Multiple Time Scale Redundancy Control for QoS-sensitive Transport of Real-time Traffic

Tsunyi Tuan Kihong Park
Network Systems Lab
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

Abstract— **End-to-end QoS control over best-effort and differentiated service networks which exhibit variability in their exported service properties looms as an important challenge. In previous work, we have shown how packet-level adaptive FEC can be used in dynamic networks to facilitate invariant user-specified QoS in an end-to-end manner.**

This paper addresses two important problems—self-similar burstiness and performance degradation of reactive controls subject to long feedback loops—complementing the stability/optimality considerations studied earlier. First, for adaptive redundancy control to be effective, its susceptibility to correlated packet drops and queueing delays stemming from self-similar burstiness must be fortified. Second, to preserve FEC's viability over ARQ when transporting real-time traffic in WANs, proactivity must be injected to offset the performance degradation of reactive feedback controls when subject to long RTTs.

In this paper, we use the recently advanced multiple time scale congestion control framework—first investigated in the throughput maximization context—to endow adaptive redundancy control with both selective protection against self-similar burstiness as well as proactivity to feedback redundancy control. We analyze, implement, and benchmark our protocol—AFEC-MT—in the context of transporting periodic real-time traffic, in particular, MPEG video.

I. Introduction

A. Background

Forward error correction (FEC) is a well-studied reliable communication technique which has been successfully used, primarily at the *bit-level*, in a number of application domains from space communication to reliable data storage on compact disks [4], [14], [15]. In the context of supporting multimedia traffic with real-time constraints over high-speed wide-area networks, *packet-level* FEC has received interest due to ARQ's inherent limitation at handling timing constraints when subject to long end-to-end latencies [2], [5], [6], [12], [19], [23], [28].

Packet-level FEC introduces further complexities due to correlated packet drops (or erasures) and delays stemming from queueing which is especially severe under self-similar bursty traffic conditions. Cidon et al. [8], [9] have studied the impact of correlated packet drops on packet-level FEC performance and shown that their impact can be significant. Queueing analysis with Poisson input is provided in [1]. Empirical evidence [5], [23] indicates that performance degradation is further amplified under self-similar traffic conditions.

When applying packet-level FEC for real-time data transport in shared dynamic networks, it is imperative that appropriate redundancy or overcode—commensurate with network state and desired target QoS—is applied such that bandwidth is not unnecessarily wasted. In previous work [22], [23], [26], we proposed an adaptive packet-level FEC protocol called AFEC and analyzed its properties with respect to optimality and stability. The control problem is nontrivial due to the fact that increased redundancy, beyond a certain point, can "backfire" resulting in self-induced congestion which impedes the timely recovery of information at the receiver. We implemented and tested AFEC in high-speed LAN environments when transporting real-time MPEG video and showed that end-to-end QoS provisioning could be facilitated using adaptive redundancy control.

B. Problem Statement

The limitations of our previous work [22], [23] are two-fold: one, AFEC's adaptive redundancy control was geared toward protecting against generic forms of burstiness without special sensitivity to self-similar burstiness [16] thus leaving room for possible improvement, and two, AFEC—being a feedback control—suffered under the problem of long round trip latencies intrinsic to all reactive controls which reduced its effectiveness vis-à-vis ARQ in WAN environments.

We remark that these two problems—although studied in the specific context of adaptive redundancy control for real-time data transport—are also relevant to other forms of end-to-end QoS control over networks exporting variable services [7], [10], [20] where end-to-end control can be used to amplify and endow robustness to the QoS experienced by an application.

C. New Contributions

In this paper we show that the aforementioned problems—self-similar burstiness and feedback redundancy control with long RTTs—can be effectively addressed yielding significant performance improvements. Our solution is based on the framework of *Multiple Time Scale Congestion Control* (MTSC) [29], [30] which has been recently advanced in the

This research was supported by NSF grant ANI-9714707.

T.T.: E-mail: tsunyi@cs.purdue.edu.
K.P.: Contact author; e-mail: park@cs.purdue.edu. Additionally supported by NSF grants ANI-9875789 (CAREER), ESS-9806741, EIA-9972883 and grants from PRF and Sprint.

context of throughput maximization. The basic premise of MTSC hinges on the fact that despite the detrimental performance effect associated with self-similar burstiness the presence of nontrivial correlation structure across multiple time scales admits to exploitation for congestion control purposes. MTSC uses information extracted at large time scales to modulate the output behavior of feedback congestion controls acting at the time scale of the feedback loop. In a nutshell, when the network contention level at large time scales is predicted to be "low," the bandwidth consumption behavior of the underlying feedback congestion control is made more aggressive, and vice versa if the opposite is true.

In the context of adaptive redundancy control for QoS-sensitive transport of real-time traffic, the main technical challenge to adopting the MTSC framework lies in devising a large time scale module which is then coupled to AFEC such that the collective behavior facilitates both selective protection against self-similar burstiness and proactivity to counteract the reactive nature of AFEC. The specific form of coupling in the new protocol—AFEC-MT—is *additive* where the amount of redundancy h applied at an instant is composed of two parts, $h = h_S + h_L$, the short time scale component h_S acting at the time scale of the feedback loop governed by AFEC and the large time scale component h_L. The latter behaves like a "DC component" over the short time scale incurring level shifts at large time scales which reflect the overall contention level and corresponding redundancy needed to achieve a target end-to-end QoS. Whereas h_S is updated using *implicit* prediction afforded by feedback control, h_L is computed using *explicit* prediction.

We give a qualitative analysis of AFEC-MT with respect to its stability and optimality properties. We demonstrate the practical efficacy of the protocol by implementing and benchmarking AFEC-MT when transporting real-time MPEG video over controlled network environments where self-similar burstiness and varying end-to-end latency are systematically injected and their impact evaluated. Of particular interest is the facilitation of proactivity under increasing RTT which mitigates some of the performance cost of reactive controls when subject to broadband networks with a high delay-bandwidth product. We show that significant performance gains are possible by engaging in multiple time scale redundancy control.

The rest of the paper is organized as follows. In the next section, we give an overview of AFEC and its use in real-time communication. This is followed by Section III which describes the MTSC framework. Section IV discusses AFEC-MT and multiple time scale redundancy control, and Section V shows performance results from implementation based experiments for real-time MPEG video transport.

II. Overview of AFEC

A. Set-up

Assume k data packets—e.g., representing an MPEG video frame—are encoded as $n = k + h$ packets where $h \geq 0$ is the degree of redundancy. We will assume that the receipt of *any* k packets out of the n total packets suffices to recover the original k data packets. FEC encoding/decoding functions with this property include Reed-Solomon [18] and IDA [27]. Figure II.1 gives a depiction of packet-level FEC in an end-to-end network environment.

