Volume 1. Issue 1

Transactions on Networks and Communications

Society for Sciences and Education - United Kingdom

Table of Contents

The Communication Performance of Link-Sharing Method of Buffer in NoC Router

Smartphone User's Traffic Characteristics and Modelling

Virtualization in Networks: A Survey

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC Transactions on Networks and Communications

Vol. 1 Issue 1

The Communication Performance of Link-Sharing Method of Buffer in NoC Router

The relation between the communication performance and the number of banks

Naohisa Fukase [1], Yasuyuki Miura[1], Shigeyoshi Watanabe[1], M.M. Hafizur Rahman[2]
[1]Graduate School of Technology, Shonan Institute of Technology,
1-1-25, Tsujido Nishikaigan, Fujisawa, Kanagawa, Japan;
[2]Dept. of Computer Science, KICT, International Islamic University, Malaysia;

ABSTRACT

We have proposed a memory sharing method of the wormhole routed network-on-chip architecture. In our method, a memory is shared between multiple physical links by using the multi-port memory. In this paper, we present the pipeline processing method, and evaluate the communication performance in the various situations. The pipeline of the proposed method has two courses of the route 1 and 2. The number of pipeline stages of route2 is 2 stages larger than the traditional router in order to use a shared memory. But delay is concealed if the capacity of a private buffer is enough. It is shown that the required number of memory banks required in multiport memory for 2-dimensional torus and 2-dimensional mesh networks is 8. Our proposed method yields high performance for both torus and mesh networks. Even this high performance is retained when the buffer size and the packet length are same.

Keywords: Router, Interconnection Network, Network-on-Chip (NoC), Multi-Port Memory.

1. INTRODUCTION

Network-on-Chip (NoC) connects hundreds of Intellectual Properties (IPs)/cores, including, programmable processors, co-processors, accelerators, application-specific IPs, peripherals, memories, reconfigurable logic, and even analog blocks. In spite of the many advantages of NoC, area overhead and power consumption still remain the drawback. Therefore, it is necessary to design a high performance router using as minimum hardware resources as possible to minimize the layout area and power consumption.

A single memory is shared by multiple virtual channels for efficient utilization of router buffer is proposed and implemented [1-3]. However, this sharing is taken place in a few virtual channels. For sharing the buffer in more channels or links, we have proposed a buffer sharing method of multiple

N. Fukase, Y Miura, S. Watanabe and M.M. Hafizur Rahman; *The Communication Performance of Link-Sharing Method of Buffer in NoC Router-The relation between the communication performance and the number of banks* in Transactions on Networks and Communications Volume 1, No 1 (2013), PP 01-13
URL: http://scholarpublishing.org/index.php/TNC/article/view/20/TNC-13-1005
Publication Date: December 19th 2013

physical links. Using the proposed method more channels can be shared and the router can utilize buffers more efficiently.

The method of sharing a ring buffer and sharing a multiple buffer are presented in [4] and [5], respectively. However, due to use of large crossbar switch, it is difficult to share large ring buffer. Since wormhole routing is not used in [5], the communication latency becomes prohibitively large because the number of pipeline stages is increased.

In our previous research [6-8], we have proposed the method of sharing a buffer by multiple physical links for effective use of a router buffer. We found that the conventional implementation of sharing technique increase the hardware cost for a large number of physical links. To overcome this problem, we introduce hardware cost reduction method which uses a Multi-bank Multi-port memory [9-10].

In our previous research, we have evaluated the performance of torus network only. It was shown that our proposed method using multi-bank memory has almost same performance with the method using conventional multiport memory when the number of banks is sufficient enough. On the other hand, it is necessary to evaluate the performance when the number of banks is not enough.

The remainder of the paper is organized as follows. In Section II, we briefly describe the conventional method. The proposed method and its hardware cost are discussed in Section III, IV and V, respectively. The communication performance of the proposed method is discussed in Section VI. Finally, in Section VII, we conclude this paper.

2. CONVENTIONAL METHOD

In NoC, a PE consists of one or more processor cores and a router circuit. In router circuit, a crossbar switch is used to connect input links to output in which the communication takes place. A physical link usually has multiple virtual channels [11], and a buffer is integrated to each channel of the input side of the crossbar switch to smooth the flow of packets in communication. Unconstrained use of hardware is strictly prohibited for cost-effective design. Wormhole routing [12] is used for cost-effective design if PE, because it can be implemented by using comparatively a little buffer.

The simple structure of wormhole routers uses a buffer of same capacity installed in each channel [12]. However, the buffer allocated to the channel is not utilized effectively because some channels and buffers remain idle or unutilized. To overcome this problem, sharing a memory by flits between multiple virtual channels of a physical link were proposed and implemented conventionally in [1-3].

By this conventional method, the memory block of a shared memory is assigned dynamically and used when the capacity of the buffer of a channel becomes insufficient. In this method, a connection between acquired memory blocks is expressed by recording the arrangement of memory to "VC Block Info" of the assigning channel. In this paper, such method is called “Channel Sharing” and the proposed method mentioned in the subsequent section is called “Link Sharing”

3. PROPOSED METHOD

3.1 Outline

Till now the sharing of buffer over a physical link is not used because of increased hardware cost. The link sharing method needs to use a multi-port memory as a shared memory because it responds to the concurrent access from multiple physical links. However, the hardware cost becomes enormous if normal multiport memory is used. It is because the required hardware cost is the square of the number of ports. The structure of the proposed method and the multi-bank multiport memory used in our method are depicted in Figure1 and Figure 2, respectively. As portrayed in Figure 1, each channel has a ‘Private Buffer’ and a shared memory is laid out between input ports in the router. In the proposed method, the ‘Multi-bank Multi-port Memory’ is applied as the shared memory to reduce the hardware cost. As illustrated in Figure 2, the Multi-bank Multi-port Memory has some memory banks which have a few ports, and banks are put between two crossbar switches. In this multi-port memory, it is not necessary to add multiple ports to each memory cell. So it can suppress the increase of hardware cost. However, the Multi-bank Multi-port memory cannot access to addresses in the same bank at the same time.

To solve this problem, we have proposed the ‘By-Block sharing method’. Here, a shared memory is divided into some block and is allocated by every block. By associating each block and bank, the link which accesses to each bank is limited to one. Moreover since the management target becomes a block of the memory, this method can reduce hardware cost. In this paper, the method of controlling a memory by every flit is called ‘By-Flit control’. And the method of controlling by every block is called ‘By-Block control’.

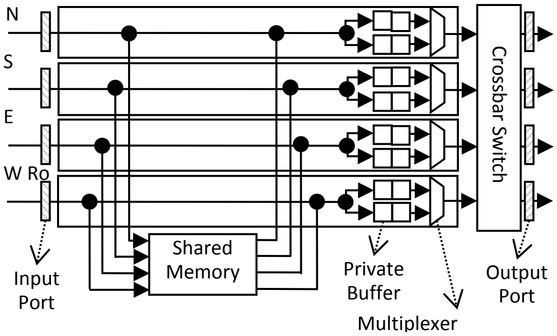


Figure 1: Router Structure of Proposed

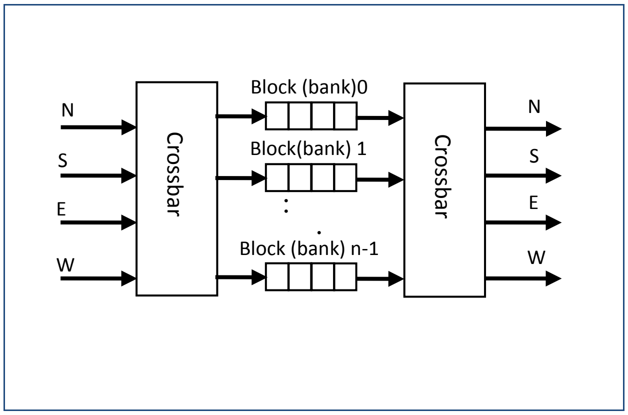


Figure 2 : Multi-bank Multiport Memory

The link sharing method may not allocate a memory to a virtual channel or a physical link due to full of the memory. Thereby, a deadlock [13] may occur. To solve the deadlock problem, a buffer called the 'Private Buffer' of minimum capacity for the communication is laid out to each channel in this proposed method. Even if a shared memory is not allocated, each channel can communicate and can avoid a deadlock.

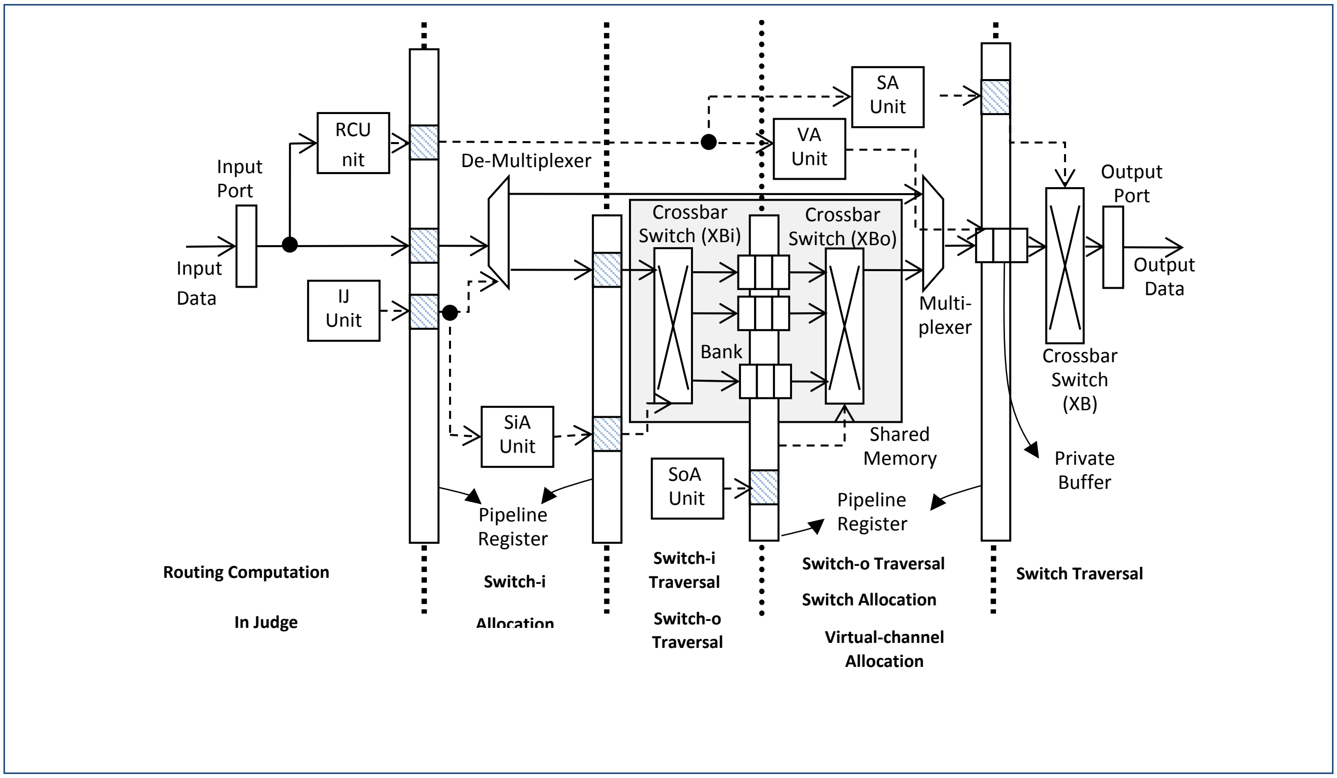


Figure 3: The Block Diagram of Proposed Method

3.2 Hardware Structure

A block diagram including the pipeline structure of the proposed method is portrayed in Figure 3. As depicted in Figure 3, the proposed method has 5 pipeline stages. Each stage is the area surrounded by the dashed line. Each stage is divided by the pipeline register (shown by rectangle in the figure) and buffers such as shared memory and private buffer.

4. PIPELINE STRUCTURE

4.1 Structure of Traditional Router Pipeline

The pipeline structure of the traditional router is shown in Figure 4[14]. As shown in Figure 4, A traditional pipeline performs the following four processes in three steps.

1) Routing Computation (RC) : An output link is determined from the information on header.
2) Virtual Channel Allocation (VA) : The virtual channel to output is assigned.
3) Switch Allocation(SA) : The arbitration and setup of a crossbar switch are performed.
4) Switch Traversal (ST) : Flit passes a crossbar switch.

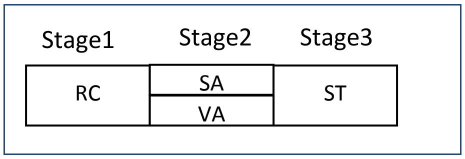


Figure 4 : Pipeline Structure of the Traditional Router

4.2 Pipeline Structure of Proposed Method

In Our method, injected flits pass along the following two routes. When not crowded, it passes along the route 1. When crowded, it passes along the route 2.

route1 : input port⇒private buffer ⇒output port

route2 : input port⇒shared memory⇒private buffer⇒output port

The route 1 is a course immediately sent to Private Buffer after arriving at an input port. The pipeline of the route 1 is shown in Figure 5(a). As shown in Figure 5(a), the route 1 operates on the same three-step pipeline as traditional router. However, In-Judge (IJ) process is executed in stage 1 as shown in Figure 5(a). In IJ process, “Whether the shared memory is used or not”, and “A now block is allocated or not” are determined. Since the output link of a packet is decided regardless of whether a shared memory is used, RC and IJ processes can be processed in parallel.

The route 2 is a course that the packet is sent into the shared memory and then goes to the private buffer after that. The route 2 needs the stage for a setup and traversal of the switch in the input and output port of the Multi-bank Multiport Memory. The pipeline of the route 2 is shown in Figure 5(b). The route 2 needs the following stages:

1. IJ(In-Judge) : “Whether the shared memory is used”, and “A new block is allocated” are determined in IJ process.
2. SiA(Switch-i Allocation) : Set the input crossbar switch of the multi-bank multiport memory.
3. SiT(Switch-i Traversal) : When it succeeds in a SiA step, a packet passes the input crossbar switch of multi-bank multiport memory and it stores to the shared memory.

4. SoA(Switch-o Allocation) : A setup of the output crossbar switch of the multi-bank multiport memory and the block release process are performed simultaneously.
5. SoT(Switch-o Traversal)

When it succeeds in a SoA step, a packet passes the output crossbar switch of multi-bank multiport memory and it stores to the private buffer.

As shown in Figure 5(b), the number of stages of route 2 is 2-stages larger than route 1. But the delay by the pipeline of the route 2 is concealed by following reasons.

- When a network is not crowded and the private buffer is not full, it becomes same stages as the conventional method since it is processed according to a three-stage pipeline (route 1).
- As a network is crowded, the private buffer becomes full and the shared memory will be used. The number of flits in the private buffer increases by blocking the packet by the crossbar switch. If the private buffer is designed to permit one blocking (If the number of flits of the private buffer is two or larger), the pipeline using the shared memory will smoothly flow.

The example of the pipeline of proposed method is shown in Figure 5(c). In Figure 5(c), the 3rd flit goes to the route 2 since top flit was blocked.

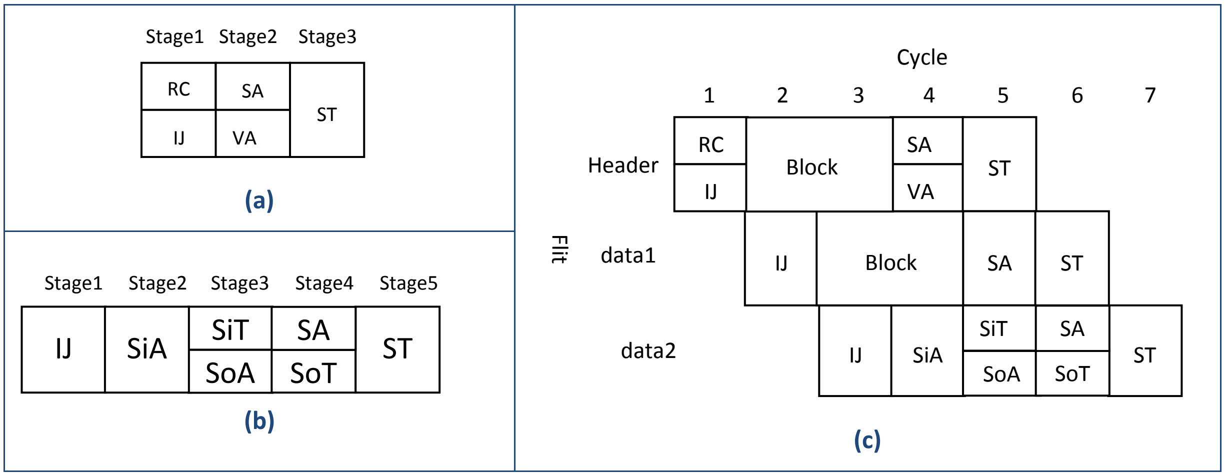


Figure 5 : (a) Route 1 Pipeline, (b) Route 2 Pipeline, (c) The Example of the Proposed Pipeline

5. HARDWARE COST

In this section, the hardware cost to implement the proposed method is estimated. In the conventional method, most of the hardware cost is 'buffer' of the physical link except crossbar switch and control circuit. 'Memory element for control information' is needed for both the traditional method and the proposed method. Memory element includes the buffers for control the shared memory [6-8]. Additional hardware costs for the proposed method are 'logic circuit for block control' and 'surrounded circuits of multiport memory'.

The hardware cost of a physical link can be roughly estimated by estimating the above mentioned elements. In this evaluation, *B*, *C*, *L*, *F*, and *W* are defined as follows:

B: Total number of memory blocks in all links

C: Total number of channels in all links

L: Number of links

F: Number of flits in a block

W: The number of bits per a flit

In this condition, the number of channels per link is *C⁄L*, and the number of memory block per link in channel sharing (conventional) method is *B⁄L*. Also, in the "By-Flit Sharing", *F* is set as one. The number of transistors for implementation is counted to evaluate the hardware cost. The cost of memory element is assumed as 6, *n*-input NAND (NOR) gate is 2*n*, inverter is 2, the cross point of crossbar switch is assumed to use a tri-state inverter so the number of transistors is assumed as 6.

The implementation cost in terms of the number of transistors of conventional and proposed method is tabulated in Table 1. In the evaluation the total amount of buffer is kept same (*B*×*F*=64) and the number of blocks (*B*) are varying. For both the conventional method and by flit and link sharing as shown in Table 1, the value of *F* is equal to 1.

It is shown in Table 1 that the hardware cost of the proposed method decreases with the decrease of the number of blocks (the value of *B* become smaller). The hardware cost can be drastically reduced compared with by-flit implementation (*F=1*). Although the additional logic circuit for block control is needed, the hardware cost reduction effect of the memory element for control information and surrounded circuits of multiport memory exceeds the proposed method. When the router circuit is implemented on the condition of $B \leqslant C$, the cost of the proposed method becomes double to that of conventional method. As mentioned above, the hardware cost of proposed method can be reduced by "By-Block Sharing".Further hardware cost reduction is possible because arbitration and switches for the bank memory can be reduced. It is to be noted that crossbar switch is used for the shared memory of the proposed method.

