

Empirically Characterizing the Buffer Behaviour of Real Devices

Luis Sequeira, Julián Fernández-Navajas, Jose Saldana, Luis Casadesus
Communications Technology Group (GTC)-Aragón Inst. of Engineering Research (I3A)
Dpt. IEC. Ada Byron Building. EINA Univ. Zaragoza
50018 Zaragoza, Spain
Email: {sequeira, navajas, jsaldana, luis.casadesus}@unizar.es

Abstract—Commercial routers include a buffer in order to enqueue packets waiting to be transmitted. The behaviour of the routers' buffer is of primary importance when studying network traffic, since it may modify some of its characteristics, as delay or jitter, and may also drop packets. As a consequence, the characterization of this buffer is interesting, especially when real-time flows are being transmitted: if the buffer characteristics are known, then different techniques can be used so as to adapt the traffic: multiplexing a number of small packets into a big one, fragmentation, etc. This work presents a preliminary study of how to determine the technical and functional characteristics of the buffer of a certain device (as e.g. size, input and output rate and others characteristics related to its behaviour), or even the one of a remote Internet network node. Two different methodologies are considered and tested on two real scenarios which have been implemented. The use of real bursts permits the estimation of the buffer size, and the input and output rates, when there is physical or remote access to the “System Under Test”. In case of having physical access, the maximum number of packets in the queue can be determined by counting. In contrast, if the node is remote, its buffer size has to be estimated. Accurate results have been obtained in wired and wireless networks.

Index Terms—Buffer size, queueing, unattended measurements.

I. INTRODUCTION

The large increase in the number of users and new multimedia services over the Internet generates a significant amount of network traffic. Moreover, the expectation of future growth for the use of multimedia applications such as, e.g. Voice over Internet Protocol (VoIP) and online games [1], indicates that this tendency will increase. Besides, the heterogeneous characteristics of the different Internet accesses, together with user demands, make it necessary to define the Quality of Service (QoS) that they offer [2], especially when accesses networks have to deal with real-time applications [3].

Packet size may vary between different Internet services and applications [4] while some of them, as e.g. VoIP, generate small packets, in the order of a few tens of bytes, others use large packets in order to reduce the overhead. In addition, traffic behaviour [5] has a significant impact on network resources: while some services inject traffic with a constant bit rate in order to provide certain QoS and a better user's experience, other applications generate bursty traffic, with a different number of frames into each burst.

The demand of high amounts of data by Internet users, in conjunction with the existing complex network architectures,

implies that some network points become critical bottlenecks. Nowadays, this mainly happens in access networks, because their capabilities are lower than the ones available in the backbone; in addition, bottlenecks may also appear at critical points of high-performance networks.

In these points, the discarding of packets in router queues is the main cause of packet loss. So the implementation of router buffers and their policies, are of primary importance in order to ensure the correct delivery of the traffic of different applications and services.

Mid and low-end routers can be found in access networks, and they normally do not implement advanced traffic management techniques. However, they always use buffers as a traffic regulator mechanism, so buffer size becomes an important design parameter. Buffer can be measured in different ways: maximum number of packets it can store, amount of bytes, or even queueing time limit. Most Internet routers use FIFO drop-tail buffers [6] [7] but there exist another techniques to manage drop-tail buffers, e.g. Random Early Detection (RED). This techniques, in conjunction with buffer size, mainly define the buffer behaviour, and therefore, how the traffic is affected by it [8] [9].

As a consequence, if the size of the buffer and its behaviour are known, some techniques can be used so as to improve link utilization: multiplexing a number of small packets into a big one, fragmentation, etc. But there is a problem: manufacturers do not normally include all the implementation details in the technical specifications of the devices, but just some of them, related to the technology used, etc. On the other hand, if a communication has to cross another network or the Internet, some knowledge about the device's characteristics or the buffer's behaviour will be interesting. For these situations, our group is currently working on the development of a tool able to discover some characteristics of the buffer and its behaviour. The idea is to permit measurements not only when we have physical access to “System Under Test” but also in the case of only having remote access.