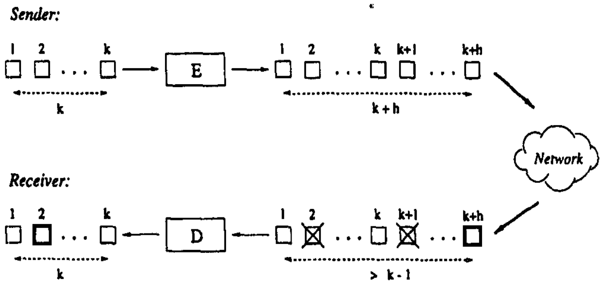

Fig. II.1. A block of k packets encoded at the sender using FEC as $k + h$ packets. If the number of dropped or untimely packets does not exceed h, the original k data packets are recovered at the receiver.

The application in question is *real-time constrained* in the following sense. The sender transmits $n(t_1), n(t_2), \ldots, n(t_i), \ldots$ $(t_i < t_j$ if $i < j)$ blocks of packets at times t_i where $n(t_i) = k(t_i) + h(t_i)$, $i = 1, 2, \ldots$. That is, $k(t_i)$ is the traffic requirement at t_i as dictated by the application and $h(t_i)$ the corresponding redundancy factor. We model QoS requirements using *hard* real-time constraints whereby we assume the existence of a monotonically increasing sequence $(t'_i)_{i \in \mathbf{Z}_+}$ of deadlines at the receiver such that all $k(t_i)$ data packets belonging to the i'th block must be recovered before time t'_i. For instance, given a frame rate of 30fps, successive frames must arrive within 33.3ms periodic intervals with some provision for decoding overhead at the receiving end station. QoS is measured at the receiver using a *recovery rate* process $\gamma(t'_i)$ which is defined to be the number of packets belonging to block i received before time t'_i. We will say there is a *hit* at time t'_i if $\gamma(t'_i) \geq k(t_i)$; i.e., decoding of the i'th frame was timely and successful.

B. Adaptive Redundancy Control

Increasing $h(t)$ blindly will adversely affect $\gamma(t + \tau)$, for some $\tau > 0$. That is, letting $\mathcal{G}$, $\gamma(t + \tau) = \mathcal{G}(h(t))$, denote the functional relationship between $h(t)$ and $\gamma(t + \tau)$, $\mathcal{G}(h) \geq 0$ is *unimodal* with peak at $h = h^*$. Thus if the goal is to achieve maximum recovery rate, then we arrive at an optimal control problem where the optimal operating point is defined as (h^*, γ^*), $\gamma^* = \mathcal{G}(h^*)$. If $\gamma^* < k$, then there is a structural problem and no amount of redundancy, small or large, can yield a positive hit rate. Figure II.2 depicts the unimodal redundancy-recovery rate relation $\gamma = \mathcal{G}(h)$.

Consider the case when $\gamma^* \geq k$. Assuming $\gamma(t)$ is fed back to the sender from the receiver, we can formulate the following

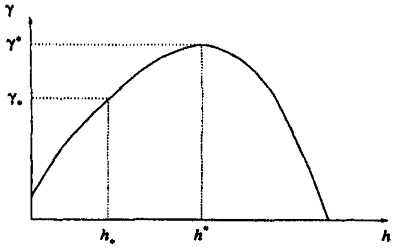

Fig. II.2. Unimodal redundancy-recovery rate function $\gamma = \mathcal{G}(h)$ with maximum recovery rate γ^* and target recovery rate γ_*.

control law

$$\frac{dh(t)}{dt} = \epsilon(\gamma_* - \gamma(t-\tau)) \qquad \text{(II.1)}$$

where γ_*, $k \leq \gamma_* \leq \gamma^*$, is the *target* recovery rate, $\epsilon > 0$ is an adjustment parameter, and $\tau \geq 0$ is a delay term introduced by feedback and network latency. The control algorithm as embodied by (II.1) just says that if the measured recovery rate $\gamma(t-\tau)$ is smaller than the target recovery rate γ_*, then the redundancy factor h should be increased, and vice versa.

Instability ensues when "too much" redundancy is applied at the sender which is tantamount to shooting oneself in the foot. Let

$$\begin{aligned} H_L &= \{(h,\gamma) : h = \mathcal{G}(h),\, h < h^*\}, \\ H_R &= \{(h,\gamma) : h = \mathcal{G}(h),\, h \geq h^*\}. \end{aligned}$$

It can be shown [22], [25] that target operating points belonging to $(h_*, \gamma_*) \in H_L$ are asymptotically stable whereas those belonging to $(h_*, \gamma_*) \in H_R$ are unstable. To achieve stability, we augment the control law given in (II.1) via a *directional check* given by the sign of $d\gamma/dh$. If $d\gamma/dh > 0$, then the system is in the stable region ($h < h^*$) and (II.1) is applied as usual. If, on the other hand, $d\gamma/dh \leq 0$, then the system finds itself in the unstable region ($h \geq h^*$) and a *backoff mechanism*—$dh/dt < 0$—is instituted until $d\gamma/dh > 0$ at which time we find ourselves again in the stable regime. Note that $d\gamma/dh > 0$ iff $(h,\gamma) \in H_L$.

The augmented AFEC algorithm containing *both* the symmetric and asymmetric control components is given by

$$\frac{dh(t)}{dt} = \begin{cases} \epsilon(\gamma_* - \gamma(t-\tau)), & \text{if } d\gamma(t-\tau)/dh > 0, \\ -ah, & \text{otherwise.} \end{cases}$$

Here $a > 0$ is a positive constant. Hence the augmented control follows (II.1), as it should, when $h < h^*$ (i.e., $(h,\gamma) \in H_L$), and it performs drastic, asymmetric backoff only when $h \geq h^*$ ($(h,\gamma) \in H_R$). The backoff mechanism, $dh/dt = -ah$, is *exponential* leading to a decay of h of the form e^{-at}.