Table 1. Implementation Cost of Proposed Method (Transistors)

W	Topology	*L*	*C*	*B*	*F*	Conventional Method	By Flit and Link Sharing	**Proposed Method**	
								Total	**Improvement rate**
64	Ring	2	4	16	4	30732	154318	**58146**	**1.89203**
				8	8			**41710**	**1.35722**
				4	16			**33494**	**1.08987**
	2D torus	4	8	32	2	29832	281678	**156034**	**5.23042**
				16	4			**90798**	**3.04364**
				8	8			**58322**	**1.95501**
128	Ring	2	4	16	4	55308	277198	**107298**	**1.94001**
				8	8			**78574**	**1.42066**
				4	16			**64214**	**1.16103**
	2D torus	4	8	32	2	54408	502862	**278914**	**5.12634**
				16	4			**164526**	**3.02393**
				8	8			**107474**	**1.97533**

6. PERFORMANCE EVALUATION - RELATION BETWEEN THE NUMBER OF BLOCKS AND COMMUNICATION PERFORMANCE

The communication performance is evaluated by software simulation. Every PE generates packet with a specified probability in every clock cycle and transmits the packet to randomly selected PE. These processes were carried out for 200000 cycles, and average transfer time and average throughput are recorded. On the same network parameters and every probability of occurrence, simulations are carried out for 10 times, and the average of transfer time and throughput are plotted in a graph. In this experiment, the average transfer time and throughput are calculated and plotted as throughput in the horizontal axis and average transfer time in the vertical axis.

We use a dimension-order routing for packets routing to route packets. We have considered 2D-mesh and 2D-torus network of size 16 (4×4) and 64 (8×8) for performance evaluation. Two virtual channels per physical link are simulated. The message length is considered as 16, 32, and 64 flits; and the buffer length of each router is 32 and 64 bits.

We evaluate the influence of the number of blocks. If the number of blocks is small value, the hardware cost of the proposed method will become small. But, communication performance may fall because the utilization efficiency of a memory falls. Figure 6-12 portrayed the results of simulations of a torus and mesh network. The upper graphs of those figures are results of torus, and the lower are mesh.

In our evaluation, we compare the following cases;

- no-sharing：It does not share.
- by-flit-link：It is one type of link sharing method. It does not use by-block memory sharing.
- B2, B4, B8：It is a link sharing method called by-block memory sharing. In those methods, the number of blocks are 2 (B2), 4 (B4), and 8 (B8).

As shown in those figures, the following points are clarified.

1. The progress ratio of the mesh is smaller than that of torus. In a mesh network with more PEs, the ratio between edge and corner PE and total number of PE is low. Since the buffer of the unused link in corner and edge PE can be used in a mesh, the performance of mesh with a few of PE improves substantially by the proposed method. Moreover, unlike torus, since the mesh network requires one virtual channel to prevent deadlock, two channels can be freely used.

2. It is shown that the performance is significantly improved when the total amount of buffers and the packet length are similar. When packet length is very large or when the packet length is extremely small, the performance improvement is not impressive.
3. The difference in performance is trivial for B8 and by-flit-link method. On the other hand, the performance of B2 and B4 are lower than by-flit-link in many cases. As stated above, eight is enough as the number of blocks in 2D mesh and torus. Henceforth, eight is used as a basic status of the number of blocks.

4. When the total amount of buffers is larger than the packet length, a clear difference is shown between B4 and B8. Compared with packet size, the block size is too large in the

case of B4 with those situations. So the memory in a block remains. Therefore, it is thought to be desirable that the block size is smaller than packet size.

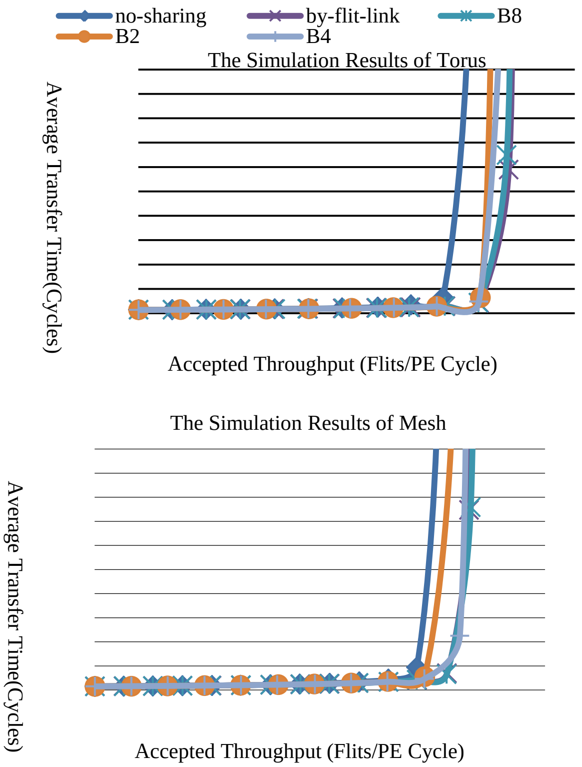


Figure 6: The communication performance of a torus and mesh: 16 PE, 32 Buffer, and 16 Flits/Packet

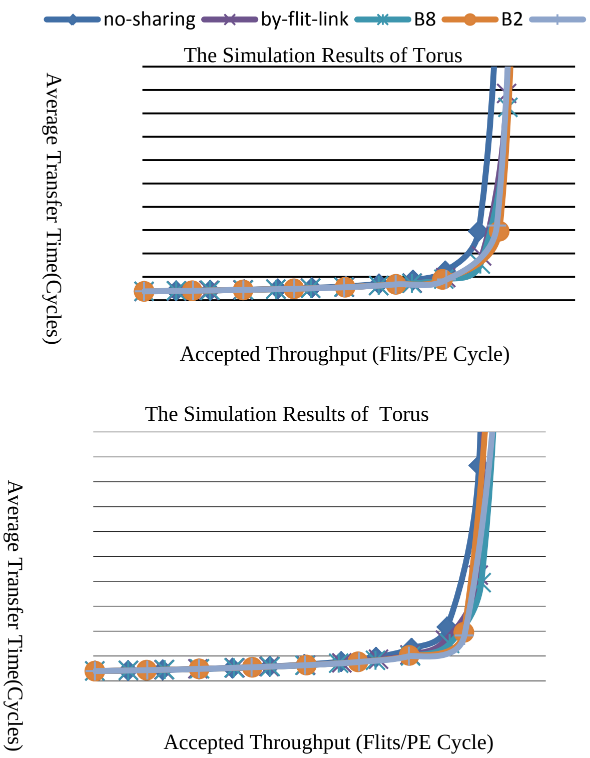


Figure 7: The communication performance of a torus and mesh: 16 PE, 32 Buffer, and 64 Flits/Packet

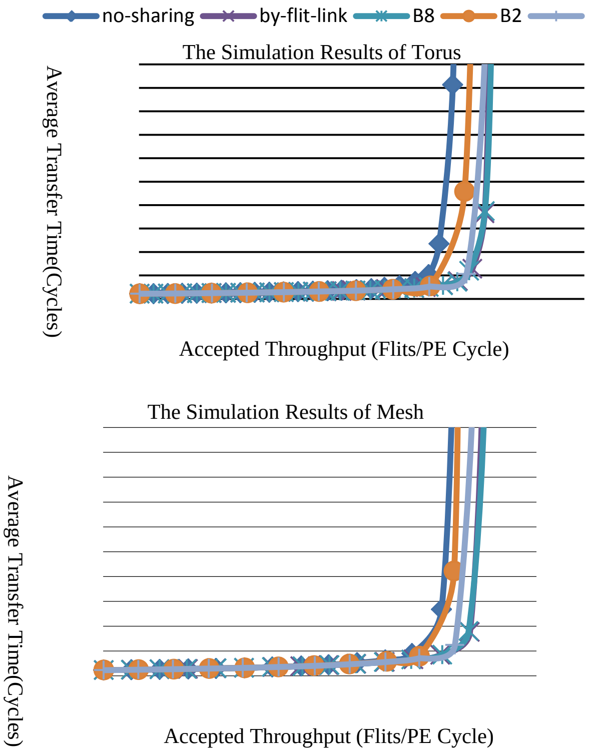


Figure 8 : The communication performance of a torus and mesh: 16 PE, 64 Buffer, and 32 Flits/Packet

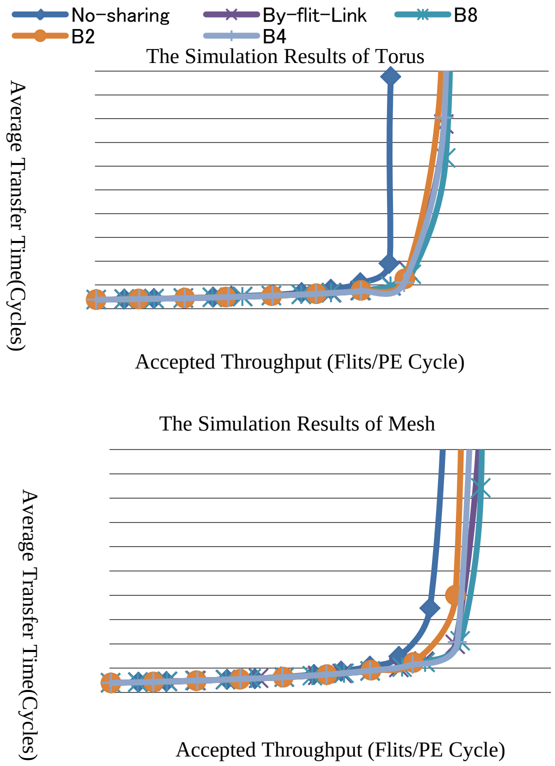


Figure 9: The communication performance of a torus and mesh: 16 PE, 64 Buffer, and 64 Flits/Packet

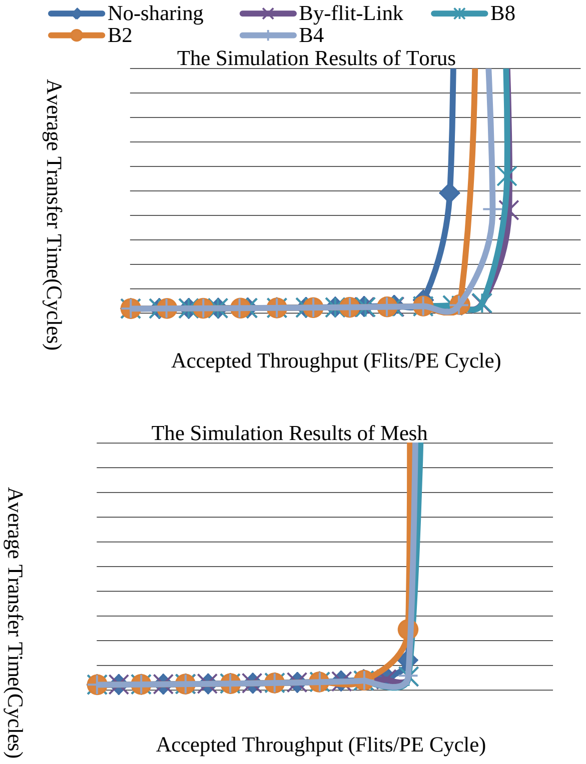


Figure 10: The communication performance of a torus and mesh: 64 PE, 32 Buffer, and 16 Flits/Packet

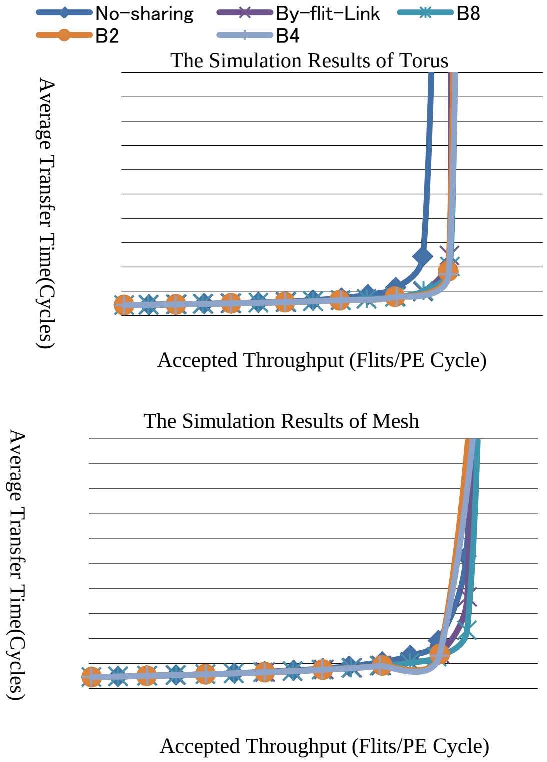


Figure 11: The communication performance of a torus and mesh: 64 PE, 32 Buffer, and 64 Flits/Packet

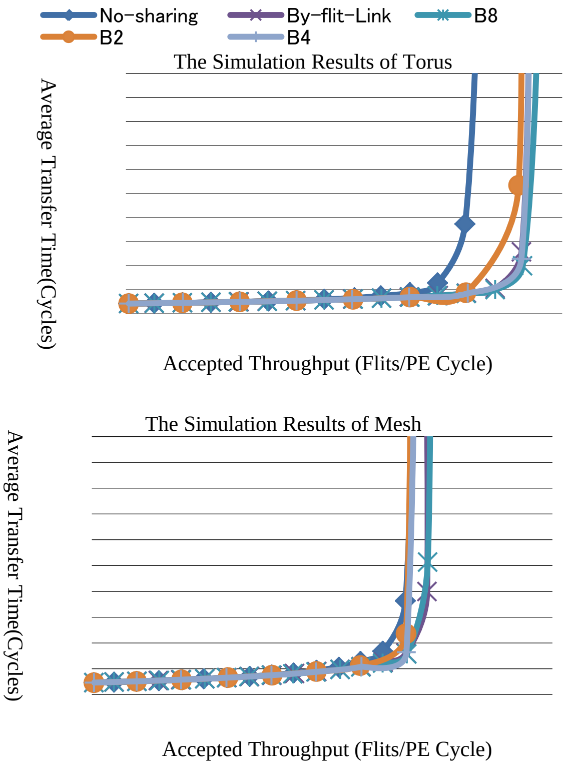


Figure 12: The communication performance of a torus and mesh: 64 PE, 64 Buffer, and 64 Flits/Packet

7. CONCLUSION

In this paper, we presented the sharing method of multiple physical links in a NoC router. And we presented the pipeline processing, method and evaluated in detail the communication performance in the different situations. The number of pipeline stages of proposed method is 2 stages larger than the traditional router in order to use a shared memory. However, we also showed that delay was concealed if the capacity of a private buffer is enough. We found that eight is sufficient enough number of banks in the multiport memory for 2D mesh and 2D-torus. We have evaluated the performance considering both 8 banks and less than 8 banks. We found that both the mesh and torus network yield the higher performance by the proposed method. Issues for future work and further exploration includes the evaluation of performance of the high dimensional network such as 3-D torus or mesh networks.

REFERENCES

[1]. Kumary, P.Kunduz, A.P.Singhx, L.-S.Pehy, N.K.Jhay, A 4.6its/s 3.6GHz single-cycle NoC router with a novel switch allocator in 65nm CMOS, 25th International Conference on Computer Design(ICCD 2007), pp.63-70, 2007.

[2]. Gregory L. Frazier, Yuval Tamir, The design and implementation of a multiqueue buffer for VLSI communication switches, Proceedings of the International Conference on Computer Design Cambridge, Massachusetts, pp.466-471, 1989.

[3]. Yuval Tamir， Gregory L. Frazier， Dynamically-Allocated Multi-Queue Buffers for VLSI Communication Switch IEEE Trans. Computers， Vol.41， No.6， pp.725-737， 1992.

[4]. Ahmadinia and A. Shahrabi, A Highly Adaptive and Efficient Router Architecture for Network-on- Chip, The Computer Journal, Vol.54 Issue 8, pp.1295-1307, 2011.

[5]. R.S. Ramanujam, V. Soteriou, B. Lin and L.S. Peh, Extending the Effective Throughput of NoCs with Distributed Shared-Buffer Routers, IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, vol.30, No.4, pp.548-561, 2011.

[6]. Naohisa Fukase，Yasuyuki Miura，Shigeyoshi Watanabe，Link-Sharing Method of Buffer in Direct-Connection Network，The 2011 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp.208-213, 2011.

[7]. Naohisa Fukase, Yasuyuki Miura, Shigeyosi Watanabe, The Hardware Cost Reduction Method of Control Circuit for Link-Sharing Method of Buffer in NoC Router, 2013 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing, March 2013.

[8]. Naohisa Fukase, Yasuyuki Miura, Shigeyosi Watanabe, The Proposal of Link-Sharing Method of Buffer in NoC Router : Implementation and Communication Performance, Jounal of Basic and Applied Physics, (In printing).

[9]. Michael Golden et al., “A 500MHz write-bypassed, 88-entry, 90bit register file,” Proc. of Symposium on VLSI Technology, Session C11-1, 1999.

[10]. H.J Mattausch, K.Kishi and T.Gyohten, “Area-efficient multi-port SRAMs for on-chip data-storage withhigh random-access bandwidth and large storage capacity,” IEICE Trans. Electron., Vol.E84-C, No.3, p410, 2001.

[11]. W.J.Dally, Virtual-Channel Flow Control, IEEE Trans on Parallel and Destributed Systems, Vol. 3, No. 2, 1992.

[12]. M. Ni and P. K. McKinley， A Survey of Wormhole Routing Techniques in Direct Networks， Proc of the IEEE， Vol. 81， No. 2， pp. 62-76， 1993.

[13]. E. Fleury and P.Fraigniaud, A General Theory for Deadlock Avoidance in Wormhole-Routing Networks, IEEE Trans. Parallel and Distributed Systems, Vol. 9, No. 7, pp. 626-638, 1998.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC Transactions on Networks and Communications

Vol. 1 Isuue 1

Smartphone User's Traffic Characteristics and Modelling

Sonja Filiposka[1,2], Igor Mishkovski[1]

[1]*Faculty of Computer Science and Engineering,*
Ss. Cyril and Methodius University, Skopje, R. Macedonia;

[2] *Departament de Ciències Matemàtiques i Informàtica,*
University of Balearic Islands, Palma de Mallorca, Spain;

ABSTRACT

The proliferation of smartphones and the demand for all-day connectivity has brought exponential growth of global mobile data traffic. To survive the explosive progression and best serve their customers, mobile network operators need to have a better understanding of the nature of traffic carried by cellular networks. Understanding the characteristics of this traffic is important for network design, traffic modelling, resource planning, and network control. In this work we investigate the basic characteristics of smartphone traffic, identifying and understanding the impact of context (location, time, physical interface) on smartphone usage for calls, messages and data traffic. In order to identify and characterize patterns in the user traffic generated by smartphone devices in the mobile networks, we employ naturalistic logging methodology based on non-obtrusive background data collection while aiming for a highly diverse study participant's backdrop. Our statistical results present a comprehensive analysis on user habits while using their smartphones on a daily or weekly basis. By taking advantage of the gathered user logs and the statistical analysis of the traffic characteristics, we attempt to design a mobile traffic generator that will create synthetic voice, message and data traffic according to the observed real life traffic characteristics. The generated mobile traffic scenarios can be used not only for modelling the mobile operators' network (such as 3G and 4G), but also WiFi, mobile ad hoc and sensor networks.