The paper is organized as follows: section II discusses the related works. The test methodology is presented in section III. The next section covers the experimental results. The paper ends with the conclusions.

II. RELATED WORKS

A. Buffer: types and behaviour

The fact of having different traffic at the input and output links of routers produce bottlenecks in the network, so packet loss may occur. Buffers are used to reduce packet loss by absorbing transient bursts of traffic when routers cannot forward them at that moment. They are instrumental in keeping output links fully utilised [6] during congestion times.

For many years, researchers accepted [10] the so-called *rule of thumb* to obtain the amount of buffering needed at a router's output interface. This rule was proposed in 1994 [11] and it is given by $B = C \times RTT$, where B is the buffer size, RTT is the average round-trip time and C the capacity of the router's network interface. This *rule of thumb* is also called the Bandwidth Delay Product (BDP). It was experimentally obtained using at most 8 TCP flows on a 40 Mbps core link, so there is no recommendation for sizing buffers when there is a significant number of TCP flows with different RTT s.

In 2004, researchers from *Stanford University* [12] proposed a reduced buffer size by dividing BDP by the square root of the number of TCP flows $B = C \times RTT / \sqrt{N}$. This new approach assumes that the number of TCP flows is large enough so as to consider them as asynchronous and independent from each other. This model was called *small buffer* [13].

In [14] it was suggested the use of even smaller buffers, called *tiny buffers*, considering a buffer size of some tens of packets. However, the use of this model presents a trade-off: on one hand, reducing buffers to only a few dozen of *KBytes* can produce a 10%–20% drop probability. On the other hand, the added delay is reduced, and this is interesting for real-time applications.

However, some real-time IP flows are bursty as e.g. video streaming, so this leaves some uncertainty in buffer sizing. In [15] and [16], TCP and UDP flows shared a connection using a small buffer, and an anomalous region, in which loss probability of UDP packets grows with buffer size, was found.

It has also been observed in the literature that the buffer size is measured in different ways: e.g., in [17] the routers of two manufactures are compared, and one gives the information in packets, whereas the other one measures it in milliseconds, which is equivalent to bytes.

On the other hand, router buffer size does not normally appear in the technical specifications that manufacturers provide with their devices. However, this design characteristic is important when planning a network. The reason for this is that there is a relationship between router buffer size and link utilization, since an excessive amount of memory would generate a significant latency increase when the buffer is full. On the other hand, a very small amount of memory in the buffer will increase packet loss during congestion time. As a consequence, the knowledge of the buffer behaviour is an interesting parameter which can be considered when trying to improve link utilization.

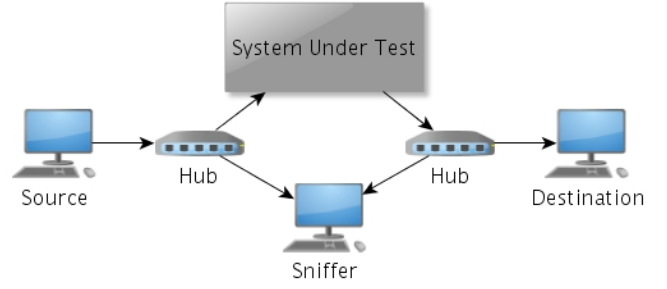


Fig. 1. Topology used for the test.

B. Influence of the buffer in different services

In [18] the authors present a Matlab simulation study of the influence of a multiplexing method on the parameters that define the subjective quality of online games: mainly delay, jitter and packet loss. The work considers two buffer implementations, each one with two buffer sizes, in order to study the relationship between router's buffer and multiplexing on subjective quality. The multiplexed traffic of a number of users share the same access link. The results show that small buffers present better characteristics in order to maintain delay and jitter in adequate levels, at the cost of increasing packet loss. In addition, buffers whose size is measured in packets also increase packets loss.