III. Overview of MTSC

A. Self-Similar Burstiness

Let $\{X_t; t \in \mathbb{Z}_+\}$ be a time series which represents the trace of data traffic measured at some fixed time granularity. We define the aggregated series $X_i^{(m)}$ as

$$X_i^{(m)} = \frac{1}{m}(X_{im-m+1} + \cdots + X_{im}).$$

That is, X_t is partitioned into blocks of size m, their values are averaged, and i is used to index these blocks. Let $r(k)$ and $r^{(m)}(k)$ denote the autocorrelation functions of X_t and $X_i^{(m)}$, respectively. Assume X_t has finite mean and variance. X_t is *asymptotically second-order self-similar with parameter H* ($1/2 < H < 1$) if for all $k \geq 1$,

$$r^{(m)}(k) \sim \frac{1}{2}((k+1)^{2H} - 2k^{2H} + (k-1)^{2H}) \qquad \text{(III.1)}$$

as $m \to \infty$. H is called the *Hurst parameter* and its range $1/2 < H < 1$ plays a crucial role. The significance of (III.1) stems from the following two properties being satisfied:

(i) $r^{(m)}(k) \sim r(k)$,

(ii) $r(k) \sim c\,k^{-\beta}$,

as $k \to \infty$ where $0 < \beta < 1$ and $c > 0$ is a constant. Property (i) states that the correlation structure is preserved with respect to time aggregation, and it is in this second-order sense that X_t is "self-similar." Property (ii) says that $r(k)$ behaves hyperbolically which implies $\sum_{k=0}^{\infty} r(k) = \infty$. This is referred to as *long-range dependence* (LRD). The second property hinges on the assumption that $1/2 < H < 1$ as $H = 1 - \beta/2$.

The relevance of asymptotic second-order self-similarity for network traffic derives from the fact that it plays the role of a "canonical" model where the on/off model of Willinger *et al.*[1] [31], Likhanov *et al.*'s source model [17], and the M/G/∞ queueing model with heavy-tailed service times [11]—among others—all lead to second-order self-similarity. In general, self-similarity and long-range dependence are not equivalent. For example, fractional Brownian motion with $H = 1/2$ is self-similar but it is not long-range dependent. For second-order self-similarity, however, one implies the other and it is for this reason that we sometimes use the terms interchangably within the traffic modeling context. A more comprehensive discussion can be found in [24].

B. LRD and Predictability

Given X_t and $X_i^{(m)}$, we will be interested in estimating $\Pr\{X_{i+1}^{(m)} \mid X_i^{(m)}\}$ for some suitable aggregation level $m > 1$. If X_t is short-range dependent, we have

$$\Pr\{X_{i+1}^{(m)} \mid X_i^{(m)}\} \sim \Pr\{X_{i+1}^{(m)}\}$$

for large m whereas for long-range dependent traffic, correlation provided by conditioning is preserved. Thus given traffic observations $a, b > 0$ ($a \neq b$) of the "recent" past corresponding to time scale m,

$$\Pr\{X_{i+1}^{(m)} \mid X_i^{(m)} = b\} \neq \Pr\{X_{i+1}^{(m)} \mid X_i^{(m)} = a\}$$

[1] That is, via its relation to fractional Brownian motion and its increment process, fractional Gaussian noise.

and this information may be exploited to enhance congestion control actions undertaken at smaller time scales. We employ a simple, easy-to-implement—both on-line and off-line—prediction scheme to estimate $\Pr\{X_{i+1}^{(m)} \mid X_i^{(m)}\}$ based on observed empirical distribution. We note that *optimum estimation* is a difficult problem for LRD traffic [3], and its solution is outside the scope of this paper. Our estimation scheme provides sufficient accuracy with respect to extracting predictability and is computationally efficient, however, it can be substituted by any other scheme if the latter is deemed "superior" without affecting the conclusions of our results. To facilitate normalized contention levels, we define a map $L : \mathbb{R}_+ \to [1, s]$, monotone in its argument, and let $x_i^{(m)} = L(X_i^{(m)})$. Thus $x_i^{(m)} \approx 1$ is interpreted as the aggregate traffic level at time scale m being "low" and $L_k \approx s$ is understood as the traffic level being "high." The process $x_i^{(m)}$ is related to the *level process* used in [13] for modeling LRD traffic. We use L_1 and L_2 without reference to the specific time index i to denote *consecutive* quantized traffic levels $x_i^{(m)}$, $x_{i+1}^{(m)}$.

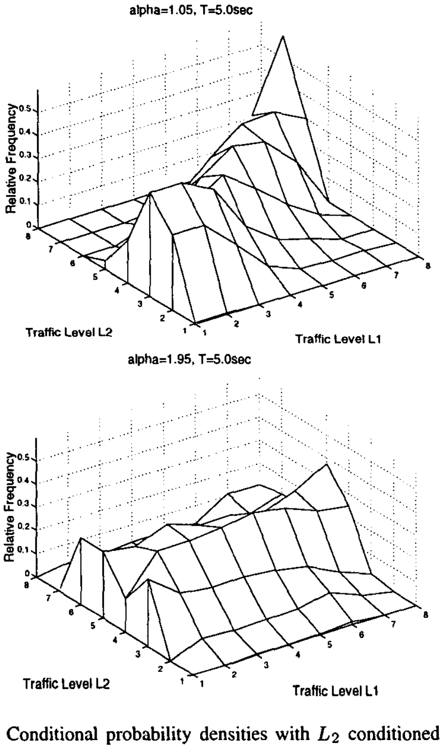

Fig. III.1. Conditional probability densities with L_2 conditioned on L_1 for LRD traffic (top) and SRD traffic (bottom).

Figure III.1 (top) shows the predictability structure of LRD traffic at a time scale of 5s by plotting its 3-D conditional probability densities. The diagonal skewness indicates that conditioning on L_1 is informative with respect to predicting L_2. Figure III.1 (bottom) shows the corresponding densities for short-range dependent traffic. We observe that conditioning has negligible influence.

C. Coupling

Multiple time scale congestion control allows for n-level time scale congestion control ($n \geq 1$) where information extracted at n separate time scales $T_1 < T_2 < \cdots < T_n$ is cooperatively engaged to modulate the output behavior of the feedback congestion control residing at the lowest time scale (i.e., $n = 1$). The objective of MTSC is to improve performance vis-à-vis the congestion control consisting of the feedback congestion control alone. We concentrate on 2-time scale congestion control where the "large" time scale module C_L—separated by an order of magnitude from the "small" time scale module C_S—is coupled to the latter to yield a new control $C_{L \otimes S}$. For throughput maximization, for example, the coupling takes on a *multiplicative* form [29]. For QoS control using adaptive FEC, we employ *additive* coupling. The latter is illustrated in Figure III.2 where a "DC" level shift is instituted with respect to the large time scale rate which results in an increase of the base rate from λ_L to λ_H.

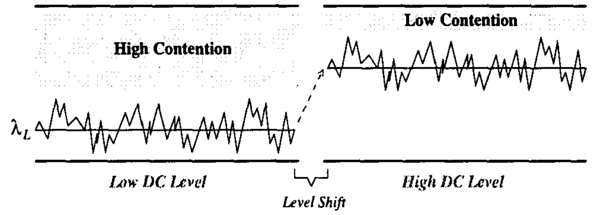

Fig. III.2. Additive coupling via selective "DC" level adjustment—i.e., level shift—between high- and low-contention periods.

