hence outperform DA, while disabling packet demotion, and hence outperform BA. DRA does not allow packet demotion and therefore cannot take advantage of such aggregation opportunities. However, considering every possible packet demotion leads to exponential complexity.

Our proposed model named as Cross Data Rate based Aggregation (CDRA) divides packets in the MAC queue into groups based on the data rate they are to be transmitted at driver domain to the VM.

2. THE RESEARCH METHOD

Virtualization technology [6] offers effective features like isolate function, manageability and live migration. Unfortunately, network performance like the overhead of network I/O virtualization significantly degrades the performance of network-intensive applications. Two extensive factors of loss in I/O performance result from the extra driver domain to process I/O requests and the extra scheduler inside the virtual machine monitor (VMM) for scheduling domains. In this system packet processing performance is less.

Existing VMMs does not provide sufficient performance isolation to guarantee the effectiveness of resource sharing, specifically when the applications running on multiple virtual machines (VMs) of the same physical machine are competing for computing and communication resources. As a result, both cloud consumers and cloud providers may suffer from unexpected performance degradation in terms of efficiency and effectiveness of application execution or server consolidation.

To overcome these problem [7] focus on performance measurement and analysis of network I/O applications (network-intensive applications) in a virtualized cloud. To improve the benefit and effectiveness of server consolidation and application consolidation, we argue that it is important to conduct in-depth performance measurements for applications running on multiple VMs hosted on a single physical machine. They overcome the bottleneck problem but in this method data aggregation schema are not performed; it becomes performance degradation and packet loss.

In [8] propose two optimization techniques. First optimization is to perform packet ‘aggregation’: Different incoming network packets for the same TCP connection are aggregated into a single huge packet; previous being processed by the TCP stack. The packet aggregation cost is much lower than the gain carry out as a result of the TCP stack having to process fewer packets. TCP header mechanism is still done on a per-packet basis, because it is fundamentals for aggregation, but it can be small part of the per-packet overhead. The most expensive components, in specific the buffer management, are performed once per aggregated packet rather than once per network packet, it should be leading to consider the overall reduction in cost.

The second optimization [8] for reducing the per-packet overhead of receive

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

processing is Acknowledgment Offload. It reduces the number of TCP ACK packets that need to be processed on the transmit path of receive processing, and thus limit the overall per-packet overhead. These optimizations result is important improvements in the performance of TCP receive processing in native Linux. These methods become less performance while using other Operating system like windows platform.

More recently, Destination based Aggregation (DA) [4] was proposed where packets that have the same destination are aggregated and unicast to that destination. Previous works have used DA to improve back haul traffic in networks. As we show, because DA is limited to aggregating packets for a particular destination, it has limited aggregation potential.

Data rate aggregation (DRA) [5] does not allow packet demotion and therefore cannot take advantage of such aggregation opportunities. However, considering every possible packet demotion leads to exponential complexity.

The design of XenSocket [9] replaces the Xen page-flipping mechanism with a static circular memory buffer shared between two domains, where information is written by one domain and read asynchronously by the other domain. It draws on best-practice work in this field and avoids incurring the overhead of multiple hypercalls and memory page table updates by aggregating what were previously multiple operations on multiple network packets into one or more large operations on the shared buffer. XenSocket we have successfully achieved our goal of same-system interdomain transport throughput that approaches that of interprocess communication using UNIX domain sockets, does not support all domain platform.

Aggregation algorithms [10] combine several small packets into one larger packet and forward this latter to an aggregation target. Then use this technique to propose a new I/O model for the communication between the driver domain and the VM. Aggregation is then used to construct trains of packets that we call containers to transfer packets between the netback and the netfront in both directions. By processing a group of packets as one unit, more data can be transferred with fewer requests for memory grants and revokings, less copy and less notification. Packets aggregation is performed based on their MAC destination address. It assumes that packets are transmitted in order, simplifying aggregation, but in general, missing out on more effective packet combinations.