Another case of multiplexing study is presented in [19], where three different router's buffer policies (dedicated, big and time-limited buffer) are tested, also using two multiplexing schemes. Router's buffer policies may produce different results in terms of packet loss and R-factor.

III. TEST METHODOLOGY

A. Test procedure

The scheme of the tests is shown in figure 1. There is a "System Under Test" (SUT from now), which may be either a device or a network. Traffic is sent from a source, and arrives to the destination traversing the SUT. Two hubs and a *sniffer* are used in order to capture the traffic at the input and at the output of the SUT. The test is based on the sending of a burst of UDP packets from the source to the destination machine, so as to produce a buffer overflow in the SUT. This test is repeated using different bandwidths. Packets of different sizes are used so as to determine if the buffer is measured in number of packets or in bytes.

B. Methodology

We will use two methodologies to estimate buffer behaviour, size and output rate and other characteristics related to its behaviour. The methods are based on the premise that output rate can be obtained from *destination* capture. Output rate depends on the technology used in each case (LAN, WiFi).

- **Method 1:** Counting the number of packets in the queue in the moment that a packet arrives at the buffer.

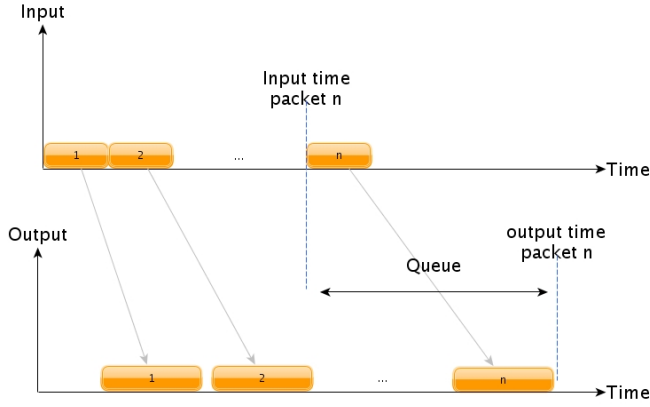


Fig. 2. Estimating packets in the queue.

- **Method 2:** If the delay of a packet in the buffer can be determined, then the variations of this delay can give us useful information for estimating buffer size.

The first option brings a more accurate estimation, but it requires physical access to the SUT. The second option can also be used when there is not direct access to the system.

1) *Finding buffer size with physical access:* The first option, which can be used to determinate buffer size in case of having physical access, is shown in figure 2. Every transmitted packet is identified by a sequence number included in the payload. Thus, buffer size can be estimated by the number of packets in queue between the arrival and leave time.

When physical access to the SUT is guaranteed, a *sniffer* can be included, capturing traffic at the ends of the device. When the capture ends, two files are saved at the input and output of the *sniffer*: we will denote them as *in-capture* and *out-capture*.

Both files are processed with a *shell script* to calculate packet delay, packet loss, interarrival packet time, input and output buffer rate and filling buffer rate. Buffer size is determined for each packet as follows: for all packets in *out-capture*, the *script* looks for the incoming time in *in-capture* and counts in *out-capture* the number of packets between incoming time and the time stamp registered in *out-capture*. The buffer size is estimated as the average of all these values.

2) *Finding buffer size with remote access:* A good estimation can also be obtained even if there is no physical access. By the use of ETG (E2E Traffic Generator), a tool which we will explain in the next section, unattended measurements in *destination* have been deployed. In spite of not having direct *sniffer* captures, it is possible to know the incoming traffic characteristics in the *destination*. Nevertheless, some modifications are required in the way calculations are done. In this case, we have used the *Method 2* to determinate buffer size.