Keywords: Smartphone traffic patterns, mobile user behaviour, analysis and modelling, traffic generator.

1. INTRODUCTION

The mobile evolution, driven by video, cloud-based services, and the Internet, changes how people behave and how they leverage mobility to communicate and to improve their daily lives, using existing and new services. Today users demand ubiquitous connectivity anywhere and

Sonja Filiposka and Igor Mishkovski: *Smartphone User's Traffic Characteristics and Modelling.* Transactions on Networks and Communications Volume 1, No 1 (2013), PP 14-40
URL: http://scholarpublishing.org/index.php/TNC/article/view/22/TNC-13-1015
Publication Date: December 19th 2013

anytime. The driving forces of these trends also include new affordable smartphones, and the vast number of new connected devices on the market. The total number of mobile subscriptions globally will reach around 9 billion in 2017 [1]. With an increased number of subscriptions, smart devices and all-day connectivity, the global mobile data traffic is expected to grow 15 times by the end of 2017. Approximately 35-40% of all mobile phones sold in the first quartile of 2012 were smartphones, compared to around 30% for the full year 2011 [2]. Only around 10-15% of the worldwide base of subscriptions uses smartphones, which means that there is considerable room for further uptake [3].

Because of rapidly growing subscriber populations, advances in mobile communication technology, increasingly capable smartphones, and the expanding range of mobile applications; mobile networks have experienced a significant increase in data traffic [4]. Mobile data subscriptions grow strongly, and drive the growth in data traffic along with a continuous increase in the average data volumes per subscription. Mobile voice traffic also continues to grow at a smaller steady rate. To cope with this explosive growth and best serve their customers, operators need to have a better understanding of the nature of traffic carried by cellular networks. Understanding the characteristics of this traffic is important for network design, traffic modelling, resource planning, and network control.

However, little is known about the characteristics of the traffic generated from smartphones. Some recent studies ([5]-[9]) have tried to analyse the characteristics of smartphone data traffic to shed light on the topic. The time between the user interactions was observed in [5] and it was shown that if the users were inactive for a longer period, than it is less probable that they would start a new interaction. The research was based on two data sets of 33 Android and 22 Windows users across different demographics. The results show that users interact with their smartphones from 10 to 200 times a day. The duration of one session can vary highly, but most of the sessions were short. The conclusion that was drawn from the data traffic was that quantity of the data traffic generated from smartphones nowadays is similar to the one generated by the computers few years ago. In [6] the authors have analysed the application usage on a national level using data from a network operator. The traffic was collected in the period of one week and distinct marketplace apps were identified using HTTP signatures. The results obtained from this research can be used to optimize the content delivery in LTE and WiFi networks, i.e. which content can be stored on servers close to the clients. Another conclusion was that the daily cyclic models of different types of applications were quite different. For instance, news applications were used more in the morning hours, while some other applications were used throughout all the day. Surprisingly, most of the traffic that the users generated in this research came from the application for personalized radio (3 TB in one week or 50% from the total traffic). In addition, there were big discrepancies between the applications concerning the consumed traffic. The amount of traffic and the access time have been shown to increase linearly with the number of users. However, because of the high variation in this correlation, it is difficult to model the traffic amount and the access time based on the number of users. The authors made additional analysis on which types of application were more used during the day or night and if the applications' use depends on the smartphone model the users possessed.

In [7] the authors present the characteristics of the mobile http-based traffic, by using stamps on packet level from a large mobile network. The main results from this work show the comparison from the dataset obtained from the wireless set of data in one mobile network and wired set obtained from one local network. 15% of the applications have generated more traffic in the mobile network, whereas 70% in the wired network. The conclusion was that the traffic in the mobile network was a lot more bursty and in smaller packets. Similar data traffic analysis was done in [8]. The authors show that the daily amount of data traffic depends on the smartphone platform and whether or not the users use WiFi. Their focus of investigation is how the data traffic impacts the battery life and the proper use of the available bandwidth (because of the additional headers from the layers below). This research also showed the users habits concerning the use of mobile and WiFi networks. Most of the traffic that the 39 users made during 5 weeks was on the WiFi network (~527 MB), compared to ~90 MB on the mobile network.

The authors in [9] have investigated how SMS messages and voice traffic are connected to the social networks' contacts. Their results show that participants in the study have dialled 57% of their contacts, while they messaged only 19%. However, the number of SMSs was increasing, whereas the number of calls was reducing during the time. The SMSs that were sent to the social network contacts were shorter, while the call duration was longer, comparing to other contacts. A different kind analysis, which tries to capture the characteristics of the users' movement by using the localization data from their smartphones was done in [10]. For instance, the results show that the users in Los Angeles traverse more than the people in San Francisco or Manhattan. This research also focused on determining the possibility for traffic jams as well as the sizes of carbon footprints of urban areas.

Yet, many unanswered questions in this area remain. Thus, one of the goals in this paper is to investigate the basic characteristics of smartphone traffic, identifying and understanding the impact of context (location, time, physical interface) on smartphone usage for calls, messages and data traffic. Though call data records (CDRs) from cellular operators are a valuable source of data for mobility studies that could benefit society at large, obtaining such data is extremely difficult due to operator regulations and the need for individual privacy preservation. On the other hand, user collected data from cellular telephone networks can help study mobile traffic patterns easily and inexpensively, as well as on an as needed frequency basis. Smartphone logging can provide tremendous access to communications data from real environments. Using this approach, special care must be taken in order to preserve naturalistic user behaviours so that the logs are true reflections of the actual everyday activity [11]. Thus, in order to identify and characterize patterns in the user traffic generated by smartphone devices in 3G mobile networks, we utilized logs of user data packets captured at the device via a specially designed nonintrusive application. Volunteer measurements were obtained by deploying a background based measurement tool. Throughout our work, we have taken measures to preserve individual privacy.

In the case of 3G terminals all of the traffic is packet-based traffic. Studying this traffic allows identification of traffic characteristics and patterns described as series of events. An event in this context is the sending or reception of a packet. If it is possible to identify patterns in the traffic and find correlations between events, researchers could exploit knowledge of these patterns and correlations on both the network and user sides. As commented earlier, an improved understanding of user-smartphone traffic patterns would yield insight into a variety of important societal and networking issues [12]. For example, evaluating the effect of traffic patterns on the network capacity depends on knowing how typical mobile users communicate during their daily lives.

Although results show that there is immense diversity among users and their traffic patterns, one common factor is the vast amount of data that is being generated by smartphone devices. Therefore, network operators have to properly dimension their networks by estimating the required capacity of the nodes and links such that they are able to carry the actual amount of traffic that the link experiences, while optimizing expenses. This optimizing of expenses implies that mechanisms to properly dimension packet switched networks are needed. In order to propose a suitable mechanism, it is desirable to evaluate the response of the network to a given set of circumstances, and this is frequently made through simulations.

Networking research has long relied on simulation as the primary vehicle for demonstrating the effectiveness of proposed algorithms and mechanisms. A simulation model is a representation of the key elements of the network. Typically one constructs either a network testbed and conducts experiments with actual network hardware and software, or one simulates network hardware and software in software and conducts experiments via simulation of the network. In either case, experimentation proceeds by simulating the use of the real or simulated network by a population of users with applications that generate representative traffic.

Source traffic generators are used to inject synthetic traffic into the network according to a model that describes the behaviour of the corresponding applications or users. The need for a realistic traffic model is not exclusive to cellular network simulations only. The growing set of diverse applications developed for MANETs and other similar mobile wireless networks, demand for far more complex traffic patterns than the simple traffic pattern of the uniform random generators [13]. Hence, the simple traffic models widely used in previous simulation studies have become inadequate in reflecting the relative performance of these networks when deployed in real life scenarios. A traffic model is a stochastic process that represents the actual traffic measured in a network in a simulation model. Traffic models are used to predict the behaviour of actual traffic streams, so ideally they should preserve all the statistical properties of the original traffic. There are some desirable properties for a traffic model, such as: it should be defined by a small number of parameters, its first and second order statistics should match those of the actual measured traffic, and if the traffic were fed through the model the results should accurately predict those of the real traffic stream fed into an actual network.

There have been many research efforts to find a traffic model that fits the properties and particularities of packet based networks. Some of the proposed models are presented in [14] and briefly described here. In on-off models, a traffic source alternates between two states, „on‟ and „off‟. During an on period, traffic is generated at a constant rate, and during an off period there is no traffic. The lengths of on and off periods are independent. It has been shown that as the number of aggregated on-off sources increases, the resulting process approaches the server occupancy of an M/G/∞ queue [15]. The Poisson-Pareto Burst Process (PPBP) uses a model with bursts arriving according to a Poisson process, and whose durations are Pareto distributed. PPBP can be considered as the limiting process for a large number of independent on-off sources aggregated together [16].

In this work, using logging methodologies as proposed in [11], we are trying to capture the users' mobile traffic characteristics. We are concerned with three different types of events: voice, messages and data. Our statistical results for these types of events present a comprehensive analysis on user habits when using their smartphones on a daily or weekly basis. By taking advantage of the gathered user logs and the statistical analysis of the traffic characteristics, we attempt to design a mobile traffic generator that will create voice, message and data synthetic traffic according to the observed real life traffic characteristics. Since the task at hand is hardly straightforward, we discuss possible approaches and their benefits and drawbacks. The first approach is based on the overall statistical traffic characteristics that we translate into probability distributions for different types of events. Using these distributions, which are the basis for random sampling for our generator, we try to represent the originally observed user behaviour. The second approach, proposed in this work, is to imitate the behaviour of a chosen user. Finally, we expect that using these results we will be able to build a first order approximation that should provide more realistic mobile traffic.

Furthermore, we elaborate the additional possibilities and mappings in order to obtain traffic scenarios even closer to the observed traffic patterns, such as adding social dimension by measuring the influence mapping between the social and the underlying communication network, determining the correlation between the user location and time of day in order to map the user behaviour in different environments. As already discussed, the resulting mobile traffic scenarios generated in this way will be of great significance not only for modelling the mobile operators' network (such as 3G and 4G), but also WiFi, mobile ad hoc and sensor networks.

Please note that in the rest of the paper, unless otherwise specifically noted, when using the term mobile network and mobile traffic we restrict to the network and traffic of cellular telephone network operators.

The rest of this paper is organized as follows. In Section 2 we present and analyse the gathered smartphone user traffic data from our logging application. The presented results are focused on three different traffic types: voice calls, short messages and Internet data traffic. We analyse the users' behaviour in different time contexts (during their daily and weekly activities) for each of the concerned traffic types. In addition, in the last subsection we include the results that rank the applications used by the users, by their popularity and by the Internet data traffic that they have consumed. In Section 3 we present a design for a mobile traffic generator based on two different approaches, statistical and averaged, which can be used for generating traffic scenarios with the three types of events. We also discuss some additional possibilities and mappings that could make the traffic generation model even more realistic and accurate. Section 4 concludes this work.

2. SMARTPHONE USERS' TRAFFIC CHARACTERISTICS

Logging methodologies have addressed many of the concerns about the observer effects on natural user behaviour by employing technology to do the observations. These methodologies provide access to data that can be collected without the presence of observer or a requirement for users to provide self-reports. Data can be pulled from study participants' daily activities on familiar interfaces within normal contexts. Thus, data collected from loggers are typically considered more objective, accurate, and realistic. Smartphone logging of communications data is a recent trend. The logged data gathered under these constraints can be used for advanced research in order to establish empirical models, development of theories and for testing different hypothesis. Although logging has been applied effectively to a number of research aims, no specific intentions where applied to preserve naturalistic behaviours. For instance, many of the previous logging studies mentioned above have reminded participants they are being measured by requiring them to report data, have introduced novel interfaces, or collected data considered private.

In order to provide a more systematic approach while preserving natural user behaviour, as part of this research we designed a background smartphone Android application, which logs different types of events (voice calls, SMS messages and Internet data). Following [11] we have implemented a naturalistic methodology for logging events, taking into account: different variables, users' privacy, participant selection, and the length of the research, level of intrusiveness, the user interface, user tasks and technology.

Thus, first we have selected the participants and we have explained which variables will be used from their smartphones. All of the chosen variables do not threaten the users' privacy. The only suspicious data, i.e. telephone numbers, were replaced by randomly chosen unique identifiers during the logging process. The users did not have any specially appointed tasks, except to install the logging application and to accept the terms of use and export the data at the end of the observation period. During the process of accepting the terms of use, all users where briefed in detail on the types of data that are logged and on the ways employed to ensure their privacy. Upon installation, the application works as a background process and does not disturb the users at any moment. The application logged voice calls, SMS messages and Internet traffic from 18 users of Android phones in the time period of over 3 months. Special attention was given for the chosen users not to be significantly correlated by choosing users with different age, city of origin, profession, smartphone generation (2.3.3 Gingerbread to 4.1 Jelly Bean) and brand (i.e. HTC, Samsung, Sony and others) and mobile operators and packages (prepaid and post-paid). This research was with medium length, thus we claim that the obtained results are realistic. Before deployment of the application, extensive accuracy testing were done in order to ensure the correctness of the gathered data.

From the obtained logs we have divided the traffic events into three separate groups: voice traffic, message traffic and data traffic. The logging variables for the three event types are as follows: 1) calls – type, caller ID, timestamp, call duration; 2) messages – type, caller ID, timestamp; 3) data – interface, application, connection ID, connection start, connection duration, number of received and sent packets in KB. In order to expand our data set, upon installation of the application, the logger crawls all past call and message logs and records the history data. Also, it is important to emphasise that records for VoIP applications that are fully integrated into the OS native call interfaces (e.g. Viber) are considered as voice calls and the corresponding data traffic is not being logged.

In the rest of this Section we summarize the characteristics of the voice, message and data traffic on a fine and coarse time scale. The analysis and results from this Section will be used as a basis for the mobile traffic modelling, presented in Section 3.

2.1 Voice traffic characteristics

Table 1 summarises the main voice call characteristics and the obtained diversity between different call types. We differentiate between 5 call types: incoming, incoming rejected (i.e. incoming calls with duration 0 s), incoming missed, outgoing rejected and missed (i.e. outgoing calls with duration 0 s), and outgoing calls.

Table 1 Voice traffic characteristics

Property	Incoming	Incoming rejected	Missed	Outgoing rejected and missed	Outgoing calls
Average number of calls	4.4	0.3	1.8	2.7	5.1
Average call duration	108.1 s	-	-	-	109.3 s
Standard deviation of the call duration	176.6	-	-	-	188.3
Maximal deviation of the call duration	2821.9	-	-	-	3643.6
Average call duration on daily basis	477.4 s	-	-	-	561.6 s

On a daily basis, the users on average had more outgoing calls (5.1 calls) than incoming calls (4.4 calls). In addition, the average of outgoing rejected and missed calls (2.7 calls) is bigger than the number of incoming rejected and missed (2.1 calls). The average call duration on a daily basis for the outgoing calls is 9.4 minutes, whereas for the incoming calls is around 7.9 minutes. However, the standard deviation and the maximal deviation of the call duration, for both the incoming and the outgoing, is quite high, which shows the diversity of the call data amplified by the heterogeneity of the study participants. The diversity of the results is expected, as it is noted in several previous attempts (see the previous section for more details). The grand scale of the user diversity is becoming a major difficulty that needs to be crossed when designing a synthetic traffic generator.

Comparing our results with the results published by the ITU [4] one can conclude that in US the number of outgoing calls is slightly bigger (8 vs. 5 in our analysis), whereas comparing them with the results from the national agency for electronic communications [17] the observed users have spoken longer than average national citizen (9.4 vs. 5.1 minutes), with a little bit longer average duration of outgoing calls (1.82 vs. 1.75 minutes).

By making an analysis of the number of calls and the call duration in the different hours of the day as well as different days in the week we attempt to capture the basic properties of the users' temporal habits concerning the voice traffic. In Figure 1, we present the average number of calls per hour.

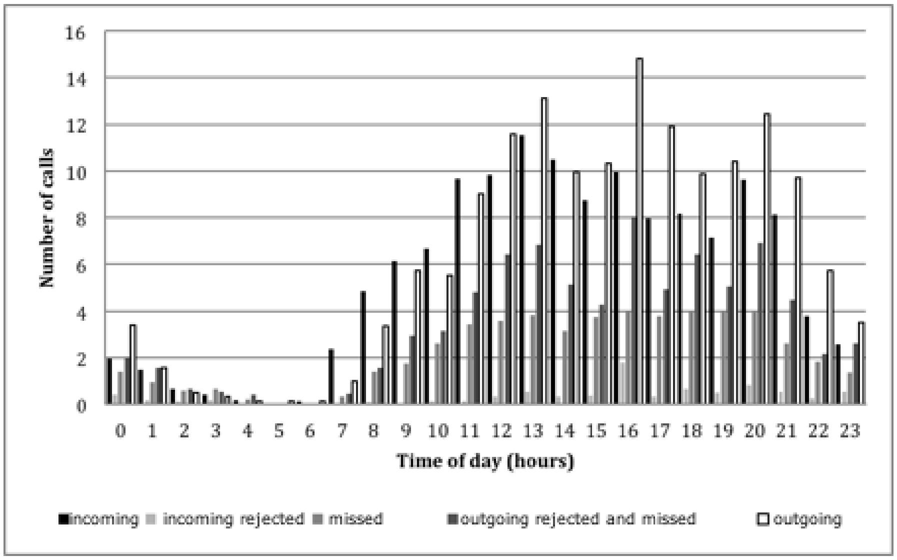


Figure 6: Number of calls distributed in different hours of the day

The average number of all types of calls is heightened in the interval between 7 AM till 1 AM, especially in the period between 12 PM and 8 PM. The maximum is reached during the typical last working hour (i.e. 4 PM). The users' activity then decreases till 6 PM, and gradually increases between 6 PM and 8 PM, which is possibly connected to the users' evening arrangements or preparation for the working activities for the next day. The incoming calls have maximum at 1 PM.

The average call duration in different time of the day is presented in Figure 2. The duration of the user calls is longer after work (from 4 PM till 8 PM) compared to the working hours from 7 AM till 4 PM. The results show that users tend to make more, but shorter calls while working. As expected, the call duration is smallest during early hours after 1 AM till 7 AM. The outgoing calls reach their maximum at 4 PM, while the incoming at 8 PM. The users received more calls than they dialled to.