In this paper we focus on performance measurement and analysis of network I/O applications (network-intensive applications) in a virtualized cloud. To improve the benefit and effectiveness of server consolidation and application consolidation, we argue that it is important to conduct in-depth performance measurements for applications running on multiple VMs hosted on a single physical machine. Such measurements can offer deeper understanding of the key factors for effective resource sharing among applications running in virtualized cloud environments.

3. OVERVIEW

A virtual driver is split into the netback and the netfront. All the virtual interfaces are connected to the bridge through the netback. The bridge multiplexes the incoming traffic to the different netbacks and demultiplexes the outgoing traffic to the NIC. Inter-domain communication as well as communication between the hypervisor and the virtual machines is ensured by the I/O channel. A piece of computer software, firmware or hardware that creates and runs virtual machines is called a hypervisor. It is a notification mechanism that is particularly used by the hypervisor to notify driver domain of the arrival of a packet, or by driver domain to notify the virtual machines that a packet was placed in its memory space. In the shared memory the aggregated packet are stored. Shared memory pages are used to

Journal of Theoretical and Applied Information Technology
10th June 2014. Vol. 64 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

really transfer the packet between domains. The process of combining multiple packets together into a single transmission unit, in order to reduce the overhead associated with each transmission is called Packet Aggregation .The related diagram is represented in figure (1):

Fig.1. shows the path of a packet in a virtualized system. Upon the arrival of a packet to the network device (1), the latter notifies the hypervisor by a physical interrupt (2). The hypervisor forwards that notification to the driver domain through the I/O channel (3) as a virtual interrupt. The notified packets are sending to network device to device driver (4). When the driver domain is next scheduled, its device driver handles the packet and transfers it into the Ethernet Bridge (5). This latter bridges it to the corresponding netback according to its data rate (6). The netback driver initializes transfer of packets from the driver to the virtual machines. The netback then notifies the net front of the arrival of the packet through the I/O channel (7) and then aggregates the packet using CDRA techniques (9). It aggregates the packet based on the data rate on each packet. The net