Figure 3 shows the relationship between sent and received packet times. The method is based on sending a continuous packet flow at a rate bigger than the output capacity. This will flood the buffer and cause packet loss. T' is the time required

to completely fill the buffer at the same time it is being emptied at an output rate. T_r is the sum of two times: the delay for completely filling the buffer until a packet loss occurs; plus the time the last accepted packet needs to go through the buffer. As a consequence, this time will be noticed at the *out-capture* when the first packet is missing. So,

$$T_r = T_{fill} + T_{empty} \quad (1)$$

Let R_{in} and R_{out} be the input and output rates of the buffer, respectively. We define R_{fill} as the rate in which the buffer fills when R_{in} is bigger than R_{out} ($R_{fill} = R_{in} - R_{out}$). L_{buffer} is the size of the buffer in bytes. A packet spends $L_{buffer}/rate$ to cross the full buffer, so we can obtain T_r as,

$$T_r = \frac{L_{buffer}}{R_{fill}} + \frac{L_{buffer}}{R_{out}} \quad (2)$$

therefore,

$$L_{buffer} = \frac{T_r}{\frac{1}{R_{in}-R_{out}} + \frac{1}{R_{out}}} \quad (3)$$

The output rate can be easily determined, because the remote capture includes the arrival time for each packet and packet length is known. Input rate is not as trivial as output rate, because in this case we only have the *destination* capture. But if transmission time could be determined, for a certain amount of packets, then input rate can also be found.

The delays T_r and T_t are not the same because the buffer never empties completely, since new packets are always arriving at the input. However, t can be exactly measured and it will have the same value in both extremes. In this case, when *destination* receives n consecutive packets, the *source* has sent $n + m$ packets (where m is the number of dropped packets). m can be known since all the packets have a unique identifier. With this information, output and input rates can be estimated only from the data contained in the *destination* capture, using the following expression:

$$R_{in} = \frac{n_{tx} + m_{tx}}{t} \times packet_{size} \quad (4)$$

IV. EXPERIMENTAL RESULTS

In order to illustrate the proposed methodology, real tests have been deployed in a testbed, which scheme is shown in figure 1. The results are analysed based on procedures shown in figure 2 and 3. Real machines have been used (Linux kernel 2.6.38 – 7, Atheros AR9287 wireless network adapter, Intel® Core™ i3 CPU 2.4 GHz) in a laboratory environment, in order to identify the buffer behaviour of different devices.

A. ETG (E2E Traffic Generator)

This tool was developed by our group [20] with the aim of using it for link analysis in VoIP. In this work we will use it to generate the IP flows. It allows us to calculate objective quality parameters (delay, jitter and loss rate) and also subjective ones

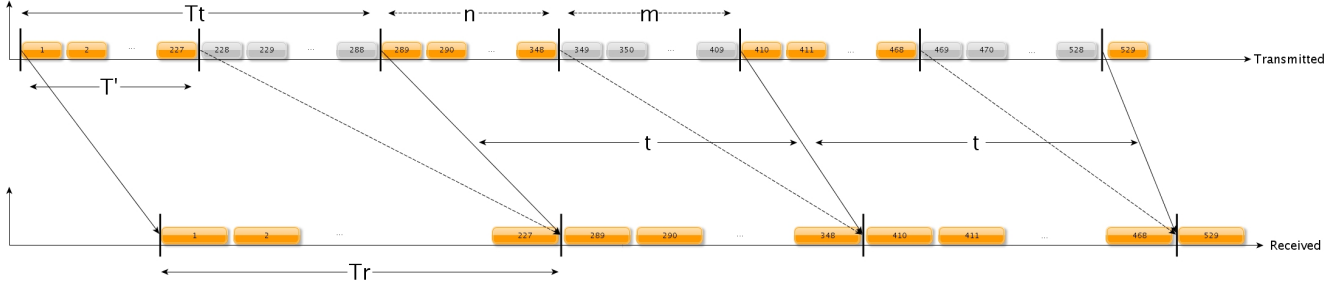


Fig. 3. Estimating packets in queue in remote access.