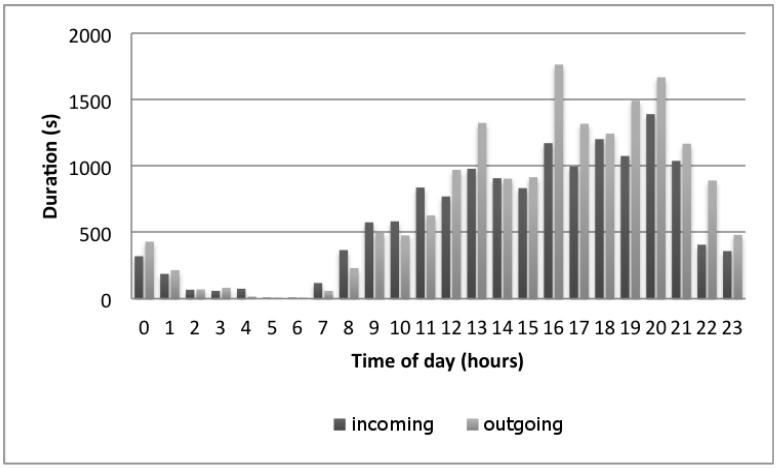


Figure 7: Average call duration in different hours of the day

For an analysis on a more coarse time scale, the average number of calls for every day of the week is illustrated in in Figure 3. It is noticeable that the users' call activity is biggest on Tuesdays, whereas the users mostly tend to reject the incoming calls on Fridays. The voice calls get more sporadic during the weekend, while the peak for missed calls is reached after the weekend.

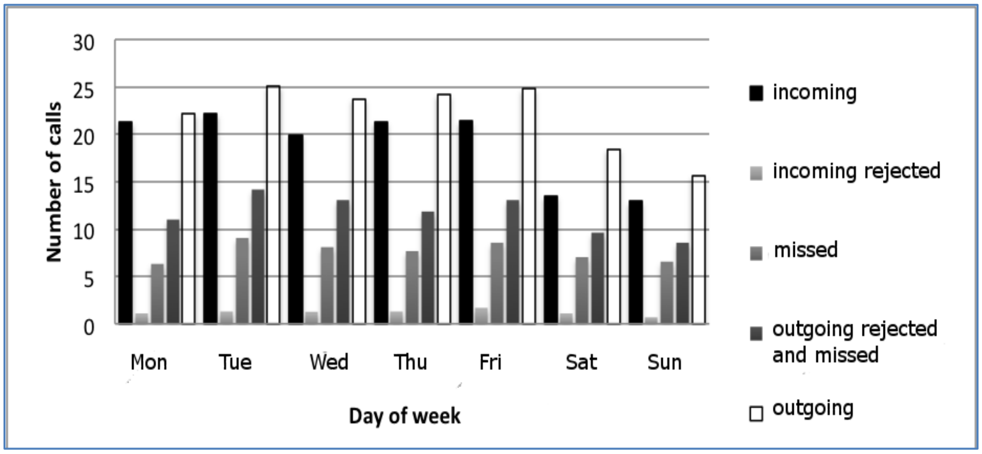
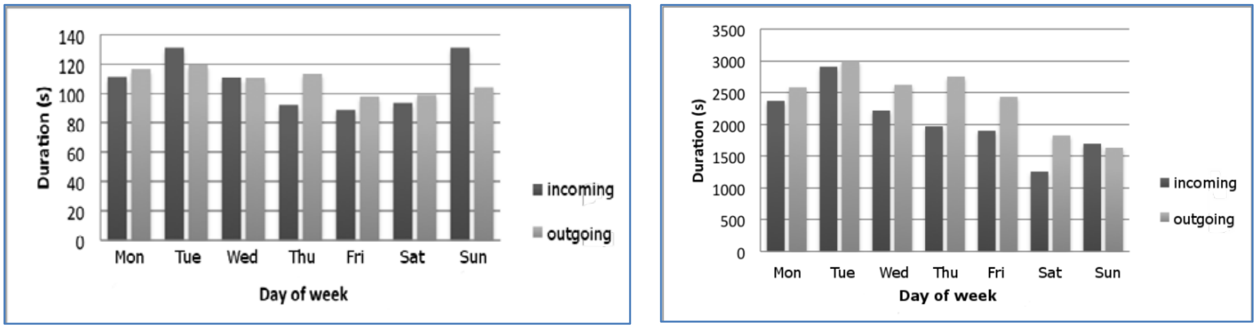


Figure 8: Number of calls in different days of the week

Regarding the call duration on different days of the week, the situation is a little bit different compared to the number of calls. Users tend to have long incoming and outgoing calls on Tuesdays. Please note that this behaviour has been reported in some of the related research studies mentioned previously. However, the duration of the incoming calls is also long on Sundays, while the number of outgoing calls is smallest during the weekends (see Figure 4(a)).

The longer, but fewer calls on Sundays can be due to the fact that the users tend to have long conversations only with their close relatives and friends on Sundays and while the short business related ones with their co-workers are not present. The duration of the incoming and the outgoing calls is shortest on Fridays.

The overall duration of all calls in different days of the week is given in Figure 4(b). By deeper inspection and comparative analysis, one can draw a number of conclusions about the user behaviour: users talk more in general during the working days. As the incoming calls duration becomes shorter towards the end of the working part of the week, the outgoing calls lessen as the week progresses but with lesser intensity. The top talk day is Tuesday for both incoming and outgoing calls.



(a) (b)

Figure 9; (a) Average call duration in different days of the week, (b) Overall call duration in different days of the week

The obtained results also show the diversity in the users' behaviour regarding the number of calls that they have made. For instance, the range of the number of calls per user varies between 8% and 63% of the total number of calls made by all the users.

Another type of analysis conducted in this research is dealing with the social aspect of user interaction. For instance, from the voice call data we have obtained it is evident that most of the users talk to a repetitive small group of contacts, while a small part of the users talk to a larger set of contacts. All users tend to make most of the calls to their top contacts when considering the number of calls and duration of calls also. However, these results are preliminaries and can be further used for mapping and research of the mutual influence between the social and underlying communication network.

2.2 Message traffic characteristics

In the following subsection, we present our analysis on the users' habits concerning sending and/or receiving SMS on different time scales.

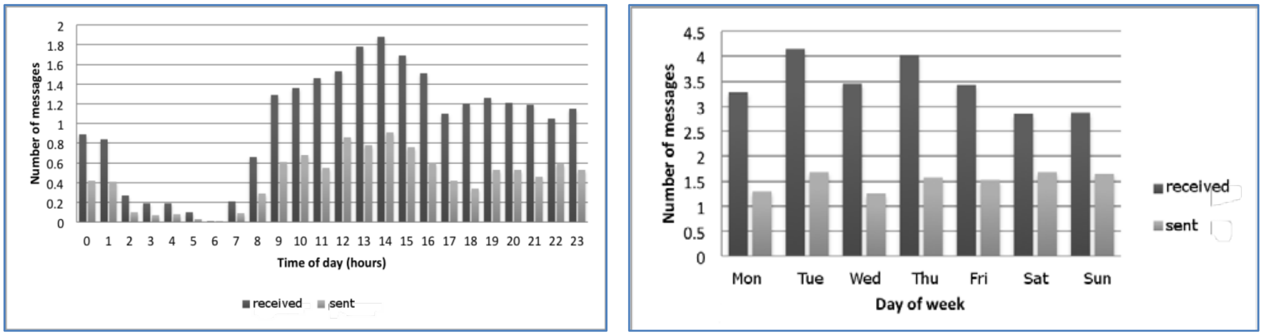
On a monthly basis, users have received 24 messages on average, while they have sent around 10.5 messages. Compared to daily basis, users receive on average around 0.8 and send around 0.35 messages (see Table 2). The obtained results exhibit similar ratio between sent and received SMS on daily basis if compared to the results from ITU (1 versus 1.15) 17]. However, when comparing the results with the national agency for electronic communications the users have sent fewer messages than an average national user (10.5 vs. 18 SMS) [17]. In unison with the comparison made for the voice calls, we can draw a conclusion that smartphone users tend to talk more compared to writing SMS, which seems to be more accepted for users with traditional older mobile phones. This could be due to the inclination towards more cheap means of communication (via SMS) for users that cannot afford smartphones.

Table 2 Message traffic characteristics

Property	Received SMS	Sent SMS
Average number of daily SMS messages	0.8	0.35
Average number of monthly SMS messages	24	10.5

In Figure 5(a) the average number of SMS per user during one day is presented. The users have sent and received most of their SMS messages between 12 PM and 3 PM, with a maximum peak at 2 PM. Another type of analysis is the distribution of average number of SMS for different days of the week. Similarly as in the case for the call duration, the users have received most of their SMS on Tuesdays and during the working days, whereas the users have sent most of their SMS during the weekend, see Figure 5(b) Thus, Tuesday seems to be the overall most "communicative" day of the week, while the weekend shows heightened activity connected with leisure and family.

The diversity in the users' behaviour regarding the maximum number of SMS messages that each study participant have sent and/or received is also prominent. Similarly to the maximum number of calls, the range of the number of SMS per user varies between 8% and 79% of the total number of messages.



(a) (b)

Figure 10: (a) Average SMS in different hours of the day, (b) Average number of SMS in different days of the week

2.3 Data traffic characteristics

The aim of this subsection is to present some of the analysis on the characteristics of the users' data traffic. The Internet data traffic is captured separately on the mobile network and WiFi network interfaces. Table 3 summarizes the main statistical characteristics of the observed traffic.

Table 3 Data traffic characteristics

Property	**Mobile network**	**WiFi network**
Average number of daily connections	6.9	5.8
Average overall Internet connection duration (per day)	3878 s	4029 s
Average amount of received traffic (per day)	8748.2 KB	8437.9 KB
Average amount of sent traffic (per day)	1135.2 KB	1046.9 KB
Average number of received packets (per day)	9588	8329
Average number of sent packets (per day)	8525	6675
Average connection duration	564 s	695 s
Standard deviation of connection duration	1878 s	1694 s
Maximal deviation of connection duration	57957 s	32338 s
Average amount of received traffic (per connection)	1271.2 KB	1455.2 KB
Standard deviation of received traffic	4898.4 KB	7702.7 KB
Maximal deviation of received traffic	118941.8 KB	195683.8 KB
Average amount of sent traffic per connection	165 KB	180.5 KB
Standard deviation of sent traffic	385.8 KB	758.3 KB
Maximal deviation of sent traffic	6206 KB	17204.5 KB
Average number of received packets (per connection)	1393	1436
Standard deviation of received packets	4246	6042
Maximal deviation of received packets	99301	138391
Average number of sent packets (per connection)	1239	1151
Standard deviation of sent packets	3753	4542
Maximal deviation of sent packets	96093	99785
Average size of received packets	0.91 KB	1.01 KB
Average size of sent packets	0.13 KB	0.16 KB

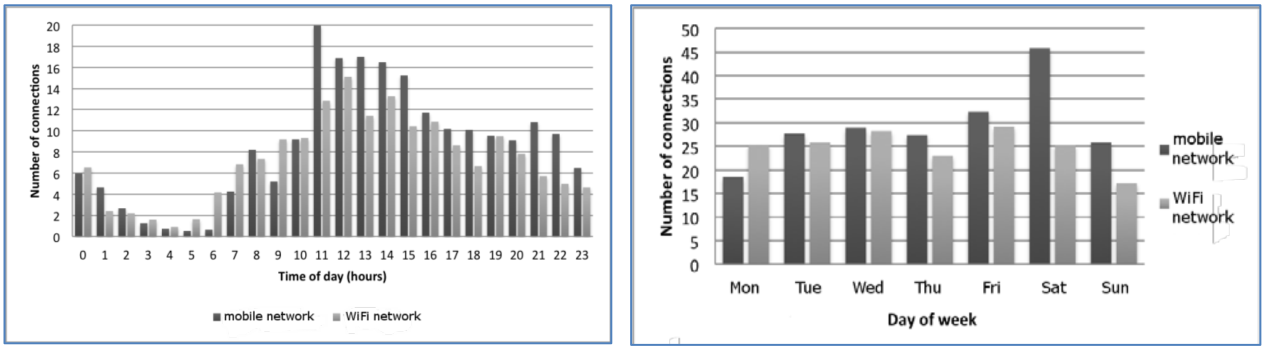
The data from the table shows that the daily number of connections made using the mobile network as carrier is somewhat bigger compared to the number of the connections made using the WiFi network interface (6.9 vs. 5.8). The users spent most of their time in the day on WiFi network (151 s more than in the mobile network), and the duration of one connection made using the WiFi interface is 131 s longer. However, the amount of daily traffic is bigger for the mobile than for the WiFi network. Users have received 310.3 KB and sent 88.3 KB more on mobile network. When observing the number of packets on daily basis, users have sent 1259 more packets, whereas have received 1850 more packets, on WiFi network compared to the mobile network. The average amount of received/sent traffic per connection is also bigger for WiFi network. On WiFi network, users have received 43 packets more per connection, while they have sent 88 packets more on mobile network. The average size of the received and sent packets is moderately equal for both types of interfaces. Please note that compared to previous studies (as discussed in Section 1) these results show that smartphone users tend to use the mobile interface for data traffic even more than just recently starting to surpass the WiFi traffic.

The logging application also differentiated the traffic on the mobile network interface according to the following types of mobile networks: GPRS (2.5G), EDGE (2.75G), UMTS (3G), HSDPA (3.5G) and HSPA (3.5G). The division of the Internet traffic realised on a mobile network, according to the type of the mobile network is shown in Table 4. It can be noticed that the majority of the connections were established on HSPA and HSDPA mobile networks, showing that the users preferred to connect to mobile networks with higher data transfer rates. The number of daily connections and the amount of daily received Internet traffic is bigger for the HSPA network, whereas, the users had longer average connection duration and they sent larger amount of Internet traffic on a daily basis using the HSDPA network. As expected, the users tend to avoid slow connections, such as GPRS, when sending or receiving Internet data. However, the results depend not only on users' habits, but also on the offered type of connection by the mobile operator and the user tariff models.

Table 4 Main characteristics of data traffic depending on the offered service

Property	**GPRS (2,5G)**	**EDGE (2,75G)**	**UMTS (3G)**	**HSDPA (3,5G)**	**HSPA (3,5G)**
Average number of daily connections	0,06	0,74	0,25	2,75	3,08
Average overall Internet connection duration (per day)	30 s	501 s	156 s	2298 s	893 s
Average amount of received traffic (per day)	10,7 KB	420,3 KB	89,7 KB	3894,6 KB	4332,9 KB
Average amount of sent traffic (per day)	3,4 KB	82,2 KB	16,6 KB	539,3 KB	493,8 KB
Average number of received packets (per day)	20	552	117	4447	4451
Average number of sent packets (per day)	22	560	112	3962	3869
Average connection duration	468 s	680 s	635 s	835 s	290 s
Standard deviation of connection duration	842 s	1759 s	1988 s	2642 s	655 s
Maximal deviation of connection duration	3491 s	18963 s	20971 s	57686 s	7762 s
Average amount of received traffic (per connection)	165,8 KB	571,3 KB	364,5 KB	1415 KB	1405,4 KB
Standard deviation of received traffic	482,9 KB	1807,9 KB	1035,1 KB	4465,6 KB	5885,5 KB
Maximal deviation of received traffic	2661,2 KB	22647,7 KB	6328,5 KB	94906 KB	118807,6 KB
Average amount of sent traffic per connection	53,1 KB	111,7 KB	67,3 KB	195,9 KB	160,2 KB
Standard deviation of sent traffic	131 KB	341,8 KB	165,5 KB	404,7 KB	391,1 KB
Maximal deviation of sent traffic	738,9 KB	5092,3 KB	1331,7 KB	5078,1 KB	6210,8 KB
Average number of received packets (per connection)	314	750	475	1616	1444
Standard deviation of received packets	702	2008	1188	4010	4956
Maximal deviation of received packets	3125	27354	7802	73978	99250
Average number of sent packets (per connection)	335	761	454	1439	1255
Standard deviation of sent packets	734	2043	1222	3467	4408
Maximal deviation of sent packets	3207	30230	9904	74761	96077
Average size of received packets	0,53 KB	0,76 KB	0,77 KB	0,88 KB	0,97 KB
Average size of sent packets	0,16 KB	0,15 KB	0,15 KB	0,14 KB	0,13 KB

The number of connections in different hours of the day for an average user is presented in Figure 6(a). Users tend to make most of their connections between 11 AM and 2 PM. The maximum number of connections on the mobile network is at 11 AM, while on WiFi is at 12 AM. During the week, via the network operator the users made most of their Internet connections on Friday and Saturday, whereas, using WiFi, the biggest number of connections is on Wednesday and Friday (see Figure 6(b)). Please note the prominent usage of the mobile Internet on Saturdays, which could be due to the users' tendency for more leisure outgoing activities during this day.

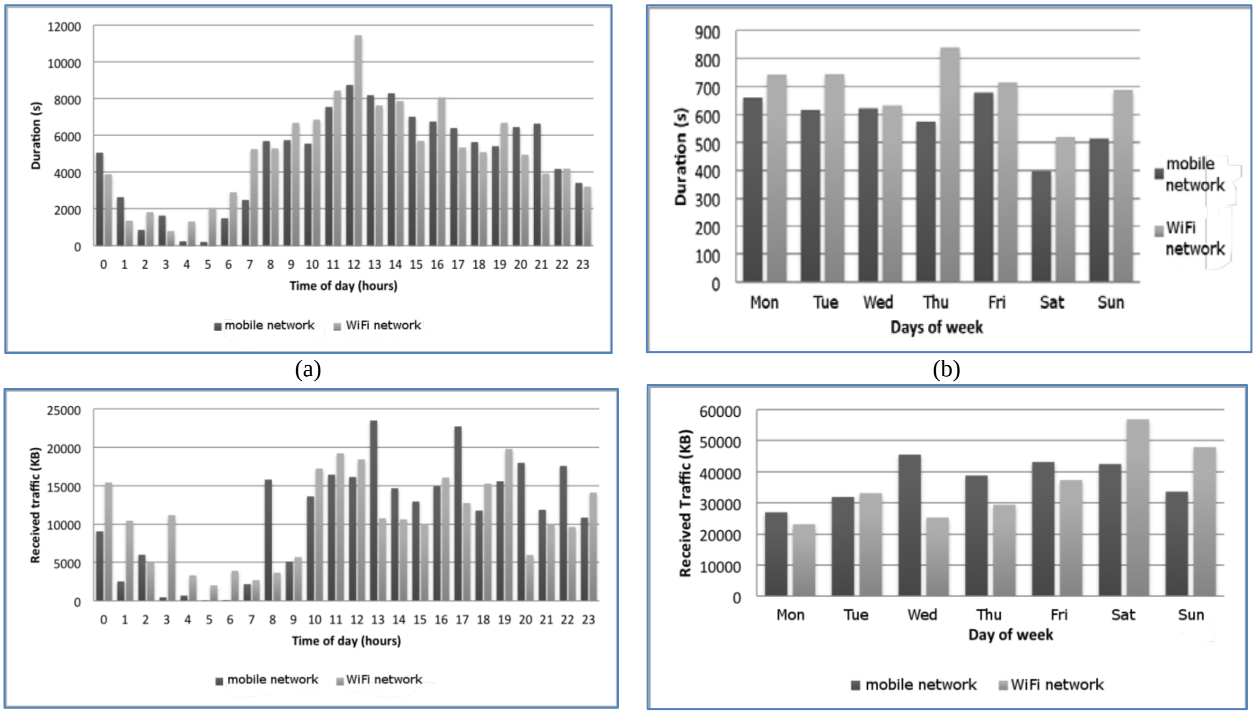


(a) (b)

Figure 11: (a) Average number of connections in different hours of the day, (b) Average number of connections in different days of the week

The duration of the connections is longest between 11 AM and 2 PM, reaching its maximum at noon, see Figure 7(a). These results show that the users are most active during their break or lunch time. Similarly, for the average duration of the connection in different days of the week, when using WiFi the users tend to have longest connections on Thursdays, while when using mobile network on Fridays, whereas the shortest duration of the connections is noticed on Saturdays (see Figure 7(b)).