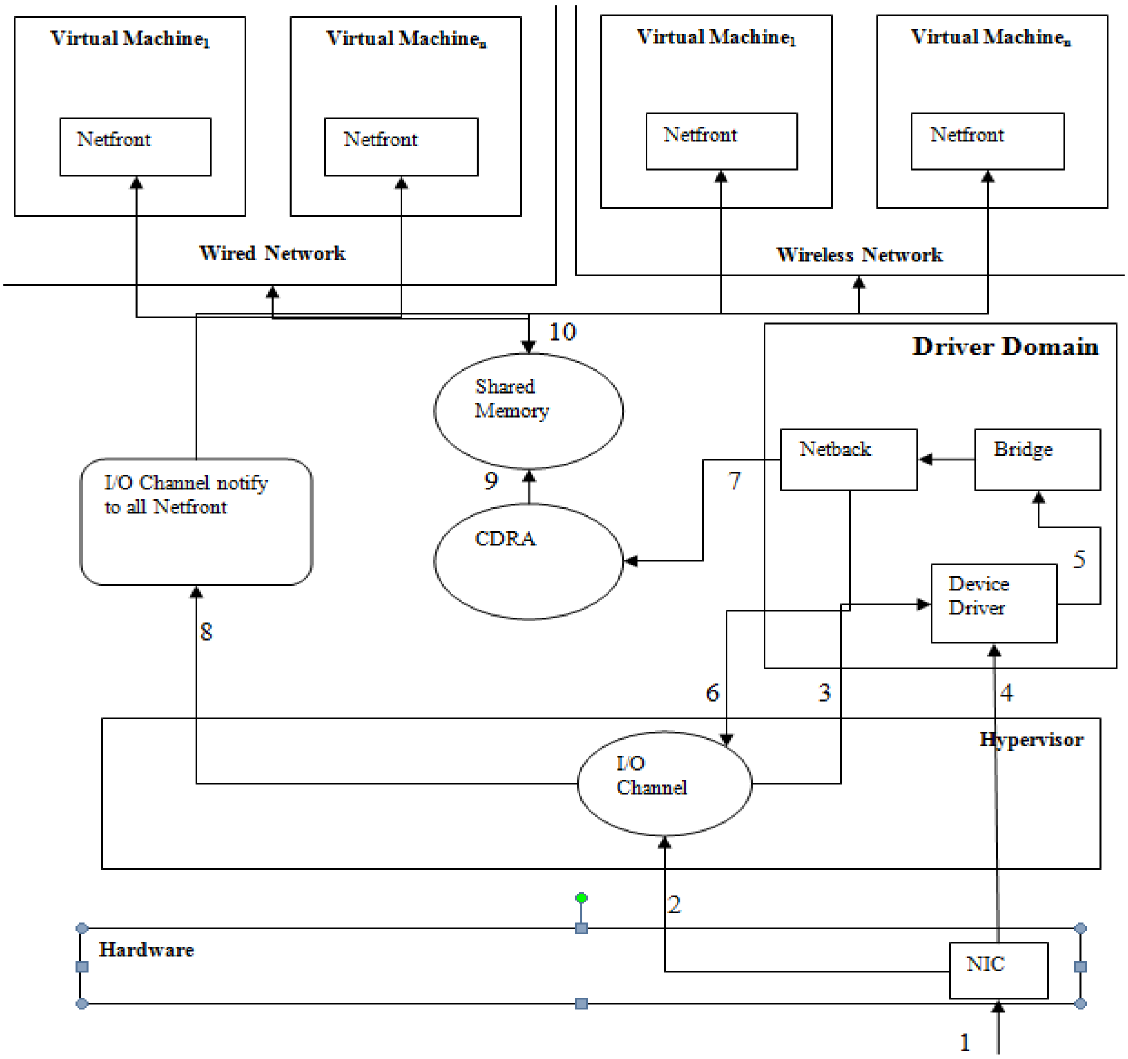

Figure 1: Packet Aggregation In I/O Virtualization

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

front driver receives packets from the driver domain and passes it onto the TCP stack. When scheduled, the destination sees the notification (8) looks for the packet in the common memory page and relays it to the virtual machines to be processed (10).

4. CROSS DATA RATE BASED AGGREGATION

4.1Algorithm

Our proposed scheme used to aggregate the packet based on data rate. Cross data rate means, to transfer the data between wired to wireless network or wireless to wired network the data rate will be varying. Based on the network we can assign the data rate for each packet and then aggregate the packet. The packets are transferred from one wired network to other wireless network; the data rate should be differing. So the proposed system to cross checks the data rate for each packet moving from one network to other network, based on the data rate we aggregate the packet.

Based on these observations we design Cross Data Rate based aggregation (CDRA) to disable packet demotion. CDRA, shown in Algorithm 1, divides packets in the queue into groups. Packet Type is representing range of packet rates. The incoming packet compare with the range of packet type then it is grouped. Each group consists of packets that are to be transmitted at the different data rate. CDRA aggregates packets from the group together and broadcasts the aggregated frame at the data rate for that group; avoiding the loss in performance linked with aggregating across the data rates. Generating each frame requires at most checking each packet size once so if there are n packets in the MAC queue the algorithm is O(n).

Algorithm1: Cross-Data Rate Based Aggregation
Input: MAC Queue
Output: Aggregated Packet

Initialize N - Number of packets in MAC queue, R - Data Rate, PT- Packet Type

1. // Initialize the parameters value
2. // Initialize aggregated packet size
3. aggPktSize = 0;i=1,j=1;
4. // Check all the packets in the queue to aggregated
5. // first 1 packet in MAC Queue
6. while (N != 0)
7. {
8. // data rate of the i^{th} packet
9. $R_i = N_i \rightarrow$ dataRate
10. // aggregate the packet from the range of packet type based on data rate
11. if(R_i € Range(PT_j)
12. {
13. // Take the minimum MTU for MAC queue
14. // MTU_i refers the maximum transmission unit of i^{th} I/O channel
15. MTU_{min} = min (MAC->MTU_i)
16. // aggregate the packet it less than the MTU
17. if ((aggPktSize$_{ij}$+ $N_i \rightarrow$ size) < MTU_{min})
18. // add pkt to aggregated frame for each data type
19. aggPktSize$_{ij}$+ = $N_i \rightarrow$ size
20. }
21. // choose next packet type and aggregate the packet
22. j=j+1
23. }
24. // choose next packet from the queue and aggregate
25. i =i+1
26. N=N-1
27. }
28. }

MAC Queue consists of packets. Each packet has its own data rate and type. Each packet from MAC Queue is taken and grouped accordingly with respective to the range of packet type based on data rate. Before group the packet, the aggregated packet size must be checked that is less than the maximum transmission unit or not. If the condition is satisfied packets are aggregated and choose the next packet from the queue. Otherwise choose the next packet type then process the packet and aggregate.

