Evaluation of Transport Services for VoIP

Panagiotis Papadimitriou and Vassilis Tsaoussidis

Demokritos University, Electrical & Computer Engineering Department
Xanthi, 67100 GREECE
E-mail: {ppapadim, vtsaousi}@ee.duth.gr

Abstract **— We study transport protocol performance from an application-specific perspective. Initially, we focus on TCP and UDP supportive role in the context of VoIP performance. Applying our metric for real-time performance, we discover that UDP has limited efficiency. Beyond UDP/TCP, we evaluate a solution-framework based on TCP protocols which incorporate variable congestion mechanisms. We also investigate VoIP traffic friendliness, as well as potential tradeoffs between protocol performance and fairness. Furthermore, we evaluate VoIP quality and protocol sensitivity versus a range of QoS parameter adjustments.**

Keywords **– Voice over IP, QoS, transport protocols, performance evaluation, congestion control**

I. Introduction

Internet is evolving into a universal communications network, hosting several types of traffic including traditional data, voice and video. Voice over IP (VoIP) has emerged as an alternative to the traditional *Public Switched Telephone Network* (*PSTN*) and is steadily gaining popularity. Unlike traditional telephony, VoIP takes advantage of packet network properties, thus achieving more effective bandwidth utilization. Furthermore, VoIP exploits advanced voice compression techniques, which reduce the size of the transmitted stream. A fundamental advantage of VoIP is that it enables the creation of applications, which integrate voice with data. Consequently, the services provided by VoIP are not restricted to voice communication, but often include other media (e.g. video) and data applications, such as white boarding and file sharing.

VoIP, as a high quality real-time voice communication, has stringent end-to-end delay and loss rate requirements [9]. Delays above 150 ms are perceived by most users, while delays exceeding 300 ms usually render the conversation annoying. Apart from high latencies, delay variations often impact voice quality. More precisely, a significant proportion of packets delivered very late either cause conversational gaps or generate a confusing conversation. Therefore, variations of throughput and delay, along with reliability parameters, such as packet loss and packet errors usually degrade the performance of such applications. Even in the situation of excellent VoIP end-systems, the varying delays and loss characteristics of the Internet may still cause perceptible degradation of voice quality.

VoIP usually runs over *User Datagram Protocol* (*UDP*), due to the impression that UDP is more suitable for real-time applications. UDP is a fast, lightweight and free-transmitting protocol, which appears to meet the demanding QoS requirements of time-sensitive applications. However, UDP can not guarantee reliability and certainly is not able to deal with network delays either, since it lacks all basic mechanisms for error recovery and flow/congestion control. Despite the impression that congestion control is not mandatory for real-time applications, such as VoIP, relevant research work has revealed that congestion control does not necessarily degrade performance. For example, in [14] we have shown that UDP occasionally achieves worse performance than TCP.

Although *Transmission Control Protocol* (*TCP*) dominates the Internet, it is apparently a less common option for real-time applications. The most fundamental design principle of TCP is reliability. The *AIMD* algorithm [2], incorporated in standard TCP versions, achieves stability and converges to fairness when the demand of competing flows exceeds the channel bandwidth. TCP is further enhanced with a series of mechanisms for congestion control, including *Congestion Avoidance*, *Slow Start*, *Fast Retransmit* and *Fast Recovery* [8, 17]. However, standard TCP exhibits limited efficiency in heterogeneous wired/wireless environments, since it is not able to detect the nature of the errors that result in packet losses and consequently determine the appropriate error-recovery strategy. More precisely, authors in [19] outline three major shortfalls of TCP: (i) ineffective bandwidth utilization, (ii) unnecessary congestion-oriented responses to wireless link errors (e.g. fading channels) and operations (e.g. handoffs), (iii) wasteful window adjustments over asymmetric, low-bandwidth reverse paths. Since the Internet provides services for various types of applications, flow contention inevitably appears in varying traffic, spatial and temporal patterns. VoIP flows competing with traditional data traffic is a common scenario. In the event of limited bandwidth availability, coexisting traffic often impacts the timely delivery of VoIP packets degrading application performance. Inversely, VoIP flows may also hurt congestion-sensitive traffic when they compete for scarce bandwidth.