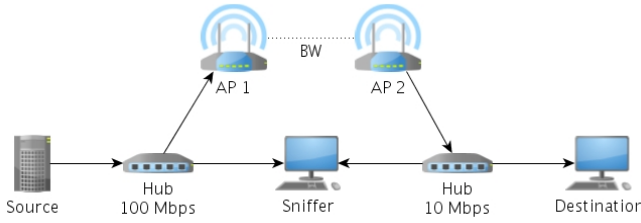


Fig. 4. First scenario: Estimating buffer size in a wireless network.

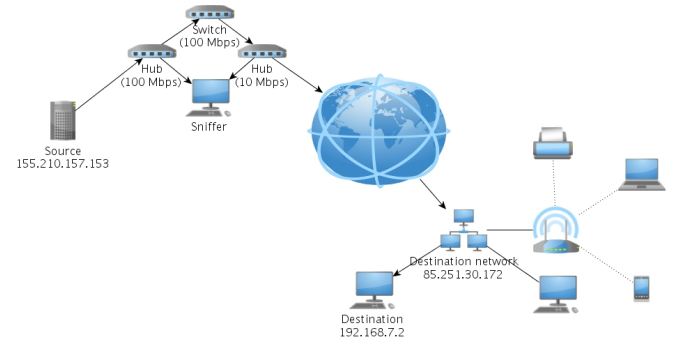


Fig. 5. Second scenario: Estimating buffer size in a wired network.

(R factor, MOS). It is capable of establishing E2E communications between hosts, by sending and receiving multiple UDP traffic bursts. It generates one-way and round trip traffic and captures packets at the ends of the communication.

One advantage of this application is that it provides repeatability for the tests. In addition, a mechanism allows task automation, allowing test iteration between start and end moments. It is also able to generate traffic emulating the characteristics of certain services, and it allows sending packets whose size and transmission intervals are read from a file.

B. Real scenarios

We have studied two different scenarios, in both cases a host sends traffic to a destination through a network or a device. A sniffer is included at a suitable location for making captures. ETG is used to generate UDP flows to flood the buffer, and also to automate *tcpdump* captures in the ends of the link. Flows are sent from *source* to *destination*. Different bandwidths are set in the hubs in order to create a bottleneck which has to be measured. Flow control options in the switch are disabled.

1) *First scenario: Wireless network:* In the first case, a link between two access points (Linksys WAP54G) has been analyzed, trying to estimate the buffer size of this access point. Figure 4 shows the used topology: both hosts are connected to the hubs using 100 Mbps and 10 Mbps links respectively. UDP flows were sent from *source* to *destination*, with the aim of obtaining the buffer size of AP1.

2) *Second scenario: Wired network:* For the second scenario, the topology shown in figure 5 has been used. In this case, a typical home network is accessed by other host from a

different network across the Internet. UDP flows are sent from the *source* to the *destination*. This scenario permits partial access to the SUT, so the measurements can estimate switch's buffer size by packets going from *source* to *destination*. Using the remote capture buffer size, the concatenation of different buffers across the internet can be estimated.

C. Measurements with physical and remote access

A particular buffer behaviour has been observed in the wired and wireless devices tested in both scenarios: when the buffer is completely full no more packets are accepted, it will be called *upper threshold*. In this moment, the buffer does not accept new packets until a certain amount of memory is available, it will be called *lower threshold*, and in this moment the filling process begins again. This can be seen in figure 6.

Although this behaviour has some similarities with Random Early Detection (RED), it is not the same: when RED is used, if the buffer is almost empty, all incoming packets are accepted but when the number of enqueued packets grows, the probability for dropping an incoming packet grows too. Finally, if the buffer is full, all incoming packets are dropped. On the other hand, the observed buffer behaviour does not use any dropping probability: if the buffer is full, it does not accept new packets until a certain amount of memory is available.