5. MATERIALS AND METHODS

In our experimental cloud setup simulation is performed by using cloudsim. Using cloudsim we create the 4 host machines and 8 virtual machines with

Journal of Theoretical and Applied Information Technology
10th June 2014. Vol. 64 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

different bandwidth. Each virtual machine runs the more number of tasks. We have developed the proposed model of Cross Data Rate based Aggregation (CDRA) divides packets in the MAC queue into groups based on the data rate they are to be transmitted at driver domain to the VM. Below we present the results and discussion for the proposed system.

6. RESULTS AND DISCUSSION

In our experimental cloud setup simulation is performed by using cloudsim. Using cloudsim we create the 4 host machines and 8 virtual machines with different bandwidth. Each virtual machine runs the more number of tasks. Measure the performance of MAC and CDA in terms of packet delay, loss rate and output rate are showed in the following graph.

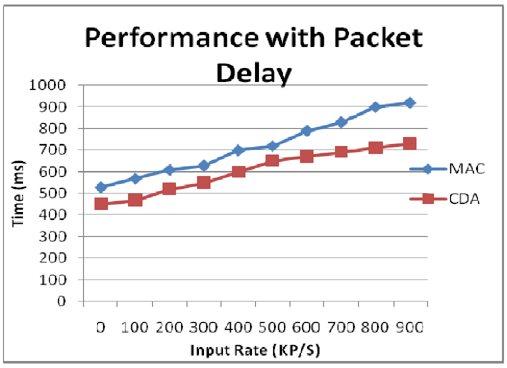

Figure 2. Performance With Packet Delay.

The packet delay behaves differently from MAC and CAD. First notice the obvious increase of the packets average delay due to the additional waiting time until the container timeout expires or the maximum size is reached. Based on the cross data rate the time to wait aggregate packets is represented in Y-axis in both MAC and CDA. Fig 2 shows that the input rate are high the packets delay rate also high in MAC system when compare to CDA system.

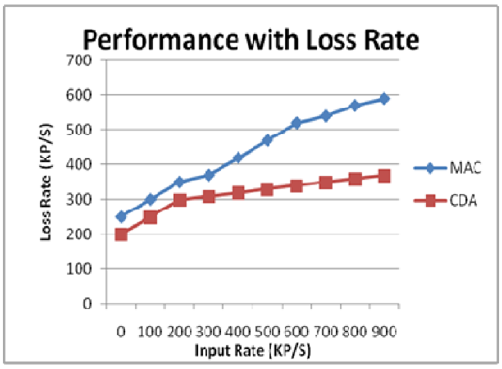

Figure 3. Performance With Loss Rate.

Fig 3. Based on the cross data rate the loss rate to aggregate packets is represented in Y-axis in both MAC and CDA. Input rate values are shown form 0-900 with 100 intervals, corresponding loss rate measured from each intervals in the x-axis. It shows that the input rate are high the loss rate also high in MAC system when compare to CDA system.

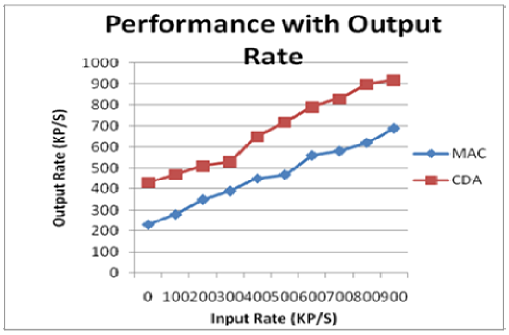

Figure 4. Performance With Output Rate.