IV. AFEC-MT

A. Multiple Time Scale Redundancy Control

AFEC-MT, in general, allows for n-level time scale redundancy control for $n \geq 1$ where information extracted at n separate time scales is additively coupled to yield the level of redundancy h applied at the FEC encoder. This is depicted in Figure IV.1.

Our design methodology is based on devising the large time scale module at time scale T_2, attaching it to the AFEC module operating at time scale T_1 of the feedback loop to improve the control actions undertaken by AFEC. Our objective is to show that this modular extension results in a control which is able to achieve significant performance gains relative to AFEC. In the 2-time scale redundancy control setting, the explicit prediction component of the large time scale module C_L outputs a redundancy level $h_L (\equiv h_2)$ which is then *additively* combined with the redundancy level $h_S (\equiv h_1)$ computed by C_S, i.e., AFEC. $h = h_S + h_L$ is then passed on to the FEC encoder component of AFEC. In specifying the control law

$$\frac{dh}{dt} = \frac{dh_S}{dt} + \frac{dh_L}{dt}$$

the control governing h_S is AFEC (cf. Section II-B) hence needs no separate description.

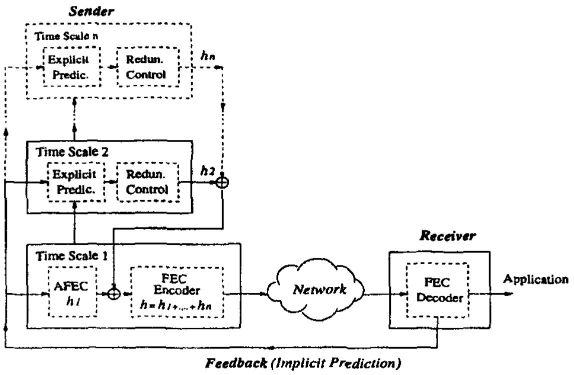

Fig. IV.1. AFEC-MT framework. Dashed lines show the potential extensibility of the framework to three or more time scales.

B. Structure of C_L

Whereas dh_S/dt is affected at the time scale T_1 of AFEC's feedback loop, dh_L/dt is affected at the much larger time scale T_2. This implies that the frequency of updates for h_S are much greater than that of h_L. Now to the description of dh_L/dt.

B.1 Explicit Prediction

The explicit prediction component of C_L performs per-connection on-line estimation of the conditional probability densities $\Pr\{L_2 \,|\, L_1 = k\}$, $k \in [1, s]$, following the method outlined in Section III-B. It turns out that *on-line* estimation can be accomplished using $O(1)$ operations at every update interval, i.e., C_L's time scale T_2. On the sender side, C_L maintains a 2-dimensional array CondProb[·][·] of size $s \times (s+1)$, one row for each $k \in [1, s]$. The last column of CondProb, CondProb[k][s+1], is used to keep track of the number of blocks observed thus far whose traffic level map to k. Since $\Pr\{L_2 = \ell \,|\, L_1 = k\} =$ CondProb[k][ℓ]/CondProb[k][s + 1], having the table CondProb means having the conditional probability densities.

B.2 Selective Redundancy Control

Using the conditional probability density table CondProb, we compute the expectation of L_2 conditioned on L_1, $\ell = \mathrm{E}(L_2 \,|\, L_1 = k)$, $\ell \in [1, s]$, at the end of each L_1. ℓ is then used to index the *redundancy schedule* $H : [1, s] \to \mathbb{R}_+$ to yield the value

$$h_L := H(\ell).$$

What remains is the computation of the function H. Let $h^1, h^2, \ldots, h^s$ denote the function values of H. Each component h^ℓ is updated according to the symmetric control

$$\frac{dh^\ell}{dt} = \begin{cases} \nu, & \text{if } d\gamma/dh^\ell > 0 \text{ and } \gamma_* > \gamma, \\ -\nu, & \text{if } d\gamma/dh^\ell < 0 \text{ or } \gamma - \gamma_* > \theta, \end{cases}$$

where $\nu > 0$ is an adjustment factor and $\theta \geq 0$ is a threshold parameter. The sign of $d\gamma/dh^\ell$ can be estimated by maintaining a history of redundancy action-QoS impact pair sequences, one for each contention level $\ell \in [1, s]$. At time $t = 0$, the initial values of H are set to zero. The value of h^ℓ, $\ell \in \{1, s\}$, is updated at the end of each block L_1 whose conditional expectation maps to ℓ.

V. Real-time MPEG Video Transport

A. System Structure

We have built an implementation of AFEC-MT customized for the transport of real-time MPEG video. For brevity, we will refer to this system as AFEC-MT in the following sections. AFEC-MT consists of a number of modules including the FEC codec, receiver-side controller C_R, and sender-side controllers C_S^1 and C_S^2. C_S^1 adjusts short-range redundancy h_1 by reacting to feedback at the time scale of RTT. C_S^2 is implemented "on top" of C_S^1 via the coupling described in Section IV and sets the long-range redundancy level h_2. The net redundancy $h = [h_1 + h_2]^+$ is then input to the FEC encoder. The system can be configured to turn off either one of the two control modules. Thus when C_S^2 is disabled, the system degenerates to AFEC. On the sender side, a stream of I, P, B frames is generated by an MPEG encoder at some frame rate f (e.g., 25–30 frames/sec). The FEC encoder applies forward error correction on each frame producing a sequence of $n = k + h$ packets which are submitted to the network.

At the receiver, upon receiving a sequence of packets belonging to the same frame, the controller C_R checks if at least k packets have arrived in a timely manner. If the number of timely packets is less than k, then the packets belonging to the current frame are discarded including those arriving subsequently. If at least k packets arrived within their deadline, then the first k packets (any k-subset will do) are forwarded to the FEC decoder proper which decodes the packet stream to recover the original k data packets constituting the sender's frame. The I, P, or B frame is then forwarded to the MPEG II player which applies its own decoding to produce the uncompressed frame which is then rendered on the terminal as part of the video stream. We employ Rabin's IDA [27] as the FEC codec. Details of the FEC implementation can be found in [22], [23]. C_R computes certain control information including

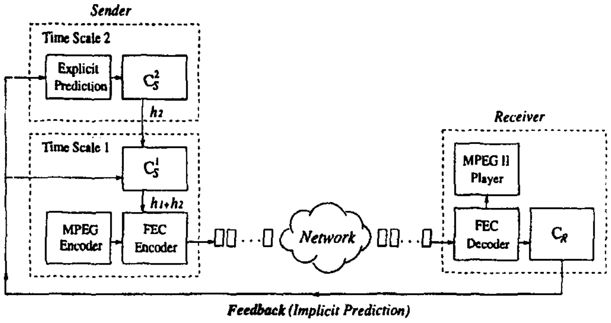

Fig. V.1. System structure of AFEC-MT.

the number of timely arrived packets γ which is then fed back via a control packet to the sender. All of the modules are implemented in software, and it is completely end-to-end such that the system can be deployed over any IP network. The forward flow structure is shown in Figure V.1.