It is well known [17] that some manufacturers measure the buffer in number of packets whereas others define it in number of bytes, or even in terms of the maximum allowed queueing delay. In order to take this into account, UDP flows using

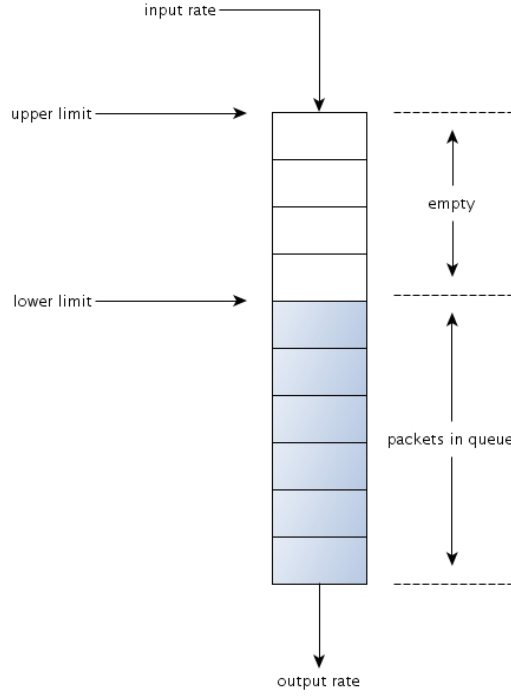


Fig. 6. Buffer characteristics.

TABLE I
INTER-PACKET TIME IN μs

Packetsize	10 Mbps	20 Mbps	30 Mbps	40 Mbps
1500	1200	600	400	300
800	640	320	213.3	160
200	160	80	53.3	40

three different packet lengths are sent to flood the buffer. If the estimated size is the same for the three traffics, then we know that the buffer is measured in number of packets. Otherwise, it is measured in bytes. Table I presents a summary of the traffic used for these tests.

For both scenarios, buffer size is the same when changes in the packet length are made, so in these cases, buffer size is defined by number of packets.

1) *First scenario*: In this wireless scenario we have compared the two methods when there is physical access to the SUT (table II). The presented results are the ones obtained using packets of 1500 bytes, since they are the most accurate. It can be seen that *method 1* is the most accurate estimation so it has been used to compare with *method 2*. Variations of the output rate in wireless network generate error growth. The reason for this is illustrated in table III: while filling rate is relatively constant, the emptying rate shows variations for the highest bandwidths. The reason for this is that the WiFi access point switches from higher to lower speeds depending on the state of the radio channel.

TABLE II
WIFI ACCESS POINT BUFFER SIZE FOR DIFFERENT BANDWIDTH.

<i>Access point bandwidth</i>		<i>Method 1</i>		<i>Method 2</i>		<i>Method 2 error (%)</i>	
<i>Mbps</i>		<i>LT</i>	<i>UT</i>	<i>LT</i>	<i>UT</i>	<i>LT</i>	<i>UT</i>
<i>Wi-Fi</i>	1	30	55	30	55	0	0
	2	30	55	30	55	0	0
	5.5	30	55	30	55	0	0
	11	30	55	32	53	6.67	3.63
	24	30	55	33	52	10	5.45
	54	30	55	36	59	20	7.27

LT: Stands for lower threshold.

UT: Stands for upper threshold.

TABLE III
OBSERVED VARIATIONS IN BUFFER OUTPUT RATE FOR DIFFERENT BANDWIDTH.

Access point rate	54 Mbps	24 Mbps	11 Mbps	5.5 Mbps	2 Mbps	1 Mbps
Minimum(Mbps)	10.88	13.7	5.75	2.29	1.24	0.65
Maximum(Mbps)	28.36	16.84	6	3.13	1.41	0.65

2) *Second scenario*: In this case we want to test the accuracy of our method when there is no possibility of physical access to the SUT. In order to be able to present a comparison we obtained the buffer size using *method 1* with physical access. Next, the estimations obtained using *method 2* are compared with previous results, and the relative error is presented. These tests are deployed using physical and remote access (see table IV). Again, the best results are obtained using 1500 bytes packets, so we present them. Three different amounts of bandwidth have been used in order to flood the buffer.