Concerning the amount of traffic, here we present the results for the amount of received and sent traffic using WiFi and mobile network. When connecting via WiFi networks, users have received most of their traffic between 10 AM and 12 AM and the maximum peak was achieved around 7 PM, whereas when the users connect via the mobile network, most of the traffic they have received was at 1 PM, 5 PM and 8 PM. The results are shown in Figure 7(c). If we analyse the behaviour during the different days of the week, most of the received traffic using WiFi network was on Saturdays and Sundays, while using the mobile network on Wednesday, Friday and Saturday. When analysing Figure 7(d), it is noticeable that the users have received more traffic during the weekends and on Fridays, while Monday is the most inert day of the week.



(a) (b)

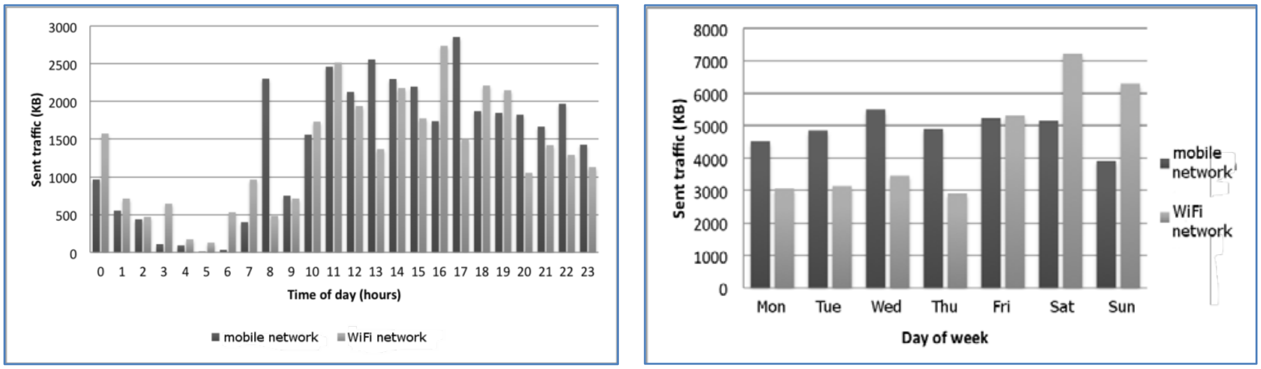
(c) (d)

Figure 12: (a) Connection duration for different hours of the day, (b) Connection duration for different days of the week, (c) Received traffic from all connections in different hours of the day, (d) Received traffic from all connections in different days of the week

Similarly to the received traffic, the traffic generated from the users on mobile networks was in its peak at 11 AM, 1 PM and 5 PM. On WiFi networks the peaks were reached at 11 AM, 4 PM and 6 PM (see Figure 8(a)). The results for the sent traffic from the users in different days of the week are quite similar to the results for the received traffic (see Figure 8(b) and Figure 7(d)). While on WiFi, the users have sent most traffic on Sundays and Sundays, while mostly on Wednesdays and Fridays they sent their traffic using the mobile network as carrier towards the Internet gateway.

Our analysis of the Internet traffic also includes data on the usage of different applications. Thus, we measured the relative popularity of mobile applications, similarly to [6]. Overall, the users have used 294 different applications that have generated Internet traffic. On average, 61 app relied on traffic every day. In

Table 5 we show the main characteristics of the applications usage. The average traffic for one app was 231.4 KB of received data and 27.5 KB of sent data.



(a) (b)

Figure 13: (a) Sent traffic from all connections in different hours of the day, (b) Sent traffic from all connections in different days of the week

Table 5 Main characteristics of app usage

Property	sent	received
Average amount of traffic for all app (per day)	14118,7 KB	1678,6 KB
Average amount of traffic for one app	231,4 KB	27,5 KB
Standard deviation of the traffic for one app	1425,2 KB	179,4 KB
Maximal deviation of the traffic for one app	105261 KB	16216,1 KB

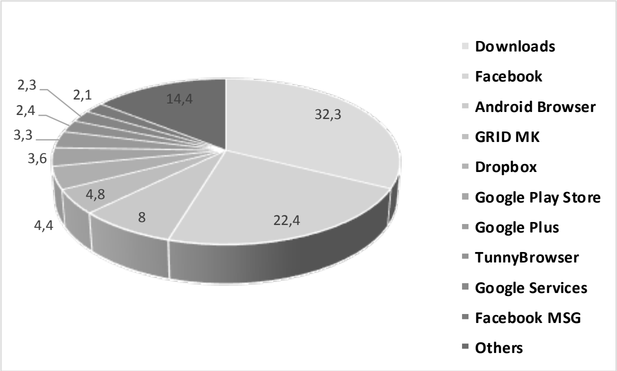


Figure 14: Top 10 data consumption applications

In Figure 9 the top 10 data consuming applications are presented. As expected, applications that are responsible for most of the overall offered traffic were: Downloads, then Facebook, Android Browser, GRID MK (local news aggregator) and Dropbox, see Table 5Usually, the overall traffic is consisted mostly of downstream traffic, i.e. from the operator to the user (around 90%, see Table 5), thus, we have obtained almost identical results for the received traffic as for the overall traffic. Also, applications that used most of the upstream traffic were Facebook, Dropbox, Facebook Messenger, Android Browser and Google Plus.

3. MODELING SMARTPHONE TRAFFIC

One of the first steps when analysing wireless mobile networks, regardless of the different infrastructure-(less) and technological possibilities, is developing a simulation environment that will as closely as possible reflect a real life scenario for the user usage of the network. After deciding on the various protocols on the different network layers, there are two more modelling decisions that need to be made which will greatly influence the obtained results. These are the mobility model that defines the nodes positioning and movement in the observed environment and the traffic model that defines the traffic patterns in terms of packets that are going to be exchanged between different pairs of nodes at given moments. However, in most of the cases these two characteristics are often neglected and overlook, while researchers habitually use a random node movement that does not reflect any realistic human behaviour in combination with a random traffic generator that creates randomly sized packets at random time intervals choosing random destinations in the network. The results obtained with simulations based on this type of modelling of the user behaviour can be very far from the real life applications of the network, thus misleading when trying to analyse the network performances or when inspecting the behaviour of a new control or routing protocol. It is imperative that when designing simulations of wireless mobile networks, wherein it is expected that in real life conditions the network nodes will be devices that will be used by humans, the mobility and traffic patterns are modelled as realistically as possible. Thus, in this section we present how the results from the previous section can be used towards creating a more realistic synthetic traffic pattern for mobile users. Please note that this type of traffic pattern does not necessarily has to be constrained for the purposes of simulation of mobile network operators' networks. Since we observe the traffic pattern on its highest application layer, the same traffic pattern can be applied to any underlining network architecture as long as its goal is to provide peer-to-peer communication between its nodes together with a gateway to the Internet. Thus, the smartphone traffic generator we propose here can be used in simulations of 3G and 4G networks, but also Wi-Fi and mobile ad hoc networks and alike.

The observation and statistical analysis of the behaviour of different smartphone users over an extended period of time can be used as a cornerstone for the creation of a realistic smartphone traffic model that can then be translated into a smartphone traffic generator. In this way, instead of using overly simplified random traffic generators that are a very crude replacement for the real expected traffic between the users, we can try to use the available abundant information on the users created traffic and model the network traffic more realistically. However, it must be taken into consideration that the creation of a traffic generator that will perfectly imitate the user behaviour is an exceedingly tedious task that can hardly capture every aspect of the observed communication pattern between the users in correlation with other factors like time of day, location, social connections, age, etc. Thus, what we propose is a first order approximation that should produce a more human like traffic compared to the pure random generators. One can always build on top of this approximation in order to put an accent on a certain aspect of interest, like for an example, the correlation between the source-destination pairs and the underlying social network of the users.

When deciding on the mechanisms for our synthetic smartphone traffic generator, we define two possible models that make use of the previously gathered user data. The first model is based on the statistical characteristics of the averaged traffic of all of the observed users over the complete period of observation. Following this line of reasoning, we can obtain functions that will represent the statistical distribution of different events of interests and then use these functions as input to random generators that will take samples from the distributions for all different types of events. By combining the samples we can obtain a synthetic traffic representation of the originally observed user behaviour. The second model is to use user emulation instead of synthetic traffic generation. The emulation will provide the basis for imitating the behaviour of the average user. However, in order to use this approach we must first discover who the average user is among the observed set of users. Towards this goal we decided to choose the average user as the user whose obtained traffic pattern characteristics are closest to the average value of the main event characteristic. In the following subsections we will consider both of the proposed models and discuss their positive and negative sides. The output of both of the models will be a traffic pattern scenario that can be fed into a network simulator and will be used to define the traffic that traverses between the nodes in the network on the application level. One must take into account that the traffic patterns for the simulation scenarios that will be provided are going to be based on results obtained using daily traffic, i.e. it will reflect the daily observed patterns of communication between the users. However, by adjusting the time scale, one can easily translate the 24h traffic into a shorter or wider timeframe as necessary.

In order to move towards a synthetic traffic generator based on the gathered data that describes the smartphone traffic pattern created by our sample users, we firstly divide the traffic event into three separate groups that we model with different types of traffic and different type of services and QoS: voice traffic, message traffic and data traffic.

3.1 Voice Traffic Representation

In Figure 10 the cumulative probability distribution function (CDF) for the number of call events per user per day according to the gathered data in our study is presented. Using the distribution function one can draw conclusions on the percentage of users that have made up to a given number of calls (e.g. 66% of the users have made less than 15 calls per day). For our synthetic traffic generation purposes, we can use this function by taking a random sample value for each node in the network that is going to be represented in our simulation. The obtained random sample will define the number of voice call events for the given node. According to our previous analysis (see Section 2, Table 1), 36% of the total number of call events are outgoing calls made by the user. Thus, the traffic generator will decide that during the simulation the given user will need to make 36%*(random sample from CDF) outgoing calls to randomly chosen destination nodes. In order to define the rest of the voice call characteristics, other additional distributions are needed.

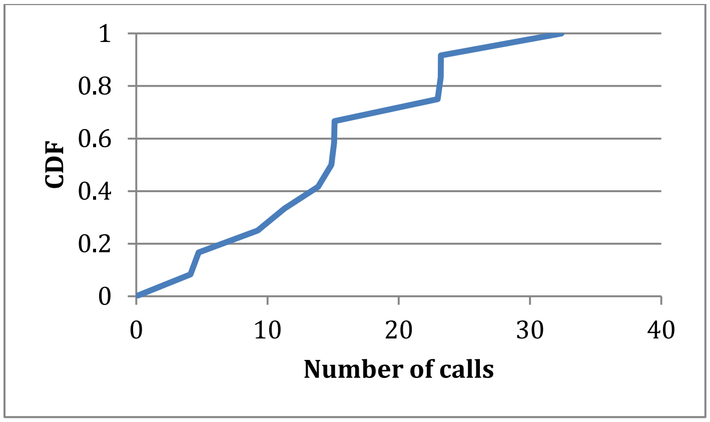


Figure 15: Cumulative distribution function of the number of calls per day

In Figure 11 the CDF for the call duration is presented. According to the results, 90% of the calls last less than 200s. For the purposes of creation of a realistic simulation scenario, each of the call events defined by sampling the previous CDF for the number of calls per day has to be sorted by type using the results given in Table 1, and afterwards for each outgoing call we need to sample the CDF given in 0In this way we obtain the number of outgoing calls per user for each node in the simulation together with the duration of each individual outgoing call. In order to complete all of the necessary information for the voice call events we need the starting moment of the calls. For this parameter we decided to use uniform random distribution that will take samples from the (0, end_time_of_simulation) in order to accommodate, and in a way scale, to any duration of the chosen simulation time. A more realistic approach would be to obtain a third CDF for the call start time, however we believe that this will decrease the usability of the generator since it is very rare that researchers create wireless mobile simulation scenarios with simulation time longer than a few hours.

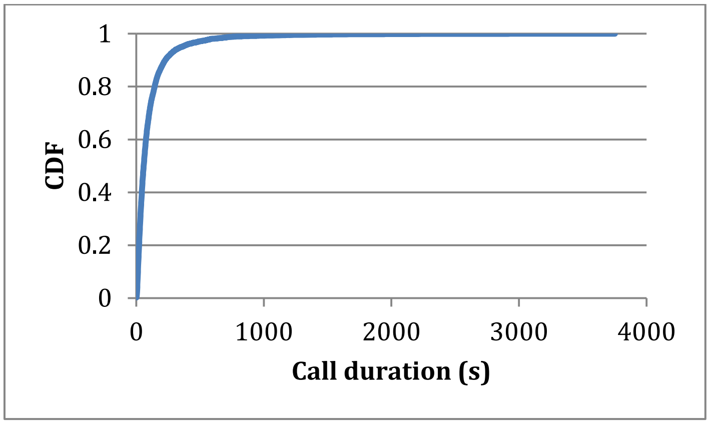


Figure 16 Cumulative distribution function of the call duration per day

The voice events simulation traffic can be created by emulating the data obtained from the user that has the closest average value to the average value of the total calls duration for all users per day. Using this approach, instead of sampling the number of calls and their duration, one simply looks them up in a table and scales each call event (in order to adapt to any given simulation time). Thus, we obtain a traffic pattern scenario wherein we simulate the behaviour of typical "average" nodes in the network. In order to provide more distinct node behaviour, different nodes can reflect the "average" user behaviour from different days randomly chosen from the full set of days that were gathered during the observation period. In this way, the different user behaviour in different days will realistically reflect in the simulation scenario.

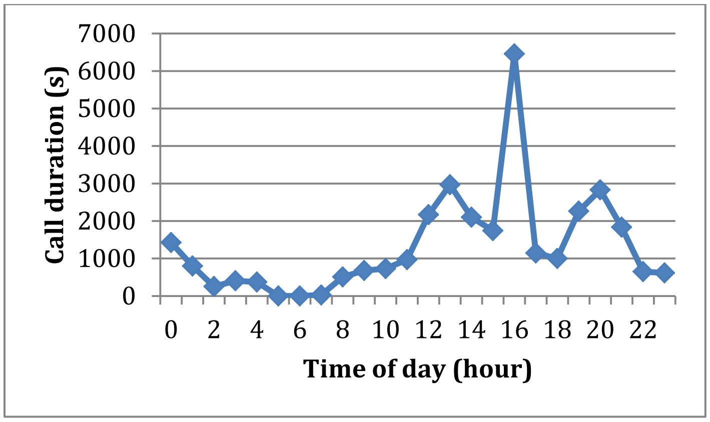


Figure 17 Average call duration in different time of day for the "average" user

Figure 12 presents the average call duration for different time of day of the chosen "average" user. The typical peaks of call duration at 1, 4 and 8 PM will also reflect in the simulation since the generated traffic will be a scaled mirror like version of the users' behaviour.

While the second type of modelling using an "average" representative of the pool of user data seems more adequate for more realistic traffic patterns, the biggest problem that arises is choosing the right representative. The results have shown that if we decide on a different parameter (e.g. average number of calls, average call duration, average number of outgoing calls only, etc.) we end up with a different candidate user whose average value is the closest to the total average, i.e. the representative user changes with the change of the observed average parameter. Thus, when deciding on using this method of modelling application traffic, one must first decide on the parameter that is of highest importance in the simulation scenarios, whether it is the number of calls, or the call duration, or some other value.

On the other hand, the average statistical method offers a more straightforward way of integrating a more humanlike behaviour into the traffic pattern scenario for a given simulation. However, the obtained traffic patterns are further away from real life scenarios since they are based on total averaged functions. This is even more pronounced if we come back to the results from the previous section and review that the standard deviation that accompanies the total average has very large values indicating that the users are extremely heterogeneous in their behaviour. Yet, this is still a very good first order approximation that has incorporated this heterogeneous behaviour by having all of the user events values influencing the obtained CDFs.

As for the voice call event itself, it can be modelled using VoIP (over UDP/IP) packets tagged with the highest quality of service (QoS). These will be packets that are sent in uniformly distributed time intervals with different size and data rate depending on the chosen codec [18]. For example, AMR codec, which is often chosen for VoIP, can support 8 different possible data rates starting from 4.5 kbps up to 12.2 kbps. It generates 244 bit packets, which represent 20 ms voice frames. In order to provide high QoS, the packet delay must be less than 150 ms, with packet loss not higher than 1% and jitter less than 25 ms.

3.2 Message Traffic Representation

The modelling of the short message traffic is a bit simpler compared to the voice calls. In this case we need to model the communication that is a short burst of acknowledged information in a given time moment and does not need to be handled with any QoS except offering reliable service. Towards this goal we can model the sending of a message by sending on application level packet to a randomly chosen destination with a random size that can be chosen from the uniformly distributed interval of (10 B, 128B) wherein a message of 128B corresponds to the largest SMS size allowed when using all of the 160 available characters. TCP on the transport layer will reflect the reliable nature of the traffic. In the case of short messages we argue that the parameter of highest importance is the number of messages an average user sends per day and thus for the purposes of message traffic modelling we propose the use of the CDF given in Figure 13. In some special cases, one could also be interested in the specific distribution of moments of sending messages for which cases another CDF is necessary.

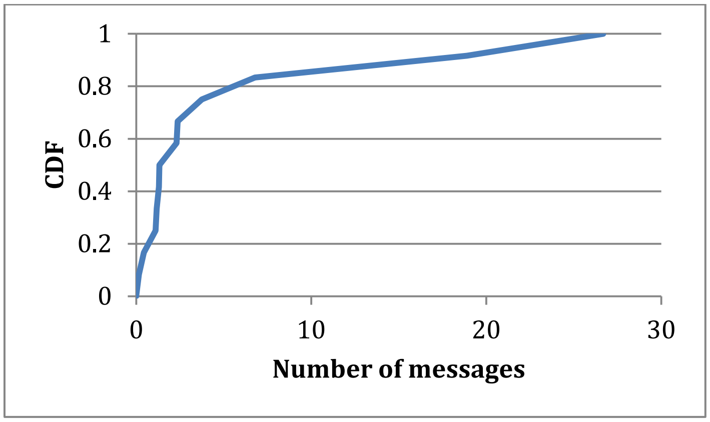


Figure 18: CDF for the number of messages per day

The result presented in Figure 13 also shows that 75% of the users deal with less than 4 messages per day. For the purposes of creating synthetic message traffic, after taking a sample from the presented CDF, we need to decide on the exact number of sent messages by referring to the results presented in Table 2.