Fig 4. Shows the cross data rate the output rate to aggregate packets is represented in Y-axis in both MAC and CDA. Input rate values are shown form 0-900 with 100 intervals, corresponding output rate measured from each intervals in the x-axis. It shows that the input rate are high the output rate also high in MAC system when compare to CDA system.

7. CONCLUSION

Virtualization is becoming a key technology to enable deploying efficient and cost-effective cloud computing

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

platforms. However, current network I/O virtualization models still suffer from performance and scalability limitations. Memory transactions are limited by the long memory latency which represents a hardware bottleneck of the system. To overcome this limitation, we proposed a packet aggregation mechanism that allows transferring containers of packets at once. In this paper we show that aggregation mechanisms proposed by previous works either lead to performance degradation in I/O, or have very limited performance gains. We formulate the optimal aggregation problem for an I/O virtualization. We propose heuristics technique to solve this problem. The packet are aggregated based on the data rate. Our proposed system is named as Cross-Data Rate Aggregation. Cross data rate means, to transfer the data between wired to wireless network or wireless to wired network the data rate will be varying. Based on the network we can assign the data rate for each packet and then aggregate the packet. The packets are transferred from one wired network to other wireless network; the data rate should be differing. So the proposed system to cross checks the data rate for each packet moving from one network to other network, based on the data rate we aggregate the packet. The heuristic, CDA, shows significant improvement in performance, compared to state of the art aggregation protocols, by aggregating packets for each data rate separately. Experimental evaluation shows that the proposed packet aggregation mechanism significantly improves the networking performance.

REFERENCES

[1] A. Menon, and W. Zwaenepoel, ”Optimizing TCP Receive Performance”, Proc. USENIX Annual Technical Conference (USENIX’ 08), 2008

[2] Adnan Majeed and Nael B. Abu-Ghazaleh,” Packet Aggregation in Multi-rate Wireless LANs”, Proc. IEEE (SECON), June 2012.

[3] Ahmed Amamou, Manel Bourguiba, Kamel Haddadou and Guy Pujolle, “A Dynamic Bandwidth Allocator for Virtual Machines in a Cloud Environment”, Proc. IEEE (CCNC), Jan 2012.

[4] Guangdeng Liao, Danhua Guo, Laxmi Bhuyan, Steve R King, “Software Techniques to Improve Virtualized I/O Performance on Multi-Core Systems”, Proc. ACM/IEEE (ANCS ’ 08), 2008

[5] Manel Bourguiba, Kamel Haddadou, Ines El Korbi, Guy Pujolle, "Improving Network I/O Virtualization for Cloud Computing," IEEE Transactions on Parallel and Distributed Systems, 25 Feb. 2013.

[6] S. Ganguly, V. Navda, K. Kim, A. Kashyap, D. Niculescu, R. Izmailov, S. Hong, and S. Das, “Performance optimizations for deploying voip services in mesh networks,” Selected Areas in Communications, IEEE Journal on, vol. 24, no. 11, pp. 2147 – 2158, 2006.

[7] W. Wang, S. C. Liew, and V. Li, “Solutions to performance problems in voip over a 802.11 wireless lan,” Vehicular Technology, IEEE Transactions on, vol. 54, pp. 366–384

[8] Wikipedia, “cloud computing”, http://en. Wikipedia.org/wiki/cloud computing, May 2008.

[9] Yan Luo, "Network I/O Virtualization for Cloud Computing," IT Professional, vol. 12, no. 5, pp. 36-41, Sept.-Oct. 2010.

[10] Yiduo Mei, Ling Liu, Xing Pu, Sankaran Sivathanu, Xiaoshe Dong, "Performance Analysis of Network I/O Workloads in Virtualized Data Centers," IEEE Transactions on Services Computing, vol. 6, no. 1, pp. 48-63, First Quarter 2013

[11] Zhang, S. McIntosh, P. Rohatgi, and J. L. Griffin, ”XenSocket: A High-Throughput Interdomain Transport for Virtual Machines”, Proc. ACM/IFIP/USENIX Middleware Conference (Middelaware’ 07), 2007.