Demonstrating Voice over an Autonomic Network

Lan Wang and Erol Gelenbe
Imperial College London
Department of Electrical and Electronic Engineering
Intelligent Systems and Networks Group
{lan.wang12,e.gelenbe}@imperial.ac.uk

***Abstract*—We demonstrate experimentally how an Autonomic Network based on the CPN protocol can provide the Quality of Service (QoS) required by voice communications. The implementation uses Reinforcement Learning to dynamically seek paths that meet the quality requirements of voice communications. Measurements of packet delay, jitter, and loss illustrate the performance obtained from the system.**

***Index Terms*—Autonomic Networks, Cognitive Packet Network, Voice Traffic**

I. Introduction

The quality a voice communication over a packet network depends on packet loss, delay, jitter, and packet "desequencing" [1], [2] resulting in Quality of Service (QoS) needs which are similar to those of real-time traffic [3], [4], [5], [6] which can be satisfied by a static and fast interconnection scheme, in contradiction with the dynamic self-organized behaviour of Autonomic Communication Systems (ACS) [7], [8]. This paper demonstrates that if an ACS is equipped with fast distributed decision algorithms that use precisely those objectives that are important to the quality of voice communications, then satisfactory overall performance is obtained for the Voice traffic stream, even when the network is shared by multiple streams which interfere with each other.

The system we use is the Cognitive Packet Network [9] which is a bio-inspired algorithm for dynamically finding network routes that best satisfy the QoS requirements of the application. CPN uses random neural networks with reinforcement learning [10], [8] at each network router, based on a class of analytically solvable probabilistic networks [11] which provide fast state predictions. The QoS metrics that are measured on-line by CPN and exploited to make routing choices for the Voice traffic are packet delay and jitter (or the second moment of delay). The network is demonstrated under conditions where it contains many traffic flows, some of which are Voice and use the same QoS goals, while other flows may just use one of delay, jitter or loss as their QoS objectives. CPN has smart packets (SPs), dumb packets (DPs) which are source routed and carry the main payload (here the Voice traffic), and acknowledgments (ACKs). SPs search for routes that may then be used by DPs along paths offering the desired QoS, while SPs collect measurements. DPs carry payload packets along source provided routes and also conduct measurements. ACKs bring back the information that has been discovered by the SPs and DPs, such as routes and their observed QoS. SPs discover the better routes, from a QoS perspective, using spiked random neural networks that are located at each intermediate node and make a "sensible" choice of the next hop [12] with a reinforcement learning algorithm at each node.

Our design is shown in Figure 1. At the sender which resides in a VOIP application installed at a CPN node, the original analog voice signals are sampled with a fixed frequency which is commonly 8000Hz and then each sample is encoded by using an audio data compression algorithm. There are several standards for audio encoding, such as G.711, G.729 (including G.729a), G.723.1, G.726, G.722, G.728, which are differ in the compression algorithms and the resulting bitstream bandwidth. The encoded bitstream is packetised into IP packets by adding RTP header, UDP header, and IP header. These packets can then be transmitted across the IP network employing the CPN protocol, where IP-CPN conversion is performed at the source node by encapsulating IP packets into CPN packets which are routed based on their QoS requirements. Due to the shared nature of the IP network, voice traffic may undergo transmission impairments including delay, jitter, packet desequencing and packet loss. CPN can alleviate these impairments by smartly selecting the path that provides the best possible QoS required by a user (or an application) and thus the packets in a voice traffic flow may traverse the CPN network successively along several different paths. Real voice traffic is generated by "Linphone" a VOIP phone installed at each node in the network testbed. At the receiver, packets are queued in the resequencing buffer which reorders packets and reduces jitter. Packets that arrive later than the maximum time allowed for voice signal recovery, or those that provoke buffer overflow, are discarded, contributing to end-to-end packet loss.

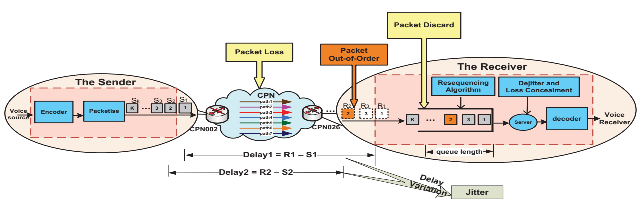

Fig. 1. VoCPN system structure distributed between sender and receiver; each end user will obviously have both sides of the ystsem for bilateral communications.

The demonstration is carried out on the test-bed network consisting of eight CPN software nodes each installed on a Linux machine with the topology shown in Figure 2, having multiple paths between a source-destination pair. CPN was implemented as a loadable kernel module under Linux 2.6.32 at each node. Adjacent nodes are connected with 100Mbps Ethernet links. Some of the experimental results we demonstrate are summarised in Figure 3, Figure 4, Figure 5 and Figure 6 whose captions explain the details that are shown.

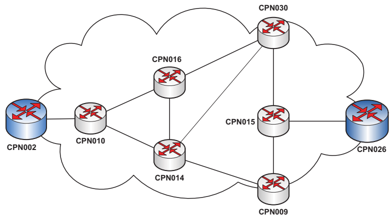

Fig. 2. Small CPN testbed network topology used in demonstration.