Since the amount of traffic of short messages is sparse compared to the voice and data traffic, for modelling of message traffic we would encourage using the second proposed method of emulating an actual user experience by choosing an "average" user from the gathered data. Another interesting observation that comes into light is that the "average" user whose average number of messages per day is closest to the total average of the gathered data is the same "average" user selected for the voice calls traffic. However, since our gathered data set is relatively small we cannot draw a conclusion that this will always be the case. Figure 14 represents the distribution of messages per hour of day for the chosen "average" user whose behaviour will be reflected into the simulation scenario.

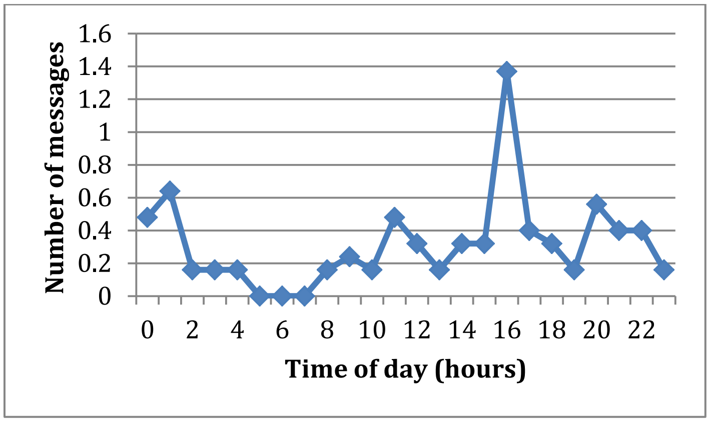


Figure 19: Number of messages per day for the "average" user

3.3 Data Traffic Representation

When simulating data traffic in a wireless mobile network scenario for the purposes of optimizing a protocol or discovering the impact of certain network parameter, researcher seldom use heterogeneous scenarios in which the data traffic can be transferred among the user nodes via different networking technologies. Thus, for the purposes of modelling and generating synthetic data traffic we will not distinguish between the different types of traffic (e.g. WiFi and 3G) and will consider the total data traffic that passes between the users. In order to simplify the simulation scenario we also do not try the divide the traffic according to type on application level, i.e. all traffic that is transferred from/to devices is considered to belong to the same application layer (we merge all applications into one). This simplification can be justified due to the fact that the generated traffic will be similar to the one observed while the application details are hidden in order to make the resulting simulation trace output more readable. Thus, we model all of the data traffic using a number of bidirectional connection channels to a given Internet gateway to/from which we send equally sized packets (the packet size is determined as the average packet size for received and sent packets, respectfully). In order to accommodate for different types of connections, for each connection we can optionally randomly choose the transport protocol (TCP or UDP).

In order to decide on the number of connections as well as the connection duration we need to sample the corresponding CDFs given in Figure 15. After sampling the number of connections for each user, we than independently sample the duration of each connection.

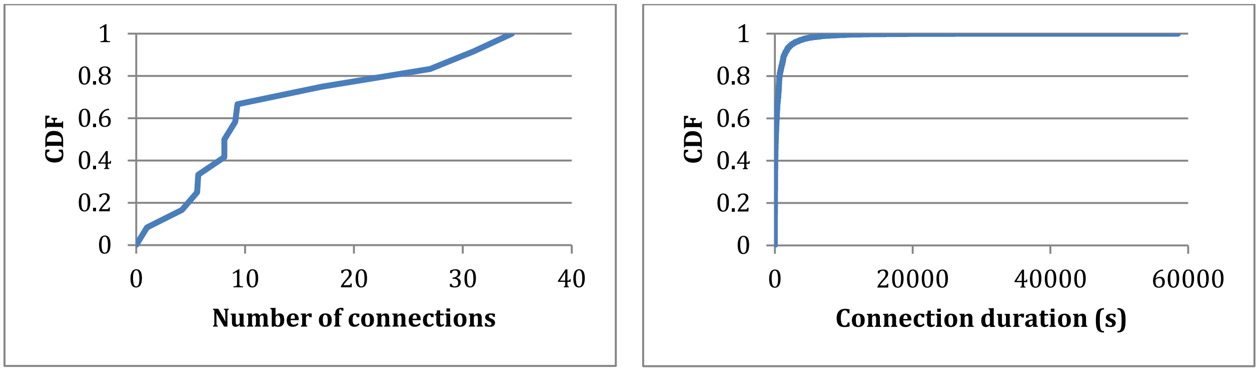


Figure 20: CDFs for the number of connections and connection duration per day

After defining the number of connections and their duration, the next step is to define the dynamic behaviour of the connection in terms of traffic received and sent. This can be done in two possible ways: by using the CDFs that describe the distribution of the amount of received/sent traffic averaged per connection per day, or by utilizing the CDFs that describe the number of packets received/sent averaged per connection per day. For simplicity, sticking to the first order approximation of real life traffic, we incline to using packets with the same size. We argue that the second option of using the CDFs that describe the number of received/send packets per connection will result in a more realistic representation inside the simulator. Thus, as a next step in the process of determining the data traffic that will be defined in the simulation scenario, the traffic generator will take samples from the CDFs that represent the number of received and sent packets per connection, as given in Figure 16, respectively.

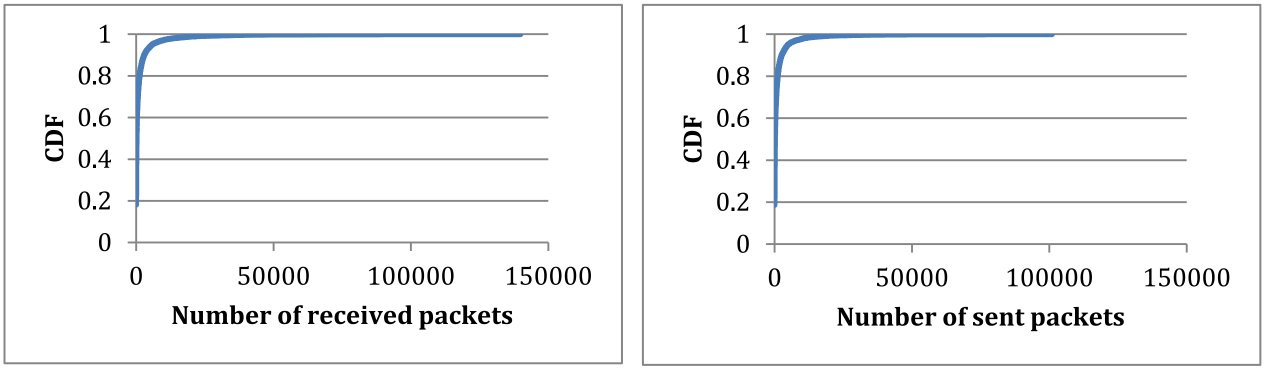


Figure 21: CDFs for the number of received and sent packets per connection

The previous discussion has already hinted that modelling a traffic generator for the observed data traffic is the most difficult task because of the non-homogenous nature of the data traffic. The distributions from which samples can be taken are numerous and which ones will be utilized will depend mostly on the goal and purposes of the simulation. We find that the one example we presented here is useful for general wireless mobile network simulations in which we can observe the network behaviour under realistic traffic on a global macro scale. Due to the highly differential nature of this type of traffic it is extremely difficult to achieve synthetizing traffic that will highly resemble the observed one. The only parameters that will fairly correspond to the real observations are the ones that are chosen to take samples from.

Additionally, the granularity of our gathered data and observations permit for other more complex scenarios that can include heterogeneous scenarios involving different types of data traffic while discerning the network interface used for data transfer, the type of application used, etc. However, for these kinds of especially detailed simulations we propose using the second method of synthetic traffic generation via emulation of an "average" user according to the parameter of interest. In this way, one can fully rely on grasping all of the traffic nuances needed for a realistic simulation scenario. For an example, Figure 17 shows the total amount of traffic received (rx) and sent (tx) for the "average" user chosen as the user closest to the total average traffic.

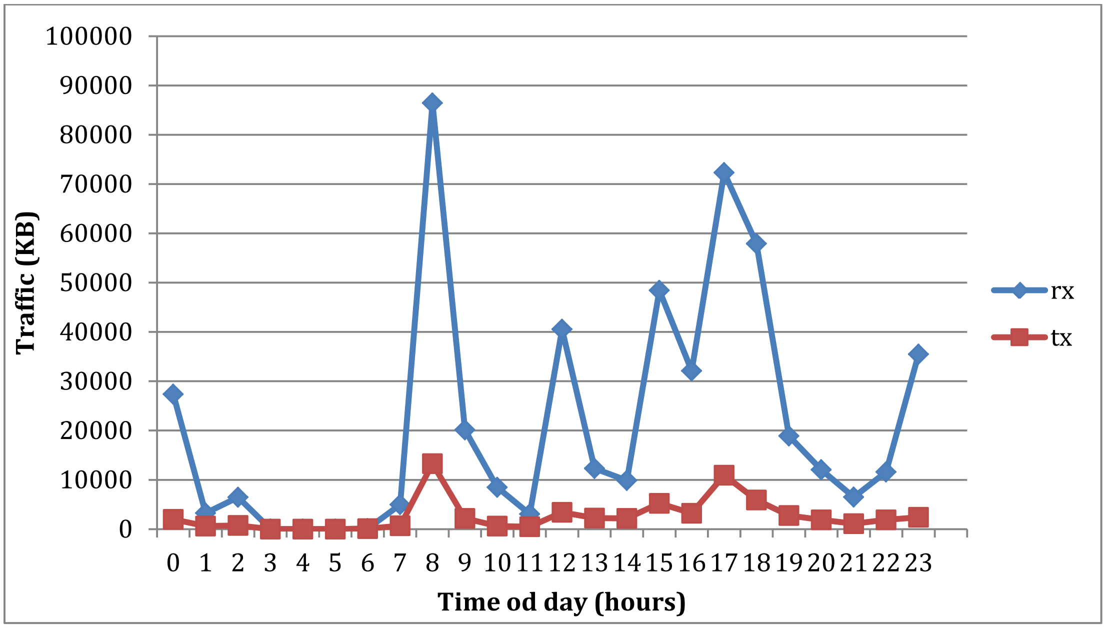


Figure 22: Data traffic exchanged at different time of day by the "average" user

3.4 Additional Possibilities and Mappings

The basic functioning of the synthetic traffic modelling discussed so far offers the possibility of generation of simulation traffic according to two types of modelling (using CDFs or emulation of average users). For each of the chosen modelling type, in order to generate data traffic the generator also needs to be fed with the particular parameters that are of highest interest for modelling. Thus, the resulting traffic scenario becomes a mix of randomly chosen values distributed according to the chosen distributions. As our envisioned generator is continuously connected to the database of gathered data from real life users, with the continuous process of data gathering and user base expansion, the data on which the distributions and emulations are based will become more reliable and will more closely reflect the major traffic properties.

Also, by adding additional mappings to the real life conditions we can obtain traffic scenarios that even closer to the observed traffic patterns. Our future efforts are focused on adding the social dimension to the generator by reverse engineering the social network among the users using the gathered voice call data. With the creation of a social graph of the users we can decide on the destination of every communication more realistically by mapping the communication intensity for each source-destination pair information exchanges.

The gathered data can be used for mapping and researching of the mutual influences between the social and underlying communication networks. The social user interconnection can be represented using a weighted directional graph that will reflect the direction of information flow as well as the frequency and duration. Our preliminary results show that the obtained social graph clearly exhibits the typical social characteristics: a small number of users talk to lots of different users, while the biggest majority call to a standardized smaller group of acquaintances. An interesting observation that arises is that the obtained graph wherein the link weights denote the call frequency and the obtained graph wherein the link weights denote the call duration are almost identical. The differences that occur can pinpoint the users that communicate while not socially acquainted at the same time, while the heavy weighted links reflect strong bonding (family members, friends, etc.).

Other possible mappings we are considering are correlating the user location and time of day in order to conclude his behaviour in different environments (e.g. work, home, out). We expect that this type of correlation should reveal more detailed patterns in the observed traffic. However, our set of users so far have been reluctant about revealing their precise location using GPS and we found that the less accurate positioning using cell towers and WiFi connection info is very coarse and does not reveal significant information so far.

4. CONCLUSION

The proliferation of smart wireless mobile devices has created an enormous and still growing change in the traffic patterns of mobile cellular and WiFi networks. And while the traffic generated from new smartphone and alike devices exponentially grows and surpasses the traditional wired and WiFi PCs networks, the new user demands for high quality wireless connections anytime, anywhere put a lot of pressure on network providers. In order to accordingly adapt to the new demands of today's modern smartphone users, network designers must understand the changes in traffic patterns induced by the smart devices.

Towards this goal, in this paper we investigated the characteristics of the overall traffic (voice calls, messages and Internet data) generated by typical smartphone users. For obtaining our data set we used a background based logging application that was collecting information on all traffic activity on the networking interfaces of the smartphone devices of our selected users. Based on the gathered observations, we have made a statistical analysis of the traffic that is generated during typical user behaviour on a daily basis. Our results have shown different patterns and inclinations of usage that help view the overall traffic characteristics in different time contexts. Thus, we concluded that users tend to use their smartphones over longer periods of interaction during their leisure time (off-work), while business communication over the smartphone is conducted via a greater amount of shorter calls and bursts of data traffic.

We used the results from our statistical analysis as input into a newly designed synthetic mobile traffic generator whose aim is to duplicate the major characteristics of our findings. The proposed generator is intended for use as a source traffic generator for wireless mobile network simulations and modelling in which a more realistic traffic is needed compared to the traditional uniformly random approach. This is especially the case when researchers want to analyse the behaviour of the network, as it would be used in real life scenarios. While designing the traffic model we offer two different approaches: using overall statistics and probability distributions of the major traffic characteristics, or determining the average user among the collected data set and emulating his behaviour. Both approaches have different benefits and pitfalls and can be used for generation of traffic with different characteristics. We also discuss possible extensions of the design that can help overcome some of the unrealistic properties of the initial design. These extensions are mainly based on examining the correlation between the users' traffic and the underlying social network as well as the location. Using these three sets of information (traffic, social connections and location) we can draw conclusions and predict the user behaviour on a very fine level.

However, the main result from our analysis is the high diversity in behaviour of the observed users and their traffic patterns. Large portion of the analysed characteristics shows very high deviations around the averaged value making it difficult to model all users using a simplified one-fits-all model. Thus, in our future work, using comparative statistical and social analysis of the gathered data, we intend to identify groups of similar users. The traffic characteristics of the identified groups can then be used as a basis for an improved socially intelligent and diversified synthetic traffic generator that will be able to capture the real life traffic characteristics in the mobile network with increased fidelity.

ACKNOWLEDGEMENTS

This work has been supported by the EUROWEB project funded by the Erasmus Mundus Action II programme of the European Commission.

REFERENCES

[1]. Erricson, "On the pulse of the networked society," Traffic and market report, (2012)

[2]. AT&T. AT&T SXSWPress Release, www.att.com/Common/docs/SXSW_Network%20Fact_Sheet.doc, (2011)

[3]. H. Verkasalo, C. Lapez-Nicola, F. Molina-Castillo, and H. Bouwman, "Analysis of users and non-users of smartphone applications," Telematics and Informatics, 27(3):242-255, (2010)

[4]. ITU reports, http://www.itu.int/ITU-D/ict/material/FactsFigures2010.pdf, (2010)

[5]. H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, D. Estrin, "Diversity in smartphone usage," ACM Proceedings of MobiSys'10, USA, (2010)

[6]. Q. Xu, J. Erman, A. Gerber, Z. M. Mao, J. Pang, S. Venkataraman, "Identifying diverse usage behaviors of smartphone apps," ACM Proceedings of IMC'11, Germany, (2011)

[7]. Y. Zing, A. Arvidsson, "Understanding the characteristics of cellular data traffic," ACM Proceedings of CellNet'12, Finland, (2012)

[8]. J. Erman, A. Gerber, and S. Sen, "HTTP in the home: it is not just about pc's," SIGCOMM Comput. Commun. Rev., 41(1):90-95, (2011)

[9]. S. Kaisar, "Smartphone traffic characteristics and context dependencies," University of Saskatchewan, Canada, (2012).

[10]. R. Becker, R. Caceres, K. Hanson, S. Isaacman, "Human mobility characterization from cellular network data," Communications of the ACM, (2013).

[11]. C C. Tossell, P. Kortum, C. W. Shepard, A. Rahmati and L. Zhong, "Getting Real: A Naturalistic Methodology for Using Smartphones to Collect Mediated Communications, "Advances in Human-Computer Interaction, (2012).

[12]. H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A first look at traffic on smartphones," Proceedings of the 10th Annual Conference on Internet Measurement, IMC '10, Australia, 281-287, (2010)

[13]. H. Pucha, S.M. Das and Y.C. Hu, "The performance impact of traffic patterns on routing protocols in mobile ad hoc networks," Computer Networks 51, 3595–3616, (2007).

[14]. R. Ling, The Mobile Connection, Elsevier, Dresden, Germany, (2004)

[15]. D. C. Ramírez, "Smartphone traffic patterns," KTH Information and Communication Technology, Stockholm, Sweden, (2011).

[16]. T. Neame, "Characterisation and modelling of internet traffic streams," Doctoral dissertation, Department of Electrical and Electronic Engineering, University of Melbourne, (2003)

[17]. National Agency for Electronic Communication, www.aek.mk, public reports, http://www.aec.mk/index.php?option=com_content&view=category&id=54&Itemid=123&lang=mk

[18]. 3GPP TS 26.090: "Transcoding functions".

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC Transactions on Networks and Communications

Vol. 1 Isuue 1

Virtualization in Networks: A Survey

Moiz Arif, Abdullah Nafis Khan, Muhammad Saad Iftikhar
School of Electrical Engineering and Computer Science
National University of Sciences and Technology, Islamabad, Pakistan
{moizarif2002, abdullahnafis}@hotmail.com, m.saad.iftikhar@gmail.com

ABSTRACT

Virtualization is considered an integral part of any network. Many new features and services can be introduced with a mere implementation in software and without deploying extensive hardware. In this survey we will be looking at the history and motivations that led to the realization of virtualization techniques. We will be studying its architecture, current technologies followed by challenges and future of virtualization.

Keywords – *Virtualization, VLAN, VPN, Cloud Computing, Virtualized Data Centers.*

1. INTRODUCION

Virtualization may be considered as implementing or creating a virtual version of any service/technique. Virtual implementation has many benefits as compared to the practical real implementation. Many new techniques are implemented in test beds which are set up virtually, may be in a single system implemented as virtual machines. Old implementation of virtualization dates back to the concepts of Virtual LAN's. We still see its implementation in today's environment. The motivations behind the idea of virtualization becoming a reality were to improve scalability, improve hardware resource utilization and of course to centralize management and administration of resources.

