Design and Implementation of a Transparent Forward Error Correction Coding Daemon for Unreliable Transports*

Albert K.T. Hui and Samuel T. Chanson

Department of Computer Science
Hong Kong University of Science & Technology
Clear Water Bay, Kowloon, Hong Kong

Abstract. In this paper we propose a novel approach to adding a forward error correction (FEC) code to UDP and other IP-based unreliable protocols. We call this Sub-socket FEC (SSF). Implemented below the socket interface, SSF provides universal and transparent FEC protection to any IP socket-based communication channels on which all existing IP applications can continue to work without any modification. A prototype system called the FEC Daemon (FECD) has been implemented using the SSF approach. It employs an erasure code based on XOR encoding, which can be extended to support Vandermonde matrix coding. Since the system is light-weight and does not use retransmission, it is particularly suitable for multimedia applications with real-time constraints. The design, implementation and performance analysis of FECD are presented in this paper.

Keywords: Forward Error Correction, Erasure Code, Socket, TCP/IP, Multimedia Applications

1 Introduction

Data communications traditionally requires reliable data transmission. To achieve this goal, various error detection and correction techniques are employed at different layers to protect the integrity of the data as discussed below:

Bit Errors. At the data link layer, bit errors can be introduced by garbled signals or wrong timings. Upon detecting an Ethernet frame containing bit errors using its frame CRC checksum, receivers will drop it immediately.

Conventional physical media are so reliable that frame losses are almost never due to corrupted frames; hence TCP assumes that packet losses indicate congestions. This assumption no longer holds in wireless networks, causing standard TCP stacks to suffer from undue slow downs. To combat this, modern wireless

* This work was partially supported by a grant from the Hong Kong Research Grant Council (Number HKUST6080/97E).
adapters generally have FEC in the physical layer to keep the error level down to that of wired networks.

These physical layer FEC systems deal with bit errors, and therefore cannot correct packet stream errors.

**Packet Stream Errors.** Because of link layer error detection, only frames passing the CRC are visible at the network layer. Network functions also introduce additional errors in the form of lost, duplicated or misordered packets. Ethernet and IP checksums, designed to detect errors in the individual frames and packets, are incapable of dealing with these packet-stream errors.

TCP/IP stacks silently discard packets failing the IP checksum check in an “all or nothing” manner—packets are either error free, or not seen at all. The users of network layer see “holes” of dropped packets in the packet stream, instead of seeing corrupted bits in a bit stream.

TCP tackles this problem with retransmission whereby lost packets are repaired, duplicate ones discarded and all the packets correctly ordered before the data stream is pushed out of sockets. This convenience introduces retransmission delay and does not support multicast communications. Applications using UDP or other non-reliable protocols have to deal with packet-stream errors themselves.

**IP Multicast and Multimedia Applications.** In an IP multicast [2] network, members of a multicast group can have paths exhibiting a wide range of network characteristics. TCP flow control will fail to work in this environment.

Reliable transports like TCP are best used for bulk data transfer where perfect transmission is required. For multimedia applications, TCP can be a very poor choice because while TCP retransmission guarantees packets are delivered to their destination reliably, it offers no guarantee regarding their timely delivery. This is a major problem for multimedia applications with real-time constraints such as video conferences. Packets in these applications have a time frame within which the packets are useful. Late coming packets that have already expired are useless and will have to be dropped; they may as well have never been sent. On the other hand, perfect delivery of data is generally not needed. Multimedia applications can usually tolerate some degree of errors, since errors in a multimedia stream typically have no long-lasting effects.

### 1.1 Related Works

RFC 2198 defines RTP payload types for redundant audio which allows for multiple encodings of the same data in a session. Most implementations limit themselves to the most simple form of FEC—data repetition in $n + 1$ frames. RAT [3], FreePhone [4] and Percept IP/TV are examples of these. CCFAudio [5] has a simplified design where the primary and secondary codecs are the same. In multimedia applications retransmission makes little sense, hence AQR is seldom used in conjunction with FEC.