If we look at the *physical access* results (first three rows of the table), we see that the accuracy of the buffer size estimation using the *method 2* is high. In addition, the error diminishes as the input rate grows.

Regarding *remote access* results, it should be noticed that the results are less accurate although they are still acceptable. As integer values for the buffer capacity are used, in some cases the results are exactly the same than the ones of *method 1*, so the obtained value of the error is null.

Again, the values presented in the tables are the ones obtained using packets of 1500 bytes. The results using smaller packets are less accurate so we have not presented them.

V. CONCLUSION

This article has presented two methods to analyse the technical and functional characteristics of commercial buffers of different devices, or even networks. This characterization is important, taking into account that the buffer may modify traffic characteristics, and may also drop packets.

The methodology can be used if there is physical access to the "System Under Test", but it is also useful, with certain

TABLE IV
ETHERNET BUFFER SIZE FOR DIFFERENT BANDWIDTH.

Bandwidth	Method 1		Method 2		Method 2 error (%)	
Mbps	LT	UT	LT	UT	LT	UT
Ethernet (physical)	20	85	115	83	113	2.35
	30	85	115	84	114	1.17
	40	85	115	84	115	1.17
Ethernet (remote)	20	85*	115*	80	111	5.89
	30	85*	115*	83	113	2.35
	40	85*	115*	84	115	1.17

LT : Stands for lower threshold.

UT : Stands for upper threshold.

*These values were calculate as physical access to compare method 2.

limitations, for measuring a remote system. Tests using commercial devices have been deployed in two different scenarios, including wired and wireless networks. A particular buffer behaviour has been observed for a device: once the buffer is full, it does not accept new packets until a certain space is again available.

The results show that accurate results of the buffer size can be obtained when there is physical access to the "System Under Test". In case of having no direct access to the system, an acceptable estimation can also be obtained if the input rate is more than three times the output rate. In this case, big packets have to be used for the tests. As future work the method has to be improved in order to minimize the error, especially when measuring wireless devices.

ACKNOWLEDGMENT

This work has been partially financed by CPUFLIPI Project (MICINN TIN2010-17298), MBACToIP Project, of Aragon I+D Agency, Ibercaja Obra Social, NDCIPI-QQoE Project of Cátedra Telefónica, Univ. of Zaragoza and Fundación Carolina.

REFERENCES

- [1] G. Huang, M. Ye, and L. Cheng, "Modeling system performance in mmorpg," in *Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE*, nov.-3 dec. 2004, pp. 512 – 518.
- [2] A. Lombardo, G. Morabito, and G. Schembra, "Statistical traffic modeling and guaranteed service disciplines: a performance evaluation paradigm," *Computer Networks*, vol. 36, no. 5-6, pp. 579 – 595, 2001. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128601001700>
- [3] M. Ries, P. Svoboda, and M. Rupp, "Empirical study of subjective quality for massive multiplayer games," in *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*, june 2008, pp. 181 – 184.
- [4] P. Svoboda, W. Karner, and M. Rupp, "Traffic analysis and modeling for world of warcraft," in *Communications, 2007. ICC '07. IEEE International Conference on*, june 2007, pp. 1612 – 1617.
- [5] A. Golaup and H. Aghvami, "A multimedia traffic modeling framework for simulation-based performance evaluation studies," *Computer Networks*, vol. 50, no. 12, pp. 2071 – 2087, 2006, network Modelling and Simulation. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128605003294>