Virtualization can take any form; it is not just limited to one aspect of networks. Some types include Hardware, Desktop, Software, Memory, Storage, Data and Network Virtualization. Virtualization has found its implementation in all the walks of life, starting from networks to medicine, defense and sports etc. Legacy virtualization was present in the form of VLAN's, VPN's and Overlay networks. Nowadays we are seeing virtualization at a new level with cloud computing, virtualized data centers, data warehousing, software defined networking and many more [1].

Moiz Arif, A. N. Khan, M. S. Iftikhar; *Virtualization in Networks: A Survey*, Transactions on Networks and Communications Volume 1, No 1 (2013), PP 41-55
URL: http://scholarpublishing.org/index.php/TNC/article/view/25/TNC-13-1025
Publication Date: December 19th 2013

2. LEGACY VIRTUALIZATION TECHNIQUES

2.1 Virtual Local Area Network (VLAN)

Before the invention of VLAN's, scientists and network administrators were facing issues which were related to the increasing number of network users and distributed administration. Due to the fact of increasing network, there was a need to connect multiple Ethernet networks together and to administer it centrally or locally. Around the year 1985, there was no unique and safe way to connect multiple Ethernet networks without addressing the issues of Ethernet being a single broadcast domain, administration and security issues. A number of other techniques were also proposed to achieve this task. Such as IP Routing to connect multiple networks together. However, this was achieved at the cost of deploying hardware way costlier at that time. Dr. W. David Sincoskie, was working at that time in Bellcore and was dedicated to find a solution to this problem. In the process of doing so he came out with the self-learning Ethernet switch which solved this problem. However, Sincoskie also found out that the implementation of this Ethernet switch would be in a redundant fashion to ensure redundancy and off course with multiple links connecting Ethernet networks together. This implementation required a Spanning Tree configuration causing less resource utilization and a centralized point of failure and congestion. This very issue restricted the scalability issue. So having found so, Sincoskie set out to develop a unique solution and invented Virtual LANs. He accomplished this by adding a tag to Ethernet packets and making the Ethernet switches smart enough to handle different families of tags, thus, creating separate different virtual instances of networks all connected together over the single Ethernet channel. Link aggregation was also implemented with this technique to ensure better network availability and speeds.

Nowadays, we know this tag as the VLAN tag present in the Ethernet Header. VLAN's have, in the modern times, found application not just addressing the main motivations for its inventions but many new implementations which as per the demand of the time got developed and VLAN's were polished and developed even further.

Another motivation that led to the invention of VLAN's was that in the old times, users were grouped into networks based on their geographic location and Ethernet implementation techniques. Two users on the opposite side of the globe can be logically made a part of a single Ethernet network using VLAN's irrespective of the geographic location and topologies as well as implementation techniques. Asynchronous Transfer Mode (ATM), Ethernet, Fiber Distributed Data interface (FDDI), Infiniband & HiperSockets all are capable of implementing VLAN's and thus solves almost all the problems being faced by the researchers in the year and before the invention of VLAN's. VLAN's also provide increased security and easy network management by logically grouping users as a part of different Ethernet Networks [2].

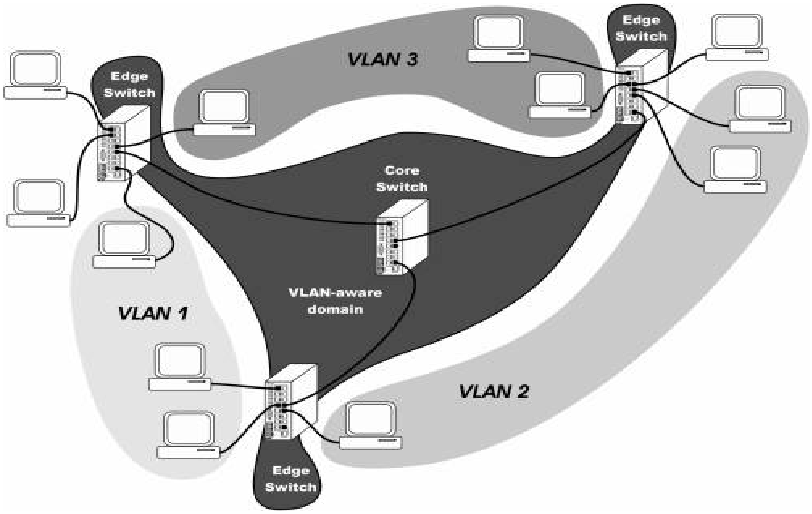


Figure 23: Example VLAN implementation over three geographically distant networks [3]

2.2 Virtual Private Network (VPN)

Virtual Private Network extends a local private area network on the WAN links to users geographically apart by using the public or shared networks. Users who are a part of a single VPN can share files, resources and perform all the functions that the users of a private network perform. End users are not aware of the implementation and the fact that they are being a part of a single Private LAN over multiple other public and shared networks. End users see themselves as a part of a single network. VPN offer us with enhanced features such as increased services, better security and better network management.

Virtual Private Networks exists in many forms and types based upon its implementation and type of service it provides. We can categorize VPNs under OSI Architecture as being Layer-2 and Layer-3 VPNs. In a more general sense we categorize VPN by level of security in provides, live or remote connectivity, termination point (customer or network end) and by the protocols it uses to channel and route traffic. Legacy implementation was rather different from current implementation. In the old days, VPN connectivity was generally provided through dial up connections over leased lines acquired from operators using frame relay or ATM technology. If we compare these implementations with today's implementation we won't call this implementation to be a true VPN based implementation. Actual implementation includes IP Based and MPLS based VPNs provided over DSL Lines or fiber optic cables providing high data rates and speeds. Such implementations are cost efficient as well [4].

Explaining further, we may notice that VLANs and VPN are practically the same as they provide the same features of bringing end users over a single private network irrespective of their geographic location. Here there is a difference; VLANs are a subnet of VPNs. More specifically VLAN are Layer-2 VPNs. VLAN may spread over a small area like a Metropolitan Area Network; however, VPNs generally extend over WAN networks. VPNs also allow connectivity between two similar networks over a different network, such as connectivity between two IPv4 networks over an IPv6 Core and vice versa. This is achieved with the concept of Tunneling. Virtual Private Networks offer many new and enhanced security features that were developed after the first implementation of VPN. VPNs usually provide security by tunneling protocols coupled with security protocols such as encryption. Encryption provides over man in the middle attacks, as only the sender and the destined user will be able to decrypt the sent data. Encryption is another domain that is way too complex to discuss here. Apart from encryption we also have authentication procedures as well. After the message is received its integrity is also verified in order to avoid data from being tampered along the path over the Public Internet.

In the modern word, VPNs are being used at a whole new level. VPN's are nowadays being used in environments where end users roam around in the network and are subject to mobile IPs may belonging from different subnets. Such implementations provide remote/live access to critical and important applications and services to intended users by providing them with access to their home network. A lot of issues are faced by implementation engineer regarding this technique. A new research is underway which deals with another method of identifying hosts on the move or mobile hosts a part from their IP Addresses to provide VPN Access. This technology is known as Host Identity Protocol (HIP) which provides VPN services by use host/device identification techniques.

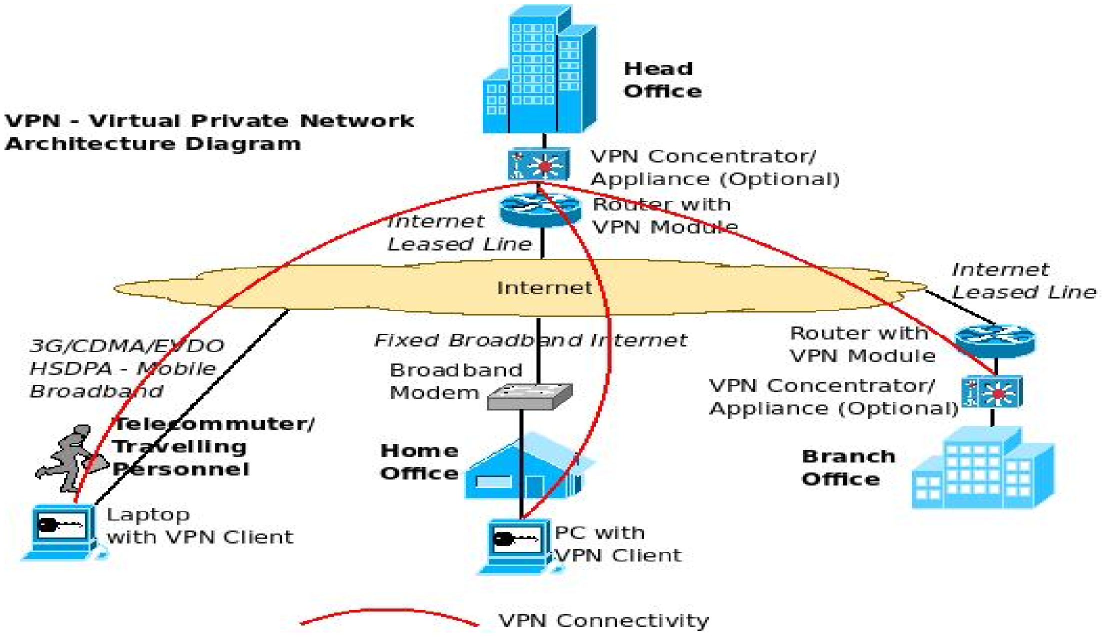


Figure 24: Example of VPN implementation over the public Internet [5]

2.3 Overlay Networks

Overlay networks are networks built on top of other networks. Overlay networks may be connected to existing networks by logical, physical or virtual links. Overlay networks uses the functionality of underlying networks are provide new features. Overlay networks are a good example of virtualization. As such networks may not involve deployment of physical network or infrastructure or modify any protocols or software. Overlay network on the other hand may require more processing power as compared to normal existing networks which may be physically deployed and are not meant to provide enhanced features.

The concept of overlay networks dates back to the early times of the development of Internetwork. Internetwork was known as the interconnection of various networks geographically distant. Internet is an example of overlay network built on top of existing Public Switched Telephone Network (PSTN) providing enhanced features and services such as packet switching to support the needs and requirement of the research community at that time. Internet was officially launched as a commercial utility in the 1980s. Since that time the overlay networks have evolved [4]. Overlay networks provide Peer to peer services being used for file sharing, content delivery to allow localized caching and storage to minimize delays and transportation charges, routing, security and many other services including Electronic mail service, VoIP etc.

The key benefits of overlay network services are that overlay networks need not to be deployed on every node of the network for its working. These networks are deployed centrally and other nodes access these via local or remote connectivity directly or by VPNs. Overlay networks on the other hand may increase complexity and increase processing delay as any extra virtualization layer has been added to the normal operation procedure. But this issue has been taken care of by implementing servers and nodes having increased processing capabilities and heavy storage space to facilitate a large number of users. Overlay network implementations exists in commercial, industrial and private applications. Over the years many new and advance services have been developed based on the test bed implementation on the overlay networks. Some examples of Overlay Networks are MBone, 6-Bone, the X-Bone, Yoid/Yallcast and ALMI [6].

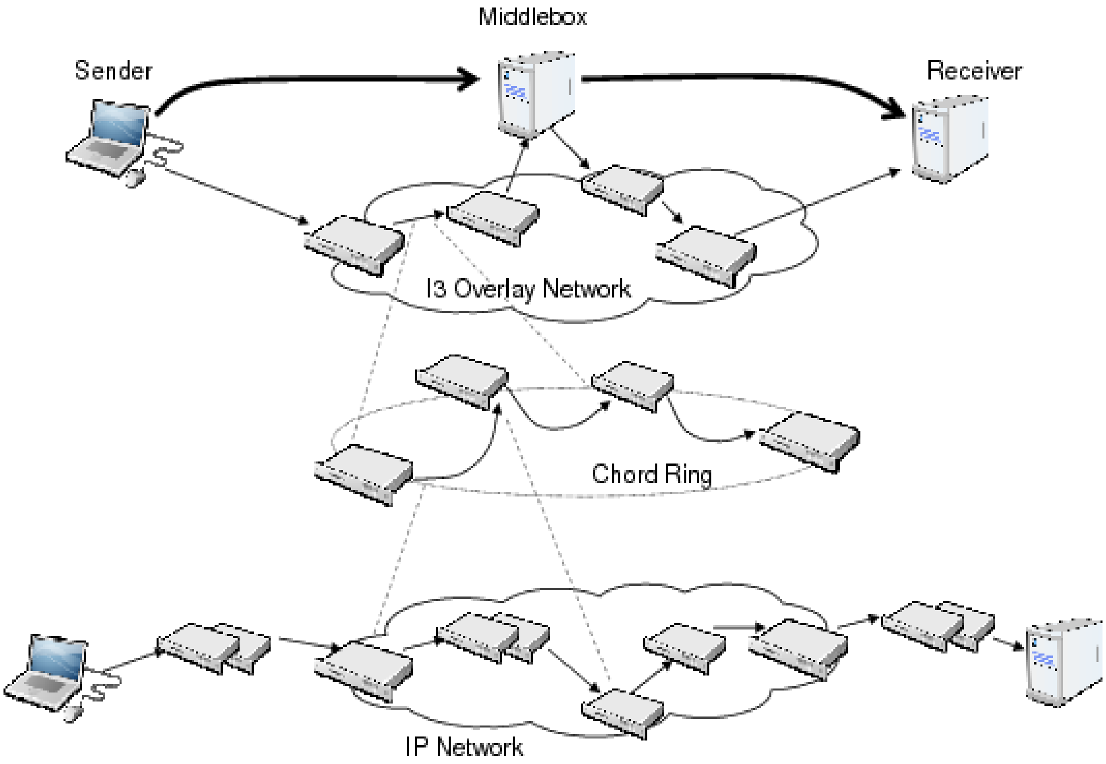


Figure 25: Example of Overlay Networks built on top of IP Core [7]

3. PRESENT & FUTURE VIRTUALIZATION TECHNIQUES

In the present, day we see Virtualization at a whole new level. Extensive features and services are provided with centralized and distributed implementations. Here we will discuss some the state of the art virtualization techniques.

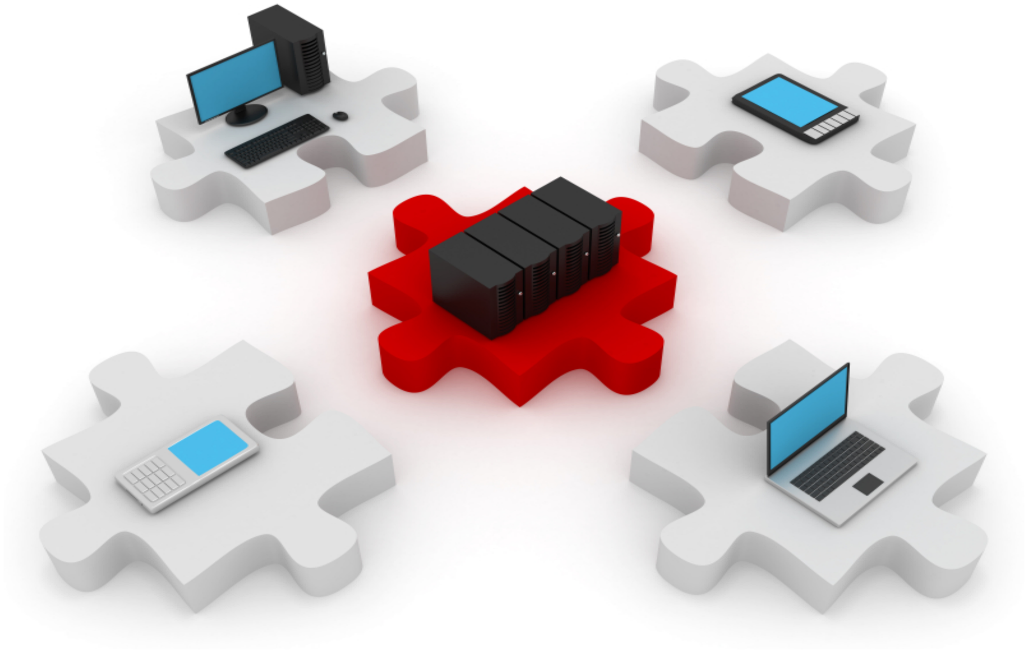


Figure 26: Generalized concepts of Cloud computing & Virtual Data Centers [8]

3.1 Cloud Computing

Cloud Computing is basically a set of services or resources being delivered to Users over an infrastructure, typically over any network. Usually the internet is the base for the provision of such services. A user can use any service such as office, storage space, any software or any feature online on the Cloud without having to install and configure software. On a generic level, cloud implementation is viewed as a single abstract cloud from the user perspective; however, from a network professional's perspective it contains number of routers, switches, servers and many other nodes.

The concept of cloud computing dates back to the early 1950s when many universities were equipped with large-scale mainframes which were accessed by thin client (Dumb terminals having no/less capabilities) to perform heavy/complex functions on their own. In 1960s John McCarthy said that in the future computation will be organized centrally as a Public Utility. In 1966, Douglass Parkhill's book "The Challenge of the Computer Utility", explained many modern day cloud computing features. He explained the ideas of elastic provision, online, illusion of infinite supply and many other ideas. Similarly, cloud computing concepts were also explained by numerous other scientists dating back to the 1950s and until now [9].

After the dot-com bubble, Amazon developed their data centers and ware houses and enhanced the concepts of cloud computing. After much research, in 2006, Amazon launched a commercial product to the end users under the name of Amazon Web Service (AWS) on a utility computing basis. In 2008 Eucalyptus became the first open source platform for deploying private clouds. Similarly, in the same year OpenNebula became the first open source software to implement and offer private as well as hybrid clouds. In the Mid 2008, the patch between the providers of IT Services via cloud and the end users was filled by Gartner. And presently we have evolved to a new era of Smarter Computing which was announced by IBM on March 1st, 2011. It is basically a framework that supports Smarter Planet which contains all concepts relating to Cloud Computing. Smarter Computing is a copy right IBM Specific name for the provision of cloud computing services for enterprise companies for Business use. They offer service via private, public and hybrid cloud delivery method as required.

Public cloud computing are of various types which include, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), Storage as a Service (STaaS), Security as a Service (SECaaS), Data as a Service (DaaS), Database as a Service (DBaaS) and Desktop Virtualization.

Figure 27: Example of Cloud Computing with all services centralized [26]

3.2 Virtual Data Centers

Data Centers are places where computer systems, telecommunication systems and other related equipments are housed. Such centers are redundant in any type of service they provide. Heavy storage and backup systems are deployed to provide non-stop service to consumers. Data centers can be as large as taking up the space of an entire building. Data centers consume loads of electrical. Backup power systems such as battery banks and backup generators are installed to provide 24/7 Service Availability. Large scale air conditioning is required to cool the equipment installed. Redundant links are installed to avoid faults causing services outage.