Our objective is to explore the performance of current end-to-end solutions from the perspective of VoIP QoS. More precisely, we justify UDP versus TCP investigating whether UDP exhibits superior performance, as implied. Beyond UDP and TCP supportive role, we study the particular methodology for congestion control by evaluating a series of TCP protocols that deal with congestion from different perspectives (i.e. congestion control, congestion avoidance). Based on comprehensive results, we explore the potential of these protocols focusing on the associated impact on VoIP quality. Furthermore, we investigate VoIP traffic friendliness, as well as potential tradeoffs between protocol performance and fairness. Since voice quality is subjective and expressed primarily with respect to the individual user, we demonstrate additional results of VoIP performance versus diverse deadlines for packet inter-arrival times.

We organize the rest of the paper as follows. In the sequel, we provide an overview of research studies and proposals dealing with various aspects of VoIP QoS. Furthermore, we summarize the most remarkable end-to-end solutions towards the improvement of real-time performance. In Section III we present our evaluation methodology, and in Section IV we analyze the results of the experiments we performed. Finally, in the last section we highlight our conclusions.

II. RELATED WORK

VoIP is rapidly evolving and has eventually attracted the required attention of the research community. Numerous research studies have been presented, which span several aspects of VoIP. Related work includes [10], where VoIP services are evaluated focusing on the ability of the Internet to effectively support interactive voice communication. Authors in [12] investigate performance issues associated with mixing voice and self-similar data traffic within the Internet. In [13] the voice quality of a VoIP system is evaluated, when voice-data length and network conditions change. Authors in [6] investigate the behavior of UDP and VoIP over 802.11 networks from the perspective of number of connections that a single access point can support. Furthermore, [18] includes an overview of perceptual QoS assessment methodologies for VoIP systems.

Since VoIP exhibits the characteristics of real-time traffic, we also refer to research efforts, which deal with efficient QoS management of real-time applications focusing on protocol design. *Real-time Transport Protocol* (*RTP*) [16] is the standard for transmitting delay-sensitive traffic across packet-based networks. The protocol runs on top of UDP or TCP and provides end-to-end network transport functions suitable for real-time applications over multicast or unicast networks. RTP, in a complementary role, uses the sequence information to determine whether the packets are arriving in order, and it uses the time-stamping information to determine packet inter-arrival times. The data transport is augmented by *Real-time Control Transport Protocol* (*RTCP*), which allows the monitoring of data delivery. RTP and RTCP are designed to be independent of the underlying transport and network layers.

Authors in [4, 5, 22, 23] proposed a family of TCP compatible protocols, called *TCP-friendly*. TCP-friendly protocols achieve smooth window adjustments, while they manage to compete fairly with TCP flows. *TCP-Friendly Rate Control* (*TFRC*) [5] is a representative TCP-friendly protocol, which adjusts its transmission rate in response to the level of congestion, as estimated by the calculated loss rate. TFRC eventually achieves the smoothing of the transmission gaps and therefore, is suitable for applications requiring a smooth sending rate. However, this smoothness has a negative impact, as the protocol becomes less responsive to bandwidth availability [21]. *TCP Westwood* is a TCP-friendly protocol, which emerged as a sender-side-only modification of *TCP Reno* congestion control [11]. TCP Westwood exploits end-to-end bandwidth estimation to properly set the values of slow-start threshold and congestion window after a congestion episode. *TCP-Real* [20, 24] is high-throughput transport protocol that incorporates congestion avoidance mechanism in order to minimize transmission-rate gaps. Therefore, this protocol is suited for real-time applications, since it enables better performance and reasonable playback timers. TCP-Real employs a receiver-oriented and measurement-based congestion control mechanism that promotes TCP to a reliable solution over heterogeneous networks and asymmetric paths.

