Communication Using Phantoms: Covert Channels in the Internet

Sergio D. Servetto Martin Vetterli

Laboratoire de Communications Audiovisuelles
Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland.
URL: http://lcavwww.epfl.ch/ccn/

Abstract — **We consider the problem of determining the transport capacity of point-to-point and broadcast channels implemented on top of a network that enforces max-min fair bandwidth allocations in its routers. Our main finding is that the use of inefficient codes to represent data that is intended to be used solely for network control operations (such as routing, sequencing, etc), gives rise to the unintended creation of a covert channel. Sources can encode some information for their destinations into network control bits (on top of the standard method of encoding data into payload bits), by means of a mechanism which we refer to as the generation of "phantom" packets. Although phantoms provide only a marginal bandwidth increase, they could have potentially vast reaching implications in terms of security issues.**

I. An Interesting Example

In the context of designing systems for video multicast, it is of interest to consider the problem of designing codes for broadcast erasure channels [1]. A simple (yet very thought-provoking) coding example came up recently, illustrated in Fig. 1.

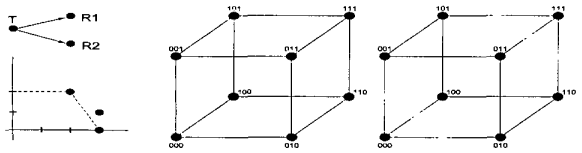

Figure 1: Consider a broadcast erasure channel defined as follows: from T to R_1 we have a noiseless channel, and from T to R_2 we have a channel that introduces exactly one erasure in any block of 3 contiguous bits.

How do we encode information to send over this channel? One possibility is to use as a code the circled nodes in the center cube in Fig. 1: this code has $d_{\min} = 2$, so it can correct up to one erasure, so both receivers can decode with no errors, thus effectively communicating two bits per three channel uses to both receivers. Another possibility is to use all possible nodes (not just the circled ones), thus communicating three bits to R_1 and nothing to R_0. These are the minimax points defined in [3], and by time sharing between these two codes all the rates on a straight line between (2,2) and (3,0) are achievable. Can this performance be improved upon? It turns out the answer is yes, provided we allow the possibility of introducing erasures at the input of the channel. Consider the code on the rightmost cube, obtained by taking a code with $d_{\min} = 3$ (000 and 111), and deliberately introducing an erasure at each possible position. Observe that now R_1 can still decode 3 bits of information: there are 8 codewords, and whenever an erasure is observed at the output it can be safely assumed that it was put at the input, since the channel is clean. But now R_2 can decode 1 bit of information as well: since we put at most one erasure into the channel, and the channel puts exactly one more, at the output there will be at most two erasures, which can be corrected by cloud centers with $d_{\min} = 3$. In Fig. 2 we compute the capacity of the modified binary erasure channel, and the capacity region of its broacast version, and we see that in both cases the capacities are higher than for standard erasure channels.

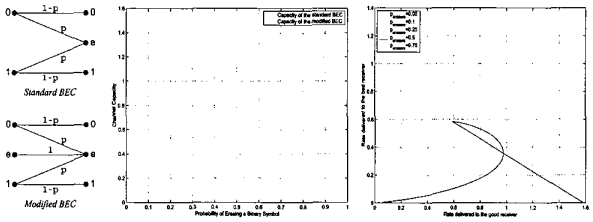

Figure 2: Capacity and capacity region for erasure channels.

II. Relevance

Prompted by the observation that under some circumstances allowing erasures at the input could actually increase channel capacity, we asked the question of what exactly does it mean to *send* an erasure. And it turns out that in practice, sending erasures translates to skipping sequence numbers. A code for sending data over this channel is a protocol that decides which sequence numbers to skip, by taking advantage of the fact that in practical implementations there is typically a lot of redundancy in their representation. Interesting coding (protocol) problems arise in this context [2].

Although the capacity increase due to phantoms is not significant in practice, and since it is not difficult to conceive situations in which they could be used to hide all sorts of low rate information (cryptographic keys, watermarks, credit card numbers, etc), a thorough study of the implications derived from our results in terms of network security appears necessary. Gallager pointed out in his classical paper [4]: *"... a conceptual understanding of protocol issues could undoubtedly give rise to much higher efficiencies in networks and, perhaps more importantly, to reduced complexity"*. Our results suggest that we should add security implications to that list as well, whose study is the focus of our ongoing research.

References

[1] S. D. Servetto and M. Vetterli, "Video Multicast over Fair Queueing Networks," in *Proc. IEEE Int. Conf. Image Proc. (ICIP)*, Vancouver, BC, 2000.

[2] S. D. Servetto and M. Vetterli, "Codes for the Fold-Sum Channel," in *Proc. 35th Annual Conf. Inform. Sciences Syst. (CISS)*, Baltimore, MD, 2001.

[3] T. M. Cover, "Broadcast Channels," *IEEE Trans. Inform. Theory*, vol. IT-18, no. 1, pp. 2–14, 1972.

[4] R. G. Gallager, "Basic Limits on Protocol Information in Data Communication Networks," *IEEE Trans. Inform. Theory*, vol. IT-22, no. 4, pp. 385–398, 1976.

0-7803-7123-2/01/$10.00 ©2001 IEEE