B. Experimental Set-up

B.1 Network Configuration

Our experiments were carried out in the test environment shown in Figure V.2. AFEC-MT, consisting of the AFEC-MT sender and receiver, sent its traffic over a router where cross traffic stemming from a separate cross traffic source was multiplexed with the application traffic. By varying the cross traffic characteristics as well as the resources (e.g., buffer capacity) at the router, a wide spectrum of contention levels could be produced. To facilitate a controlled environment for measurement purposes, the camera and MPEG I encoder at the sender were replaced by an emulator that fed the frames of *stored* MPEG I video at real-time frame rates to the AFEC-MT sender. Thus as far as AFEC-MT was concerned, its functioning was not affected since—in either case—I, P, B frames were input to the FEC encoder at real-time speeds[2]. The MPEG II player at the receiver was always run, and the performance measurements reflect the overhead incurred by the player and FEC decoding processing overhead. Thus modulo *where* the I, P, B frames came from at the sender-side, the system depicted in Figure V.1 was faithfully implemented with all the components implemented in software without specialized hardware support.

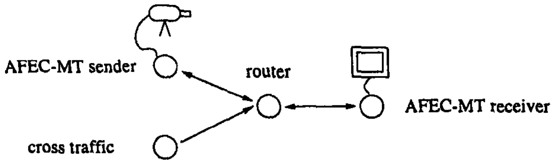

Fig. V.2. Experimental set-up for AFEC-MT performance measurements.

The topology depicted in Figure V.2 was realized over a private FastEthernet LAN environment with the AFEC-MT sender/receiver, cross traffic source, and router running on four Sun UltraSparc 1 & 2 workstations. Without the configurable router, we found that the 100 Mbps bandwidth of FastEthernet was too large relative to the data rate of MPEG I video (~1.5 Mbps), even with our cross traffic source active, to cause significant contention. With an UltraSparc 2 node acting as a configurable router implementing FIFO packet scheduling, a wide range of contention levels could be created under controlled conditions. The AFEC-MT protocol as well as the cross traffic source ran on top of UDP. AFEC-MT was realized as an application layer process portable to other UNIX environments.

B.2 Benchmark Traffic

Self-similar cross traffic was generated by utilizing traces from [21]. We emphasize that the performance results are obtained using actual MPEG I video rather than frame size traces commonly used in simulation and experimentation setups. Thus the cost of FEC encoding and decoding, control actions, and MPEG II player's processing overhead are all reflected in the performance results. We use the MPEG I video clip, Beauty and the Beast, as the main benchmark data source. The clip consists of 36000 frames which, at 15 f/s frame rate, has a duration of 40 minutes. Other real-time video data employed include Simpsons cartoon and Terminator clips.

C. Performance Measurement

C.1 Unimodal Redundancy-Recovery Relation

Figure V.3 (top) shows the measured redundancy-recovery function for the Beauty and the Beast MPEG I video clip transmission when using *static* FEC with *h fixed* in the range 0–14. We observe that the redundancy-recovery function is unimodal with peak at $h = 8$. Figure V.3 (bottom) shows the corresponding packet loss rate curve which monotonically increases with h. The increase in packet loss rate stems from the higher traffic rate associated with increased redundancy.

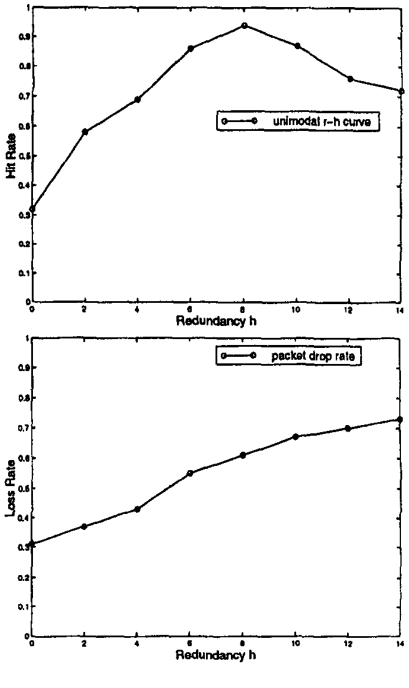

Fig. V.3. Top: Measured unimodal redundancy-recovery function for Beauty and the Beast MPEG I video clip. Bottom: Corresponding packet loss rate function.

[2] An implementation of the AFEC-MT sender which interfaces with an Optibase real-time MPEG I compression engine is available for Windows NT.

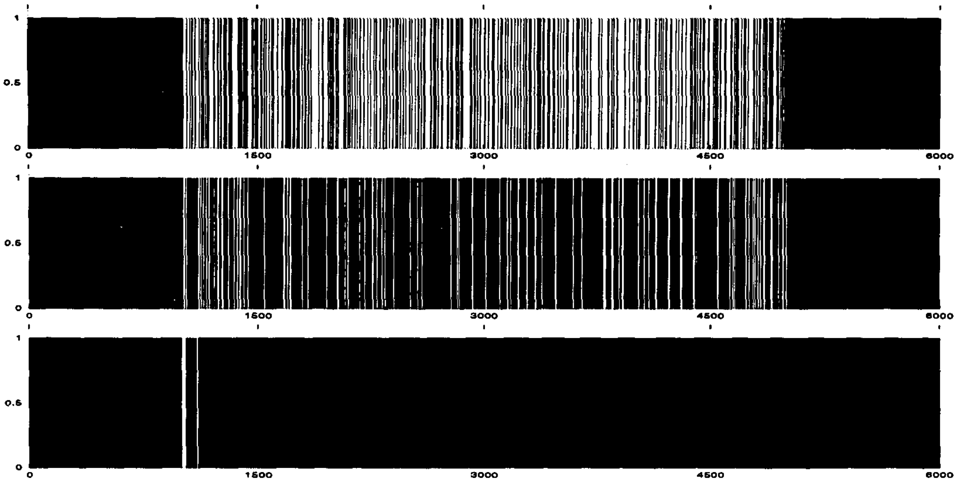

Fig. V.4. Hit trace for static FEC (top), AFEC (middle), and AFEC-MT (bottom) for self-similar cross traffic active during the middle time interval [1000,5000], silent otherwise.