Congestion avoidance mechanisms usually prevent congestion episodes, which damage the timely delivery of packets and consequently degrade real-time application performance. Hence, protocols which incorporate such mechanisms are efficient for time-sensitive applications, such as VoIP. Congestion avoidance may be achieved through packet dropping (i.e. *RED*) or otherwise through bandwidth and delay estimation, which trigger transport-level adjustments prior to congestion. Alternatively, *ECN* is proposed [15], where packets are marked rather than dropped when congestion is about to happen. A well-designed, congestion avoidance mechanism is *TCP Vegas* [1, 7]. Every *Round Trip Time (RTT*) the sender calculates the throughput rate which subsequently is compared to an expected rate. Depending on the outcome of this comparison the transmission rate of the sender is adjusted accordingly. Based on [1] admissions, Vegas achieves higher transmission rates than TCP Reno and TCP Tahoe.

III. EVALUATION METHODOLOGY

A. Scenarios and Parameters

The evaluation plan was implemented on the *NS-2* network simulator. The experiments were conducted based on realistic VoIP scenarios. More precisely, we used a network topology (Fig. 1), which includes multiple bottlenecks, cross traffic and wireless links. The router R1 is the bottleneck for VoIP traffic (flows between source nodes and sink nodes), while the router R2 is another bottleneck for competing VoIP and cross traffic (flows between peripheral source and peripheral sink nodes). Cross traffic includes various FTP flows over TCP Reno: a common application running over a widely used transport protocol. The number of source and sink nodes is always equal. We also attached an equal number of peripheral source and sink nodes. In all experiments, we used droptail routers.

During a conversation, speakers alternate between activity and idle periods. Taking into consideration the ON and OFF periods, as well as the heavy-tailed characteristics and self-

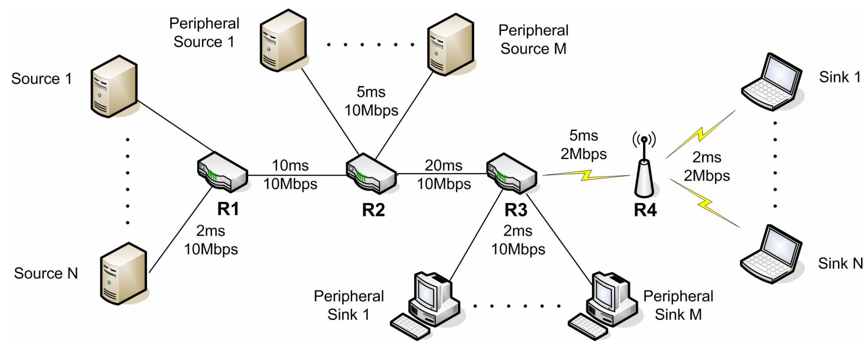

Figure 1. Simulation topology

similarity of VoIP traffic [3], we used the *Pareto* distribution for modeling the call holding times. We configured *Pareto* with a mean rate of 64 Kbps (inline with the widely-used ITU-T G.711 coding standard) and packet size was set to 160 bytes (20 ms of speech).

NS-2 error models were inserted into the access links to the main sink nodes. Error models were configured on both (forward and reverse) directions of the link traffic. We used the Bernoulli model in order to simulate the link-level errors with packet error rate (*PER*) adjusted at 0.01. The simulation time was fixed at 60 seconds, an appropriate time-period for all the protocols to demonstrate their potential. We performed our experiments running VoIP over UDP and various TCP protocols. However, due to space limitations we include results from UDP and TCP versions Vegas, Westwood and Real.