- [6] R. Stanojević and R. Shorten, "Trading link utilization for queueing delays: An adaptive approach," *Comput. Commun.*, vol. 33, no. 9, pp. 1108–1121, Jun. 2010. [Online]. Available: <http://dx.doi.org/10.1016/j.comcom.2010.02.014>
- [7] K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye, "Bandwidth estimation in broadband access networks," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, ser. IMC '04. New York, NY, USA: ACM, 2004, pp. 314–321. [Online]. Available: <http://doi.acm.org/10.1145/1028788.1028832>
- [8] A. Lakshmikantha, R. Srikant, and C. Beck, "Impact of file arrivals and departures on buffer sizing in core routers," *IEEE INFOCOM The 27th Conference on Computer Communications*, vol. 19, pp. 86–90, 2008. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4509621>
- [9] A. Dhamdhere and C. Dovrolis, "Open issues in router buffer sizing," *SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 87–92, January 2006. [Online]. Available: <http://doi.acm.org/10.1145/1111322.1111342>
- [10] A. Vishwanath, V. Sivaraman, and G. N. Rouskas, "Considerations for sizing buffers in optical packet switched networks," *IEEE INFOCOM 2009 The 28th Conference on Computer Communications*, pp. 1323–1331, 2009. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5062047>
- [11] C. Villamizar and C. Song, "High performance tcp in ansnet," *SIGCOMM Comput. Commun. Rev.*, vol. 24, pp. 45–60, October 1994. [Online]. Available: <http://doi.acm.org/10.1145/205511.205520>
- [12] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 281–292, August 2004. [Online]. Available: <http://doi.acm.org/10.1145/1030194.1015499>
- [13] N. Beheshti, Y. Ganjali, M. Ghobadi, N. McKeown, and G. Salmon, "Experimental study of router buffer sizing," pp. 197–210, 2008. [Online]. Available: <http://doi.acm.org/10.1145/1452520.1452545>
- [14] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden, "Part iii: routers with very small buffers," *SIGCOMM Comput. Commun. Rev.*, vol. 35, pp. 83–90, July 2005. [Online]. Available: <http://doi.acm.org/10.1145/1070873.1070886>
- [15] A. Vishwanath and V. Sivaraman, "Routers With Very Small Buffers: Anomalous Loss Performance for Mixed Real-Time and TCP Traffic," pp. 80–89, Jun. 2008. [Online]. Available: <http://dx.doi.org/10.1109/IWQOS.2008.16>
- [16] A. Vishwanath, V. Sivaraman, and M. Thottan, "Perspectives on router buffer sizing: recent results and open problems," *SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 34–39, March 2009. [Online]. Available: <http://doi.acm.org/10.1145/1517480.1517487>
- [17] J. Sommers, P. Barford, A. Greenberg, and W. Willinger, "An sla perspective on the router buffer sizing problem," *SIGMETRICS Perform. Eval. Rev.*, vol. 35, pp. 40–51, March 2008. [Online]. Available: <http://doi.acm.org/10.1145/1364644.1364645>
- [18] J. Saldana, J. Fernández-Navajas, J. Ruiz-Mas, E. Viruete Navarro, and L. Casadesus, "Influence of online games traffic multiplexing and router buffer on subjective quality," in *Proc. CCNC 2012- 4th IEEE International Workshop on Digital Entertainment, Networked Virtual Environments, and Creative Technology (DENVECT)*, pp. 482–486, Las Vegas, January 2012.
- [19] J. Saldana, J. Murillo, J. Fernández-Navajas, J. Ruiz-Mas, E. Viruete, and J. I. Aznar, "Evaluation of multiplexing and buffer policies influence on voip conversation quality," in *Proc. CCNC 2011- 3rd IEEE International Workshop on Digital Entertainment, Networked Virtual Environments, and Creative Technology*, pp. 1147–1151, Las Vegas, January 2011.
- [20] L. A. Casadesus Pazos, J. Fernández Navajas, J. Ruiz Mas, J. M. Saldana Medina, J. I. Aznar Baranda, and E. Viruete Navarro, "Herramienta para automatización de medidas de tiempo real extremo a extremo," *Actas del XXVI Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2011)*, Leganés (España). ISBN 9788493393458. Sept. 2011.