C.2 AFEC-MT vs. AFEC

The dynamical performance of AFEC-MT vis-à-vis AFEC and static FEC can be gleamed from Figure V.4. The figures show impulse plots over time—represented as frame sequence numbers—where the presence of a unit impulse indicates that the corresponding frame was decoded timely at the receiver. The absence of unit impulse (i.e., white stripe) shows frames which did not meet their deadline. Figure V.4 (top) shows performance of static FEC with $h = 0$ when a self-similar cross traffic source was active during the middle time interval [1000,5000] while being silent otherwise. Interference of cross traffic sharing a FIFO queue at the router degrades performance of the application flow which manifests itself as degraded QoS—i.e., hit rate—during the middle interval. Figure V.4 (middle) shows corresponding performance of AFEC for the same set-up and cross traffic conditions. We observe that performance is significantly improved as shown by the reduction in missed deadlines which translates to improved end-to-end QoS as perceived by the user. Figure V.4 (bottom) shows performance of AFEC-MT which further improves upon AFEC's achieved QoS yielding a measured hit rate that is close to the user's desired hit rate. We observe a small interval of timeliness violation at time 1000—the onset of cross traffic—which is a nonstationary, unpredictable event to which AFEC-MT then adjusts subsequently.

Whereas Figure V.4 depicts "instantaneous" QoS, Figure V.5 shows the running average of measured hit rate at the receiver for a set-up involving 12000 frames during which the self-similar cross traffic was always active. Figure V.5 (left) shows mean hit rate for static FEC with $h = 0$ whose hit rate at 0.21 is significantly below the target hit rate of 0.92. The middle figure shows corresponding performance of AFEC which improves upon the performance of static FEC but is still noticeably below the user-specified target level. Figure V.5 (right) shows performance of AFEC-MT which converges to the target hit rate after a transient initial adjustment period. The target hit rate is reached much faster than indicated in the plot: the latter shows the long-term running average of instantaneous hit rates from time 0 onwards, not a local window.

We remark that the improvement of AFEC-MT over AFEC stems from multiple time scale AFEC's ability to more accurately discern short-term fluctuations from persistent changes which allows it to be less reactive to short-range variations. AFEC's advantage over static FEC lies in is its ability to tailor redundancy to current network state, increasing it when needed to shield QoS, and decreasing redundancy when not needed to reduce wastage of shared network resources. This trade-off between QoS and bandwidth imparts a cost for reducing redundancy in response to short-term changes in network state as, in the near future, a QoS violation is likely to arise due to reduced protection which will then subsequently trigger an increase in redundancy. Bandwidth may have been temporarily "saved" by decreasing redundancy in response to short-term changes, but given the *primary goal* of achieving a target QoS while minimizing bandwidth usage in so doing—a *secondary objective*—responsiveness to short-term fluctuations is undesirable. Thus AFEC-MT's ability to discriminate and undertake differentiated action with respect to redundancy in the form of a short-range component h_1 and long-range component h_2 endows it with enhanced prowess to deliver end-to-end QoS while being efficient with respect to its resource usage.

C.3 RTT and Proactivity

As the round trip time (RTT) associated with the feedback loop increases, the state information conveyed by feedback becomes more outdated, and the effectiveness of reactive actions undertaken by a feedback control diminishes. The penalty is

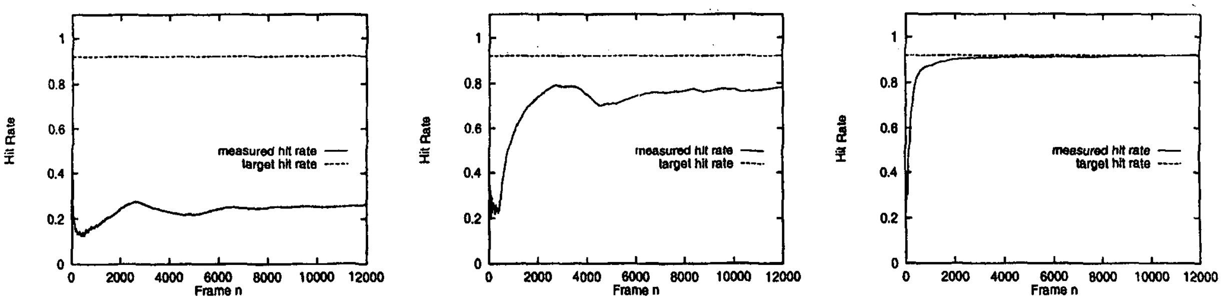

Fig. V.5. QoS performance comparison: static FEC (left), AFEC (middle), and AFEC-MT (right).

amplified in broadband wide area networks where the delay-bandwidth product increases proportionally with delay or bandwidth. Figure V.6 (left) shows measured hit rate as a function of RTT for both AFEC-MT and AFEC. We observe that AFEC's hit rate decreases significantly as RTT is increased which is commensurate with the effect of outdatedness of feedback information. AFEC-MT, on the other hand, suffers significantly less under the same conditions maintaining a much flatter performance curve. Thus AFEC-MT is able to mitigate part of the cost incurred by reactive control by exploiting long-range correlation structure to reduce the performance impact of outdated feedback.

Figure V.6 (right) shows the relative performance gain of AFEC-MT vis-à-vis AFEC where *performance gain ν* is defined as

$$\nu = \frac{\gamma_{\text{AFEC-MT}} - \gamma_{\text{AFEC}}}{\gamma_{\text{AFEC}}}.$$

Assuming $\gamma_{\text{AFEC-MT}} \geq \gamma_{\text{AFEC}}$, $\nu \geq 0$ represents the percentage of improvement achieved by AFEC-MT over AFEC. We observe that ν increases with RTT indicating that AFEC's susceptibility to long round-trip times increases vis-à-vis the corresponding susceptibility of AFEC-MT.

C.4 Impact of Long-range Dependence

We have shown that the correlation structure in self-similar traffic—upon effective utilization by AFEC-MT—leads to enhanced QoS above and beyond what AFEC can provide. Yet another dimension of interest is the impact of long-range correlation structure—i.e., its strength—on performance gain. Table I shows the hit rate of AFEC, AFEC-MT, and performance gain when the long-range dependence present of cross traffic is increased from weak ($\alpha = 1.95$) to strong ($\alpha = 1.05$). Network traffic measurements correspond to traffic with $\alpha \approx 1$. The α measure is related to the Hurst parameter for measuring long-range dependence, and we refer the reader to [21], [30] for a more detailed discussion.

Table I shows that performance gain amplifies as long-range dependence is increased with α approaching 1. Thus at the same time that long-range dependence exerts a negative influence on performance from the queueing perspective, the same structure can be exploited to affect control decisions that mitigate the very performance effects that are caused by long-range correlations in the first place. This is the "good news within the bad news" syndrome [29]. We remark that when varying the long-range dependence associated with cross traffic, it is important that all other things are kept equal—including the average traffic rate—to preserve comparability. Generating traffic loads with this property is a nontrivial task due to sampling error introduced when engaging heavy-tailed distributions in physical traffic models. Details for generating normalized workloads is provided in [30].