B. Measuring Performance

Since the simulated topology includes cross traffic, our performance metrics are applied separately to main VoIP traffic and cross FTP traffic. Exploiting this issue, we are able to perform a thorough study of the implications between interfering traffic. In the sequel, we refer to the performance metrics supported by our simulation model. System goodput is used to measure the overall system efficiency in bandwidth utilization. Fairness is measured by the *Fairness Index*, derived from the formula given in [2], and defined as $(\sum_{i=1}^{n}\text{Throughput}_i)^2/(n\sum_{i=1}^{n}\text{Throughput}_i^{\,2})$, where $Throughput_i$ is the throughput of the i^{th} flow and n is the total number of flows.

Evaluating the performance of real-time traffic based on traditional performance metrics (e.g. throughput) may produce misleading results, since such metrics do not account for variable delays which impact the efficiency of time-sensitive applications. In voice communications, several quality evaluation methods have been proposed [18]. The *Mean Opinion Score* (*MOS*) provides a numerical measure of human speech quality at the receiving end. MOS virtually indicates the speech quality perceived by the listener on a scale from 1 to 5. Most popular objective measurements include *Perceptual Evaluation of Speech Quality* (*PESQ*) and *E-model*.

In order to quantify the impact of individual impairments on voice quality, we exploited a revised version of the *Real-Time Performance* metric, which we initially proposed in [14]. The metric achieves the efficient performance evaluation of real-time traffic and can be easily configured according to individual application requirements, such as VoIP. More precisely, *Real-Time Performance Index* is defined as the ratio of the number of *timely received packets* over the total number of packets sent by the application:

$$\text{Real}-\text{Time Performance Index} = \frac{\#\text{timely received packets}}{\#\text{sent packets}} \leq 1$$

Real-time performance monitors packet inter-arrival times and distinguishes packets arriving on time from excessively delayed packets (according to a configurable inter-arrival threshold). Practically, delayed packets are either discarded and considered lost, or at the worst they obstruct the proper reconstruction of oncoming packets. The proportion of the number of delayed packets is denoted by *Delayed Packets Rate*. Since VoIP is sensitive to packet losses, we additionally define *Packet Loss Rate* as the ratio of the number of lost packets over the number of packets sent by the application. We applied the new metric by extending the functionality of the receiver based on the following algorithm.

Algorithm 1. Timely Received Packets

```
# For each packet received with sequence number i, determine
# whether it is a timely received packet or a delayed packet

if threshold > 0 then
   set packetTime[i] = currentTime
   increase packetsReceived
   if i = 0 then
      increase timelyPackets
   else
      if packetTime[i] - PacketTime[i - 1] > threshold then
         increase delayedPackets
      end if
   end if
end if
set timelyPackets = packetsReceived - delayedPackets
```

Several notations used in the pseudocode algorithm are as follows:

1. *threshold* : packet inter-arrival time threshold
2. *timelyPackets* : number of packets with inter-arrival times within the threshold
3. *delayedPackets* : number of packets with inter-arrival times exceeding the threshold
4. *packetTime* : packet arrival time
5. *packetsReceived* : number of packets reaching the receiver

In accordance with VoIP delay guidelines, we adjusted the inter-arrival threshold at 300ms. Consequently, arriving packets exceeding this deadline are marked as *delayed packets*, since they do not effectively participate in the reconstruction of voice data. Since our experiments were performed on several VoIP flows, we present the average of the real-time performance of each VoIP flow.

IV. RESULTS AND DISCUSSION

We hereby demonstrate and comment on selected results from the experiments we performed based on three distinct scenarios. The basic parameters of each simulation scenario are as described in the previous section.

A. TCP vs. UDP

In the first scenario, we performed a series of experiments in order to evaluate the performance of VoIP over TCP and UDP. We simulated a wide range of VoIP flows (10-80) adjusting the contention accordingly. Apart from the VoIP connections, we included FTP cross traffic (10 flows). Along these lines, we measured: *goodput*, *fairness index* and *real-time performance*. In addition, we selected statistics from delayed and lost packets, since are all influencing factors which impact VoIP performance. We hereby present some conclusive results from TCP Vegas, TCP Westwood, TCP-Real and UDP (Figs. 2-6).