Data Center concept dates back to the old times when we use to have main frame systems and servers. With the dawn of the micro-processor industry, personal computing become very much popular and advances in centralized computing or data center halted. With the introduction of client-server computing in the early 1990s, computers started finding their way into old computer rooms. Data center technology observed a boom during the dot-com bubble period. Many companies started deploying heavy equipment in data centers and started providing industries, vendors and operators with unified services and solutions. With the introduction of cloud computing data centers started to get regularized and standardized. Proper data center requirements, designing and safety procedures were published.

Data Centers are categorized into four tiers, which are based on their functionality and availability. For Example, tier 1 is just a mere Server Room housing small scale computing equipment. This server room is non-redundant and offers expected availability of 99.671%. Tier

2 meets all the requirements of Tier 1 but provided redundancy and offers expected availability of 99.741%. Tier 3 meets all the requirements for Tier 1 and 2 with extra specification of dual power sourced and independent connectivity to the IT Room and expected 99.982% availability. Tier 4 is the most crucial mission sensitive level which fulfills requirements of all lower tiers plus all equipment being dual powered HVAC systems and offers 99.995% availability. All these specifications are standardized [10].

All solutions and services being offered by Data Centers are effectively virtualized and all the end users can access and utilize these services 24/7 from any geographical location. Virtualization in data centers are effectively divided into five levels of degrees. Degree 1 is the virtualization of a specific application or a specific task of a company's array of services and solutions and Degree 5 being the Virtualization and Automation of the entire services platform. All the degrees in between these two extremes offer and follow a hierarchical pattern. Virtualization of services and application is much cheaper than the actual real life replica of the same services. We can achieve 10 times the processing power with 1/3rd of the cost of actual implementation by including and deploying such a system in the virtualized space [11].

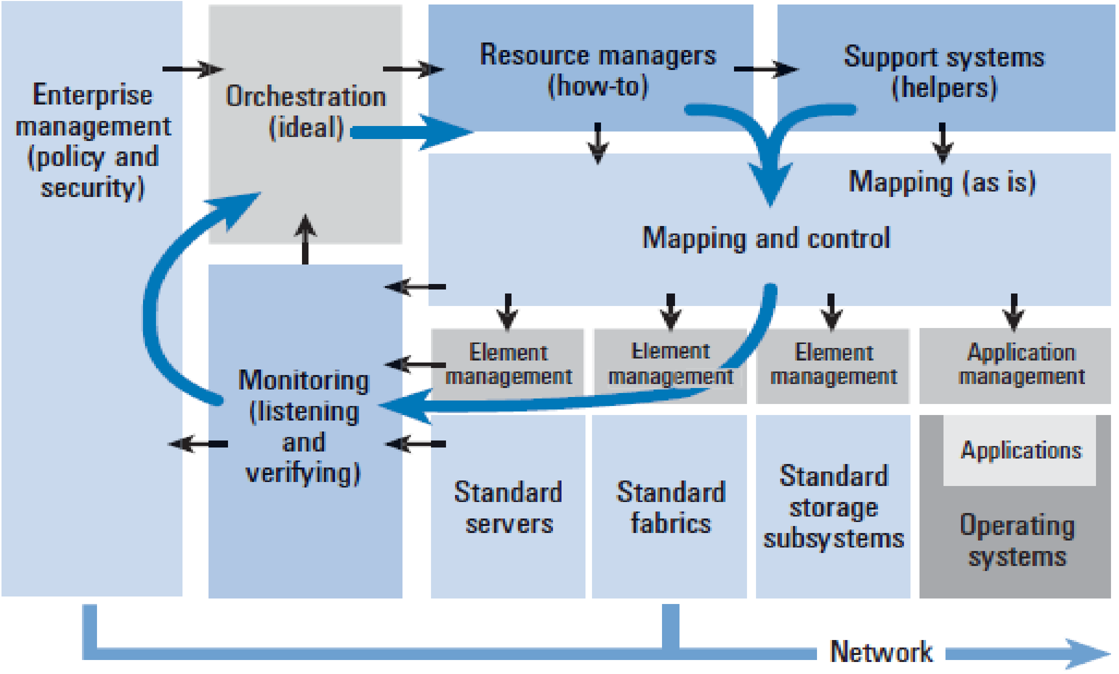


Figure 28: Example of Implementation of a Virtual Data Center [7]

3.3 Software Defined Networking

Software defined networking is an architecture specifically designed for the networking domain. This architecture is based upon the segregation of the control and data plane in networks. The control plane is implemented in software away from the networking equipment installed at different locations. The data plane is however implemented in the network equipment such as router and switches. SDNs offer many new features which include simplified management and configuration, optimization of routing policies and protocols, remote access to network equipment and much more. SDN implementation breaks the barriers of vendor specific implantation and works over all equipments of various vendors. The most popular specification for implementing SDN is the Open Flow standard.

This technique is very useful for network administrators as it simplifies networking and managing the traffic. Administrators can block users, change switching policies, prioritize packets and even block packets at granular level. This architecture is very much useful in Cloud Computing, Virtualization and management of VPNs. There are various deployment strategies for software defined networking implementation. Some techniques include Symmetric vs Asymmetric information handling, floodless vs flood based and host based vs Network-centric models. Administrators can access and configure different aspects of any network by using or logging onto a centralized console at the virtualized data center via VPNs or virtual tunnel techniques.

The motivation behind the implementation of software defined networking is fairly simple. Management of large networks which are deployed globally was very difficult task. Access to remote sites was not present which in case of an outage or fault caused a lot of time for rectification. SDN can be implemented in Data centers at any severity degree level and at any tier as per requirement.

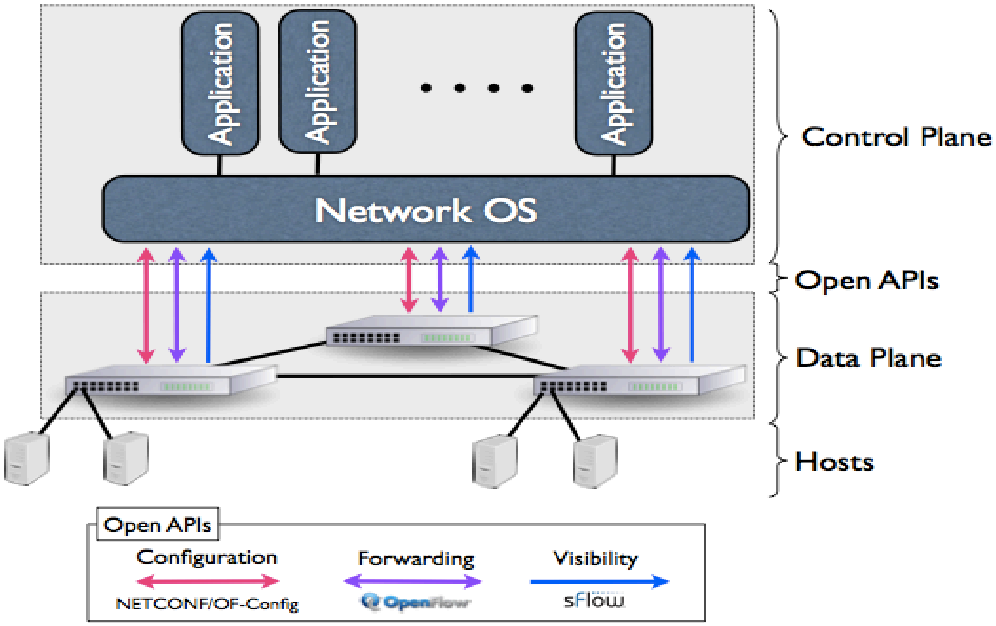


Figure 29: Example of Implementation of SDN [12]

4. CHALLENGES

There are a lot of challenges that have to be dealt with while vitalizing some enterprise environment. Virtualization of applications and software on a single server may be questioned to be as a single point of failure. Dependence on a single server may be considered as risky. This challenge is reduced with the implementation of a redundant server and redundant links. Redundant servers can run in any configuration, such as Hot Standby, Dual Homing or in a load balancing configuration. This solution comes with an increase in cost. Servers are high end computing machines and costs a lot. Another challenge that deployment specialist face is that they have to ensure that the deployment is fault tolerant and avoid network outages at all costs. Servers and systems should be self-sustaining and should have the capabilities to self-recover after any sort of failure.

Resource Starvation is also considered to be an important aspect of virtualization. Running several instances of virtualization or virtual machines can significantly increase hardware resource utilization and choke resources. It burdens the infrastructure causing delays, decreased performance and resource/service availability. I/O bottlenecks can be observed if several resources are being used from the same network interface card or any other I/O equipment. Experts have solved this problem by offloading such extensive I/O hungry operations on a separate device dedicate to perform only a specified task. This will only increase the cost by a small amount but it avoids bottlenecks which is a critical to consumers and service providers.

On many occasions it has been observed that, implementation engineers virtualized the OS itself on a server while the applications are not virtualized which can cause the application to not respond to a large number of requests. Applications have separate I/O, memory and storage requirements which are very different from the requirements of the OS itself. So when large number of users tries to access the application, it can cause the server hardware to overload and cause service outage. Engineers must ensure the resource and other requirements of all the applications being virtualized by the OS, so as to avoid unnecessary after implementation costs and outages [13].

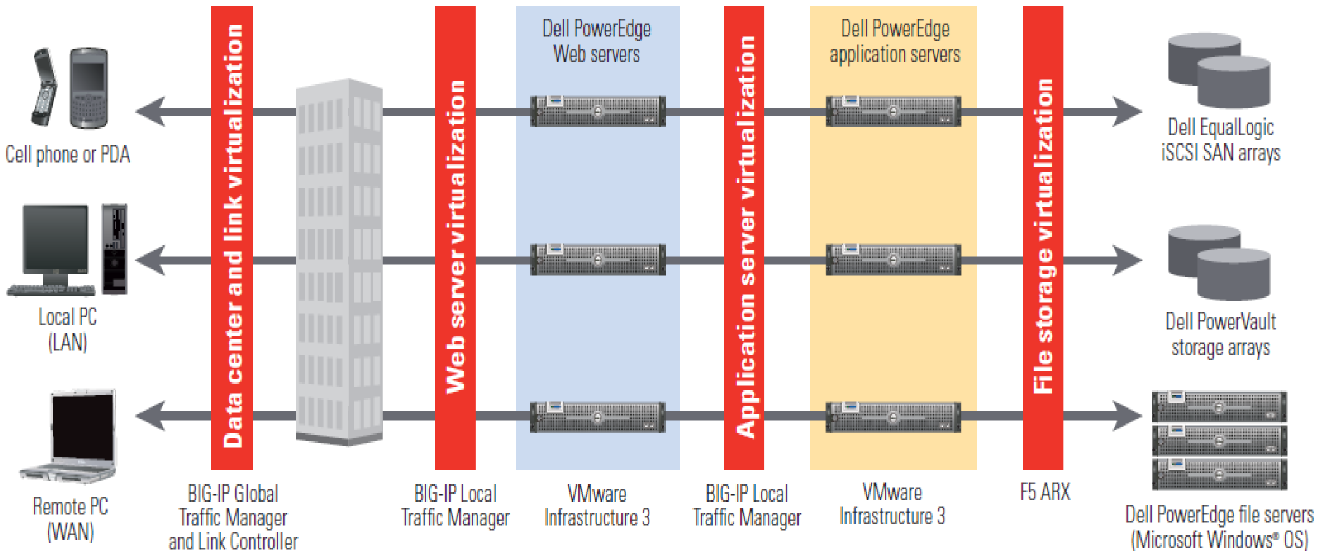

Data center and link virtualization
Web server virtualization
Application server virtualization
File storage virtualization
Dell PowerEdge Web servers
Dell PowerEdge application servers
Cell phone or PDA
Local PC (LAN)
Remote PC (WAN)
BIG-IP Global Traffic Manager and Link Controller
BIG-IP Local Traffic Manager
VMware Infrastructure 3
BIG-IP Local Traffic Manager
VMware Infrastructure 3
F5 ARX
Dell EqualLogic iSCSI SAN arrays
Dell PowerVault storage arrays
Dell PowerEdge file servers (Microsoft Windows® OS)

Figure 30: Different Types of virtualization at different stages [8]

Virtualization is implemented to reduce data center costs, minimizing power consumption and increase business productivity. In some cases, unanticipated costs can reduce the above mentioned objectives. Ineffective planning and incomplete design can cause the service provider to offload services or to increase physical servers to cater for the growing requirements of the service. Virtualizations of Operating systems need to run many instances of virtual machines causing large storage requirements for storing and running Virtual instances of OS and applications. Same is the case when the server is not being used to its intended purpose causing resource wastage and ineffective allocation of resources causing many problems. Storage requirements can increase over time as OS can occupy over hundred giga bytes of space after virtualization on shared file sharing servers [14].

Management of a virtualized environment can often turn into a complex and difficult task. Management terminals and interface can give us an insight to virtualized features, their usability, resource utilization, load information and many other performance metrics but will not provide an insight on the data center as a whole. For this administrators have to use many other utilities to manage every aspect of the data center which increases complexity, cost and time to manage a virtualized environment. The implementation, as mentioned before should be self-sustaining. Decision should be based upon data being received from different nodes and monitors should monitor difference performance metrics and deciding to offload, balance or notify for a hardware capacity increase if the limit is reached. Modern techniques and experts cater for all these challenges at the time of deployment and leave ample space for fault tolerance and scalability keeping in view the exponential nature of growth of users over the internet.

5. NETWORK VIRTUALIZATION PROJECTS

Many projects have been introduced that offer different services and platforms related to VPNs, cloud computing and Virtual Data Centers. Some of the projects are worth mentioning here. These projects are differentiated by their characteristics and features. They vary on the basis of Network Technology, layer of virtualization, Level of Virtualization and Architectural Domain [15].

Table 6: Comparison of various network Virtualization projects

Project	Architectural Domain	Networking Technology	Layer of Virtualization	Level Of Virtualization
VNRMS [16]	Virtual network management	ATM/IP		Node/Link
Tempest [16]	Enabling alternate control architecture	ATM	Link	
NetScript [17]	Dynamic composition if services	IP	Network	Node
Genesis [13]	Spawning virtual network architectures		Network	Node/Link
VNET [18]	Virtual machine grid computing		Link	Node
VIOLON [19]	Deploying on-demand value added services on IP overlays	IP	Application	Node
X-Bone [16]	Automating deployment of IP overlaps	IP	Network	Node/Link
PlanetLab [17]	Deployment and management of overlay-based test beds	IP	Application	Node
UCLP	Dynamic provisioning and reconfiguration of light paths	SONET	Physical	Link
AGAVE [12]	End to end QoS aware services provisioning	IP	Network	
GENI	Creating customized virtual network test beds	Heterogeneous		
VINI [21]	Evaluating Protocols and services in a realistic enjoinment		Link	
CABO [11]	Deploying value added end to end services on shared infra structure	Heterogeneous		Full

6. CONCLUSION

Following from the beginning of the internet to the modern day version of the internet it is evident that Virtualization has been the corner stone for development in the IT Infrastructure. Many forms of virtualization have been seen over the course of time. Each instance and version was designed to address a specific issue faced at that time. The computing trend varied from being Centralized to distributed and then back to being centralized as we see nowadays which is the basis of Virtualized data centers and data ware houses. Virtualization allows businessmen, professionals, policemen, doctors to stay connected with crucial applications and resources on the go 24/7. Virtualization has reached every scope of life with a new twist to every deployment. It has made the life of administrators simpler and easier.

REFERENCES

[1]. http://whatisvirtualization.com/.

[2]. http://en.wikipedia.org/wiki/Virtual_LAN.

[3]. http://en.wikipedia.org/wiki/Virtual_private_network.

[4]. Andersen, D., et al., Resilient overlay networks. Vol. 35. 2001: ACM.

[5]. http://en.wikipedia.org/wiki/Cloud_computing#Cloud_clients.

[6]. Geisa, E., Data centre virtualization q&a. 2006.

[7]. ABELS, J.P.A.T., Progressive Degrees of Automation toward the Virtual Data Center. 2005: p. 3.

[8]. Overcoming 7 Key challenges to virtualization: how Dell- and F5-BaseD it infrastructures smooth the way. 2009.

[9]. Chowdhury, N. and R. Boutaba, A survey of network virtualization. Computer Networks, 2010. 54(5): p. 862-876.

[10]. Chowdhury, N.M.K. and R. Boutaba, Network virtualization: state of the art and research challenges. Communications Magazine, IEEE, 2009. 47(7): p. 20-26.

[11]. N. Feamster, L. Gao, and J. Rexford, “How to Lease the Internet in your Spare Time,” SIGCOMM Comp. Commun. Revi., vol. 37, no. 1, 2007, pp. 61–64.

[12]. M. Boucadair et al., “A Framework for End-to-End Service Differentiation: Network Planes and Parallel Internets,” IEEE Commun. Mag., vol. 45, no. 9, Sept. 2007, pp. 134–43.

[13]. M. Kounavis et al., “The Genesis Kernel: A Programming System for Spawning Network Architectures,” IEEE JSAC, vol. 19, no. 3, 2001, pp. 511–26.

[14]. J. Touch, “Dynamic Internet Overlay Deployment and Management using the X-Bone,” Comp. Networks, vol. 36, no. 2–3, 2001, pp. 117–35.

[15]. W. Ng et al., “MIBlets: A Practical Approach to Virtual Network Management,” Proc. 6th IFIP/IEEE Int’l. Symp. Integrated Net. Mgmt., 1999, pp. 201–15.

[16]. J. E. van der Merwe et al., “The Tempest: A Practical Framework for Network Programmability,” IEEE Network, vol. 12, no. 3, 1998, pp. 20–28.

[17]. S. da Silva, Y. Yemini, and D. Florissi, “The NetScript Active Network System,” IEEE JSAC, vol. 19, no. 3, 2001, pp. 538–51.

[18]. A. Sundararaj and P. Dinda, “Towards Virtual Networks for Virtual Machine Grid Computing,” Proc. 3rd USENIX Virtual Machine Research Tech. Symp., 2004.

[19]. P. Ruth et al., “Virtual Distributed Environments in a Shared Infrastructure,” Computer, vol. 38, no. 5, 2005, pp. 63–69.

[20]. L. Peterson et al., “A Blueprint for Introducing Disruptive Technology into the Internet,” SIGCOMM Comp. Commun. Rev., vol. 33, no. 1, 2003, pp. 59–64.

[21]. A. Bavier et al., “In VINI veritas: Realistic and Controlled Network Experimentation,” Proc. ACM SIGCOMM, 2006, pp. 3–14.

[22]. http://www.industrialethernetu.com/courses/405_4.htm

[23]. http://www.excitingip.com/780/an-introduction-for-enterprise-vpn-virtual-private-network/

[24]. https://www.usenix.org/legacy/event/nsdi07/tech/full_papers/fonseca/fonseca_html/index.html

[25]. http://www.comdesigninc.com/comdesign/Services/VoiceandDataIntegration/tabid/138/Default.aspx

[26]. http://dcvizcayno.wordpress.com/2012/04/13/cloud-computing-tips-for-financial-industry/

[27]. http://www.surf.nl/en/knowledge-and-innovation/innovationprojects/2012/software-defined-networking.html