TABLE I

IMPACT OF LONG-RANGE DEPENDENCE ON PERFORMANCE GAIN OF AFEC-MT VS. AFEC.

α	1.05	1.35	1.65	1.95
AFEC	0.764	0.831	0.882	0.920
AFEC-MT	0.919	0.918	0.921	0.917
Gain	20.3%	10.47%	4.42%	-0.0%

Table I also shows that when traffic is weakly correlated at large time scales ($\alpha = 1.95$), the performance difference between AFEC-MT and AFEC is minimal (0.917 vs. 0.920). This is not surprising since given that there is little structure at large time scales to exploit, the performance benefit ensuing from coupling AFEC with the large time scale control module should be commensurately low or nonexistent. At the very least, AFEC-MT should not "hurt" performance vis-à-vis AFEC which we find is the case.

D. Redundancy Schedule

Related to the impact of long-range dependence is the time evolution of the redundancy schedule $H(\cdot)$ which for predicted traffic level ℓ at time scale T_2 assigns the base redundancy $h_L = H(\ell)$. The update of the components of $H(\ell)$, $\ell = 1, 2, \ldots, 8$, is affected by the symmetric control law given in Section IV-B.2. Figure V.7 (top) shows the evolution of $H(\cdot)$ as a function of time (represented as frame numbers here) for $\alpha = 1.05$ traffic. Figure V.7 (bottom) shows the corresponding values for $\alpha = 1.95$ traffic. We observe that when large time

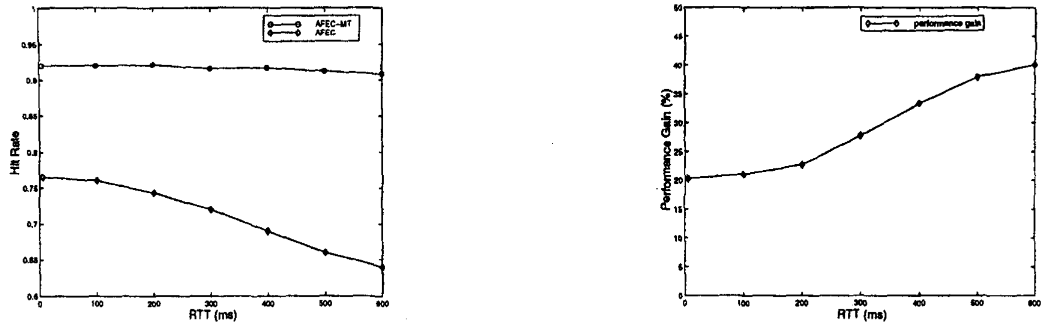

Fig. V.6. Proactivity of AFEC-MT as a function of RTT. Left: Hit rate of AFEC-MT vs. hit rate of AFEC. Right: Performance gain of AFEC-MT relative to AFEC.

scale correlation structure is weak, then the values are concentrated around a narrow range (i.e., 3–5) which points toward the fact that conditioning on predicted large time scale traffic level is of limited utility. For $\alpha = 1.05$ traffic, on the other hand, the redundancy values are spread out over a much wider range (i.e., 0–8) which indicates that conditioning does provide the ability to discriminate with respect to the future. In particular, this allows a base redundancy of $h_L = 8$ to be applied when traffic contention is high ($\ell = 8$) to facilitate increased protection, and a correspondingly small "DC" redundancy $h_L = 0$ when persistent traffic contention is low ($\ell = 1$).

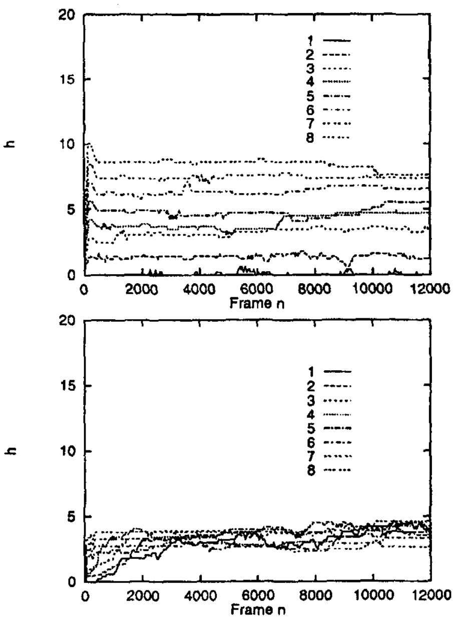

Fig. V.7. Evolution of redundancy schedule as a function of time. Top: $\alpha = 1.05$ traffic. Bottom: $\alpha = 1.95$ traffic.

D.1 Multiple AFEC-MT Connections

AFEC-MT is an end-to-end protocol designed to run in shared network environments where multiple AFEC-MT flows compete for available resources. As seen in the control laws for AFEC and its large time scale module that adjusts the redundancy schedule (cf. Sections IV-B.2 and II-B), AFEC-MT tries to achieve a target QoS—i.e., hit rate—while applying an amount of redundancy deemed adequate to do so, but not more. In this sense, it is a cooperative protocol, just like TCP, which stands in contrast to noncooperative protocols that would exploit other flows' cooperativeness and not back off even under adverse conditions[3], thereby absorbing their bandwidth [7]. In fact, if the user-specified target hit rate is sufficiently low, then AFEC-MT—when compared to TCP—may end up consuming less bandwidth than TCP which tries to maximize throughput. Of course, AFEC-MT can be transformed into a protocol which seeks to maximize QoS, in which case, its modus operandi is analogous to that of TCP: back-off is instituted only when a further increase in redundancy would result in a decreased hit rate.

Table II shows hit rate for a network environment consisting of three AFEC-MT connections. The three connections compete for resources at the same bottleneck router shown in Figure V.2 as for the single AFEC-MT connection case. The cross traffic course remains the same. Table II shows bandwidth sharing behavior for three network conditions—when the cross traffic source is sending at a rate of 6.4Mbps (network contention is high), 4.8Mbps, and 3.2Mbps (network contention is low). The target hit rate for each connection was set at 0.92. When available bandwidth is small (first column), network resources are insufficient to achieve the target hit rate 0.92. The flows remain stable due to the back-off mechanism of AFEC-MT, and all three connections end up sharing an approximately equal amount of bandwidth which results in similar hit rates of 0.56, 0.53, and 0.57, respectively. Note that this is equivalent to a "maximize QoS" mode. Bandwidth sharing behavior stays invariant as the traffic rate of the cross traffic source is decreased, eventually leading to all three connections achieving their target hit rate 0.92 (the second connection is a fraction short) when network contention is low (last column). Performance evalua-

[3] That is, network state where increasing redundancy leads to a decrease in hit rate for the flow in question.

tion with four or more connections were carried out using simulations yielding qualitatively similar results.