Inline with our expectations, UDP achieves slightly higher goodput performance (Fig. 2) in comparison with the TCP protocols. More precisely, UDP transmits steadily at application rate, while TCP purposely backs off in periods of congestion. The reported goodput results (Fig. 2) outline further deficiency of TCP Westwood, especially with increased contention. The remarkable packet loss rate (Fig. 6) of TCP-Westwood indicates that the protocol is unable to effectively recover from excessive congestion, due to its smooth window adjustments. Furthermore, Vegas and especially Real, which is able to detect the nature of the error, respond appropriately to wireless errors. On the contrary, TCP Westwood, which does not incorporate an inherent mechanism for error classification and the corresponding recovery tactics, fails to deliver satisfactory performance in heterogeneous wired/wireless scenarios. Its performance is further diminished due to the unnecessary congestion-oriented responses to wireless link errors.

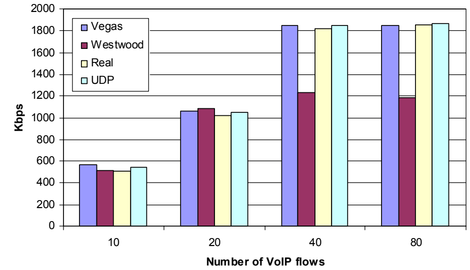

Figure 2. Goodput of VoIP flows

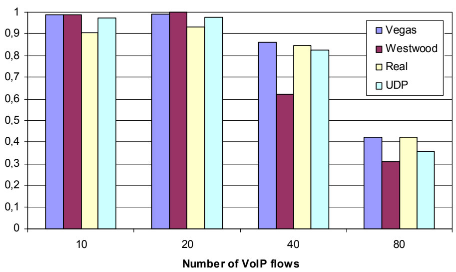

Figure 3. Average Real-Time Performance

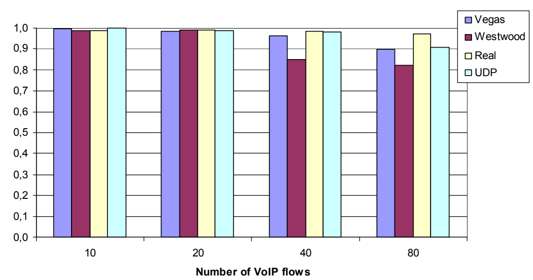

Figure 4. Fairness Index

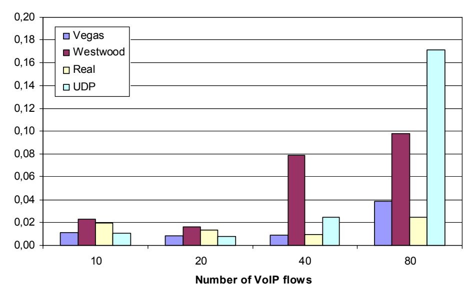

Figure 5. Delayed Packets Rate

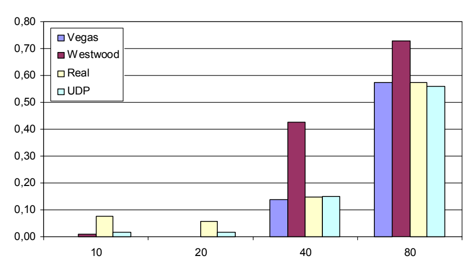

Figure 6. Packet Loss Rate

The high goodput rates achieved by UDP are not reflected in the real-time performance results. UDP does not yield satisfactory performance, especially as contention increases. A comparative view of the results in Figs. 5-6 justifies this unexpected result. Initially, UDP delivers the most packets to the receiver (Fig. 6): That is, less voice data are lost. However, Fig. 5 indicates that a significant proportion of the data (especially above 40 flows) reach the recipient exceeding the delay requirements of VoIP. Therefore, UDP results in long and variable delays, which usually cause conversational gaps and generally degrade the quality of online communication.