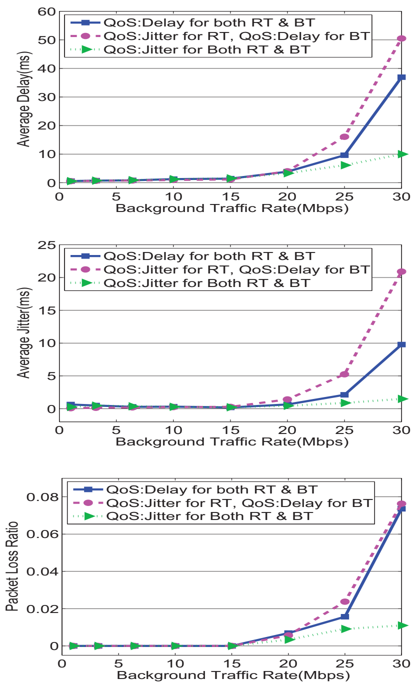

Fig. 3. The measured average delay (top), delay variation or jitter (middle), and packet loss ratio (bottom) are shown for a voice flow from CPN002 to CPN026 in the test-bed with different values of the overall background traffic rate.

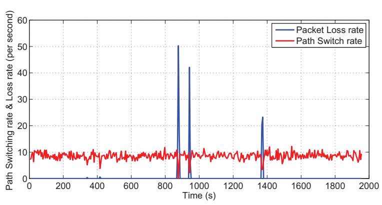

Fig. 4. Simultaneous measurement of Packet Loss and Path Switching in the test-bed shows its adaptive behaviour: a "good" path attracts more traffic, resulting in subsequent packet loss, followed by the traffic moving away to better paths thanks to CPN's autonomic adaptation.

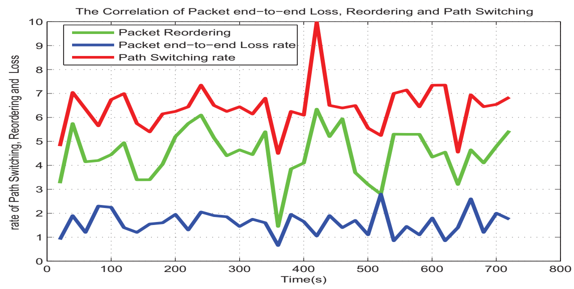

Fig. 5. The correlation of Packet end-to-en Loss, reordering and Path Switching under heavy traffic conditions: the data that is collected in the experiment, is processed using regression analysis.

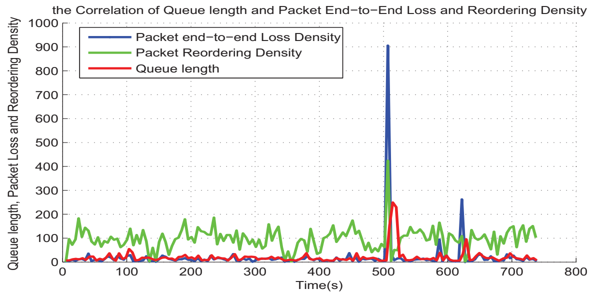

Fig. 6. The Correlation of the Queue Length in the Resequencing Buffer, with Packet Loss and Reordering

References

[1] F. Baccelli, E. Gelenbe, and B. Plateau, "An end-to-end approach to the resequencing problem," *J. ACM*, vol. 31, no. 3, pp. 474–485, Jun. 1984.

[2] C. Diot and F. Gagnon, "Impact of out-of-sequence processing on the performance of data transmission," *Computer Networks*, vol. 31, no. 5, pp. 475–492, 1999.

[3] J.Soldatos, E. Vayias, and G. Kormentzas, "On the building blocks of quality of service in heterogeneous ip networks," *Commun. Surveys Tuts.*, vol. 7, no. 1, pp. 69–88, Jan. 2005. [Online]. Available: http://dx.doi.org/10.1109/COMST.2005.1423335

[4] C. Aras, J. Kurose, D. Reeves, and H.Schulzrinne, "Real-time communication in packet-switched networks," *Proceedings of the IEEE*, vol. 82, no. 1, pp. 122–139, Jan 1994.

[5] A. P. Markopoulou, F. A. Tobagi, and M. J. Karam, "Assessing the quality of voice communications over internet backbones," *IEEE/ACM Transactions On Networking*, vol. 11, pp. 747–760, 2002.

[6] T. Kostas, M. Borella, I. Sidhu, G. Schuster, J. Grabiec, and J. Mahler, "Real-time voice over packet-switched networks," *Network, IEEE*, vol. 12, no. 1, pp. 18–27, Jan 1998.

[7] S. Dobson, S. Denazis, A. Fernández, D. Gaiti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli, "A survey of autonomic communications," *ACM Trans. Auton. Adapt. Syst.*, vol. 1, no. 2, pp. 223–259, 2006.

[8] E. Gelenbe, "Natural computation," *The Computer Journal*, vol. 55, no. 7, p. 848 !* 851, 2012.

[9] ——, "Steps toward self-aware networks," *Comm. ACM*, vol. 52, no. 7, pp. 66–75, July 2009.

[10] E. Gelenbe and S. Timotheou, "Random neural networks with synchronized interactions," *Neural Computation*, vol. 20, no. 9, pp. 2308–2324, 2008. [Online]. Available: http://dx.doi.org/10.1162/neco.2008.04-07-509

[11] E. Gelenbe, "The first decade of g-networks," *Europ. J. Operational Res.*, vol. 126, no. 2, pp. 231–232, 2000.

[12] ——, "Sensible decisions based on qos," *Computational Management Science*, vol. 1, no. 1, pp. 1–14, dec 2003.