TABLE II

QoS PERFORMANCE OF MULTIPLE AFEC-MT CONNECTIONS WITH RESPECT TO FAIRNESS.

Cross Traffic	6.4 Mb/s	4.8 Mb/s	3.2 Mb/s
Connection I	0.56	0.82	0.92
Connection II	0.53	0.83	0.91
Connection III	0.57	0.79	0.92

VI. Conclusion

In this paper, we have introduced a multiple time scale extension of AFEC—a protocol that performs packet-level adaptive FEC for real-time payload transport [22], [23], [26]. A large time scale module that extracts long-range correlation structure in network contention was constructed and coupled with AFEC yielding its extension AFEC-MT. The large time scale module augmented the feedback redundancy control mechanism employed by AFEC by additively engaging a "DC" redundancy level as a function of predicted, long-range network state. We implemented AFEC-MT for real-time MPEG video transport and showed that significant performance gains are achievable by utilizing long-range correlation structure present in self-similar traffic. An important consequence of AFEC-MT is its ability to impart proactivity when subject to long round-trip times. By exploiting long-range correlation structure at time scales an order of magnitude higher than the RTT of the feedback loop, the detrimental performance effect of outdated feedback information is mitigated, which is especially severe in broadband wide area networks that possess a high delay-bandwidth product.

References

[1] O. Ait-Hellal, E. Altman, A. Jean-Marie, and I. Kurkova. On loss probabilities in presence of redundant packets and several traffic sources. *Performance Evaluation*, 36:485–518, 1999.

[2] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan. Priority encoding transmission. *IEEE Trans. Information Theory*, 42(6):1737–1744, 1996.

[3] Jan Beran. *Statistics for Long-Memory Processes*. Monographs on Statistics and Applied Probability. Chapman and Hall, New York, NY, 1994.

[4] E. Berlekamp, R. Peile, and S. Pope. The application of error control in communications. *IEEE Communications Magazine*, 25(4):44–57, 1987.

[5] Ernst Biersack. Performance evaluation of forward error correction in ATM networks. In *Proc. ACM SIGCOMM '92*, pages 248–257, 1992.

[6] J. Bolot, S. Fosse-Parisis, and D. Towsley. Adaptive FEC-based error control for Internet telephony. In *Proc. IEEE INFOCOM '99*, pages 1453–1460, 1999.

[7] S. Chen and K. Park. An architecture for noncooperative QoS provision in many-switch systems. In *Proc. IEEE INFOCOM '99*, pages 864–872, 1999.

[8] I. Cidon, A. Khamisy, and M. Sidi. Analysis of packet loss processes in high-speed networks. *IEEE Trans. Information Theory*, 39(1):98–108, 1993.

[9] I. Cidon, A. Khamisy, and M. Sidi. Delay, jitter and threshold crossing in ATM systems with dispersed messages. *Performance Evaluation*, 29(2):85–104, 1997.

[10] D. Clark and W. Fang. Explicit allocation of best-effort packet delivery service. *IEEE/ACM Trans. Networking*, 6(4):362–373, 1998.

[11] D. R. Cox. Long-range dependence: a review. In H. A. David and H. T. David, editors, *Statistics: An Appraisal*, pages 55–74. Iowa State Univ. Press, 1984.

[12] S. Dravida and R. Damodaram. Error detection and correction options for data services in B-ISDN. *IEEE J. Select. Areas Commun.*, 9(9):1484–1495, 1991.

[13] N. Duffield and W. Whitt. Network design and control using on-off and multi-level source traffic models with heavy-tailed distributions. In K. Park and W. Willinger, editors, *Self-Similar Network Traffic and Performance Evaluation*. Wiley Interscience, 1999.

[14] S. Kaneda and E. Fujiwara. Single bit error correcting double bit error detecting code for memory systems. *IEEE Trans. on Computers*, C-31(6):596–602, 1982.

[15] K. Kawashima and H. Saito. Teletraffic issues in ATM networks. *Computer Networks and ISDN Systems*, 20:369–375, 1990.

[16] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of ethernet traffic. In *Proc. ACM SIGCOMM '93*, pages 183–193, 1993.

[17] N. Likhanov, B. Tsybakov, and N. Georganas. Analysis of an ATM buffer with self-similar ("fractal") input traffic. In *Proc. IEEE INFOCOM '95*, pages 985–992, 1995.

[18] F. MacWilliams and N. Sloane. *The Theory of Error Correcting Codes*. North-Holland, New York, 1977.

[19] Anthony McAuley. Reliable broadband communication using a burst erasure correcting code. In *Proc. ACM SIGCOMM '90*, pages 197–306, 1990.

[20] K. Nichols, V. Jacobson, and L. Zhang. A two-bit differentiated services architecture for the Internet. Internet Draft, 1997.

[21] K. Park, G. Kim, and M. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proc. IEEE International Conference on Network Protocols*, pages 171–180, 1996.

[22] K. Park and W. Wang. AFEC: an adaptive forward error correction protocol for end-to-end transport of real-time traffic. In *Proc. IEEE IC3N*, pages 196–205, 1998.

[23] K. Park and W. Wang. QoS-sensitive transport of real-time MPEG video using adaptive forward error correction. In *Proc. IEEE Multimedia Systems '99*, pages 426–432, 1999.

[24] K. Park and W. Willinger. Self-similar network traffic: An overview. In K. Park and W. Willinger, editors, *Self-Similar Network Traffic and Performance Evaluation*. Wiley Interscience, 1999.

[25] Kihong Park. Warp control: a dynamically stable congestion protocol and its analysis. In *Proc. ACM SIGCOMM '93*, pages 137–147, 1993.

[26] Kihong Park. AFEC: an adaptive forward error-correction protocol and its analysis. Technical Report CSD-TR-97-038, Department of Computer Sciences, Purdue University, 1997.

[27] Michael Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM*, 36(2):335–348, 1989.

[28] N. Shacham and P. McKenny. Packet recovery in high-speed networks using coding. In *Proc. IEEE INFOCOM '90*, pages 124–131, 1990.

[29] T. Tuan and K. Park. Multiple time scale congestion control for self-similar network traffic. *Performance Evaluation*, 36:359–386, 1999.

[30] T. Tuan and K. Park. Performance evaluation of multiple time scale TCP under self-similar traffic conditions. Technical Report CSD-TR-99-040, Department of Computer Sciences, Purdue University, 1999.

[31] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. In *Proc. ACM SIGCOMM '95*, pages 100–113, 1995.