Similar to UDP, TCP Westwood exhibits inadequate real-time performance, due to low goodput rates and increased number of delayed packets. TCP-Real achieves superior performance, as a result of its receiver-oriented congestion control and its advanced error detection. Apart from goodput and real-time performance, TCP-Real excels in bandwidths sharing, regardless of link multiplexing (Fig. 4). On the contrary, TCP Vegas trades fairness for a remarkable performance; its congestion avoidance mechanism can not handle bandwidths sharing efficiently.

B. *The Impact of VoIP on Corporate FTP Traffic*

Departing from a comparative overview of protocol efficiency, we investigate the impact of VoIP on corporate FTP traffic. Our objective is to explore whether there is a tradeoff between protocol performance and efficiency of interfering traffic. Along these lines, we simulated various VoIP flows (10-80) competing with light (20 flows) and heavy (60 flows) cross FTP traffic over TCP Reno. We repeated the experiment for VoIP running over Vegas, Westwood, Real and UDP, and we measured the aggregated goodput of all FTP flows, correspondingly (Figs. 7, 8).

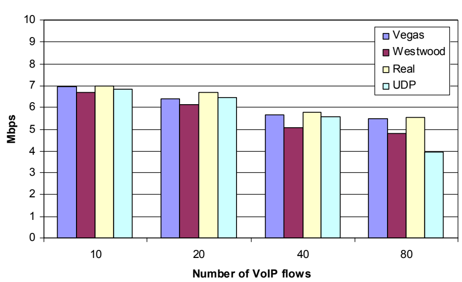

Figure 7. Goodput of light cross FTP traffic (20 flows)

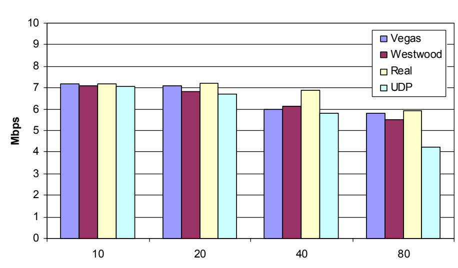

Figure 8. Goodput of heavy cross FTP traffic (60 flows)

UDP has a negative impact on the corporate FTP flows either with light or heavy cross traffic. This impact becomes more significant when contention increases and UDP allocates excessive network resources "stealing" bandwidth from the FTP flows. More precisely, UDP's free rate-transmitting policy enforces the interfering TCP flows to sharp window adjustments. Therefore, all applications sharing the same network resources with UDP face an unfavorable situation. On the contrary, TCP is designed to converge to fairness, and indeed, all TCP protocols included are relatively fair to the interfering FTP flows. Figs. 7, 8 depict that TCP-Real achieves a remarkable performance, since it mostly favors other applications sharing common network resources.

C. *VoIP Performance vs. A Range of Inter-arrival Thresholds*

In the last scenario, we performed a series of experiments using variable adjustments of the inter-arrival time threshold (100ms-400ms). Our primary goal is to investigate protocol sensitivity under these conditions. Furthermore, we show that VoIP performance is tightly coupled with the individual perception of quality. Along these lines, we simulated 40 and 80 VoIP flows competing with cross FTP traffic (20 flows). Diverse threshold adjustments affect delayed packets exclusively and, consequently real-time performance, so we present only the corresponding results (Figs. 9-12).

TCP Vegas and TCP-Real are most sensitive to the adjustments of the inter-arrival time threshold. The configuration of threshold at 200ms and above leads to respectable real-time performance gains for these protocols (Figs 9, 11). Generally, all TCP protocols exhibit a slight inefficiency for threshold adjustments below 200ms, which is reflected in the delayed packets results (Figs. 10, 12). In the situation of 40 VoIP flows (Figs. 9, 10), a user who is not annoyed by the conversation impairments caused by the adjustment at 200ms will be generally satisfied by the overall quality. On the contrary, another user with stringent QoS requirements (e.g. threshold at 100ms) will perceive only acceptable quality. The difference in perceived quality is significant, if VoIP runs over TCP Vegas. UDP generally exhibits less sensitivity to such adjustments in comparison with the TCP protocols. Hence, VoIP quality over UDP is less subjective to user perception of QoS parameters.

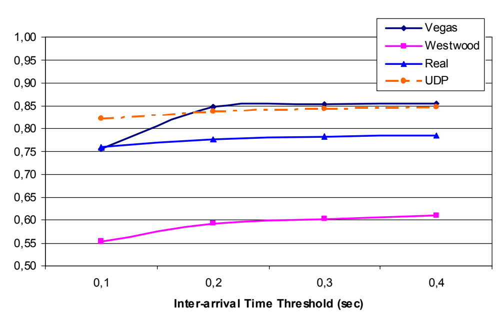

Figure 9. Average Real-Time Performance (40 VoIP flows)

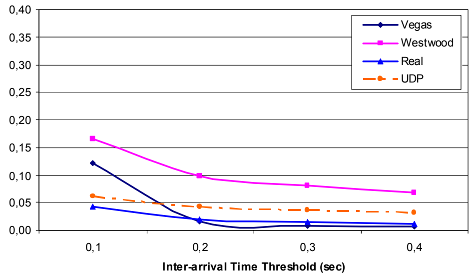

Figure 10. Delayed Packets Rate (40 VoIP flows)

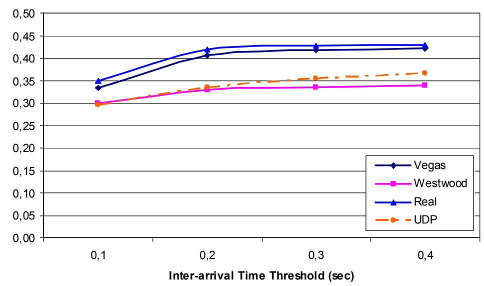

Figure 11. Average Real-Time Performance (80 VoIP flows)

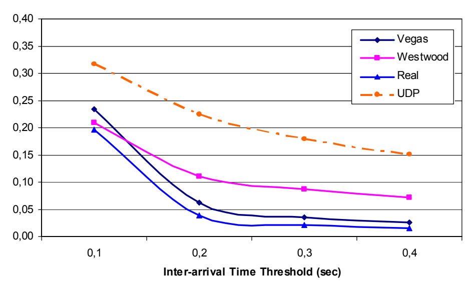

Figure 12. Delayed Packets Rate (80 VoIP flows)

V. CONCLUSIONS

We explored the supportive role of selected transport protocols in terms of VoIP performance. Our research efforts motivated from the (occasionally false) impression that UDP is the most prominent solution. However, our experimental results demonstrate the limited efficiency of UDP from the perspective of VoIP performance, as well as its destructive impact on interfering data traffic. Beyond that, we explored the potential of TCP protocols with different characteristics, such as congestion control and error detection mechanisms. A comparative overview of the associated results revealed that congestion control does not hurt time-sensitive applications. On the contrary, it occasionally results in performance gains. We also highlighted the importance of error classification for improved performance over wireless links. TCP-Real, which incorporates a sophisticated congestion avoidance mechanism along with error classification, is the most prominent solution among the protocols tested. The investigation of additional protocols performance (e.g. TFRC, SCTP), as well as an extension of current TCP-Real's functionality is under way.

REFERENCES

[1] L. Brakmo and L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet", *IEEE Journal on Selected Areas of Communications (JSAC)*, Vol. 13, No. 8, pp. 1465-1480, October 1995

[2] D. Chiu, R. Jain, "Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks", *Journal of Computer Networks*, Vol. 17, No. 1, pp. 1-14, 1989

[3] T. D. Dang, B. Sonkoly and S. Molnár, "Fractal Analysis and Modelling of VoIP Traffic", In Proc. of *NETWORKS 2004*, Austria, June 2004

[4] S. Floyd, M. Handley and J. Padhye, "A Comparison of Equation-based and AIMD Congestion Control", May 2000

[5] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications", In Proc. of *ACM SIGCOMM 2000*, Stockholm, Sweden, August 2000

[6] S. Garg and M. Kappes, "An Experimental Study of Throughput for UDP and VoIP Traffic in IEEE 802.11b Networks", In Proc. of *IEEE WCNC 2003*, New Orleans, Louisiana, USA, March 2003

[7] U. Hengartner, J. Bolliger, and T. Cross, "TCP Vegas Revisited", In Proc. of *IEEE INFOCOM 2000*, Tel-Aviv, Israel, March 2000

[8] V. Jacobson, "Congestion Avoidance and Control", In Proc. of *ACM SIGCOMM '88*, Stanford, USA, August 1988

[9] M. Karam and F. Tobagi, "Analysis of the Delay and Jitter of Voice Traffic over the Internet", In Proc. of *IEEE INFOCOM 2001*, Anchorage, Alaska, USA, April 2001

[10] A. Markopoulou, F. Tobagi and M. Karam, ''Assessment of VoIP Quality over Internet Backbones", In Proc. of *IEEE INFOCOM 2000*, Tel-Aviv, Israel, March 2000

[11] S. Mascolo, C. Casetti, M. Gerla, M. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", In Proc. of *ACM MobiCom '01*, Rome, Italy, July 2001

[12] E. Noel, K. W. Tang, "Performance Analysis of a VoIP Access Architecture", In Proc. of *IEEE ICPPW '04*, Canada, August 2004

[13] H. Oouchi, T. Takenaga, H. Sugawara and M. Masugi, "Study on Appropriate Voice Data Length of IP Packets for VoIP Network Adjustment", In Proc. of *IEEE GLOBECOM 2002*, Taipei, Taiwan, November 2002

[14] P. Papadimitriou, V. Tsaoussidis and S. Tsekeridou, "The Impact of Network and Protocol Heterogeneity on Real-Time Application QoS", In Proc. of *10th IEEE ISCC*, Cartagena, Spain, June 2005

[15] K. Ramakrishnan, S. Floyd, "A proposal to add explicit congestion notification (ECN) to IP", *RFC 2481*, January 1999

[16] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications", *RFC 1889*, January 1996

[17] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", *RFC 2001*, January 1997

[18] A. Takahashi and H. Yoshino, "Perceptual QoS Assessment Technologies for VoIP", *IEEE Communications Magazine*, Vol. 42, No. 7, pp. 28-34, July 2004

[19] V. Tsaoussidis and I. Matta, "Open issues on TCP for Mobile Computing", *Journal of Wireless Communications and Mobile Computing*, Vol. 2, No. 1, pp. 3-20, February 2002

[20] V. Tsaoussidis and C. Zhang, "TCP Real: Receiver-oriented Congestion Control", *Computer Networks*, Vol. 40, No. 4, pp. 477-497, Nov. 2002

[21] V. Tsaoussidis and C. Zhang, "The Dynamics of Responsiveness and Smoothness in Heterogeneous Networks", *IEEE Journal on Selected Areas in Communications (JSAC)*, Vol. 23, No. 6, pp. 1178-1189, June 2005

[22] Y.R. Yang, M.S. Kim and S.S. Lam, "Transient Behaviors of TCP-friendly Congestion Control Protocols", In Proc. of *IEEE INFOCOM 2001*, Anchorage, Alaska, USA, April 2001

[23] Y.R. Yang and S.S. Lam, "General AIMD Congestion Control", In Proc. of *8th Int/nal Conference on Network Protocols (ICNP)*, Osaka, Japan, November 2000

[24] C. Zhang and V. Tsaoussidis, "TCP Real: Improving Real-time Capabilities of TCP over Heterogeneous Networks", In Proc. of *11th IEEE/ACM NOSSDAV*, Port Jefferson, New York, USA, June 2001