

A New Feedback Control Mechanism for Error Correction in Packet-Switched Networks

Oscar Flårdh, Karl H. Johansson and Mikael Johansson

Department of Signals, Sensors and Systems

KTH, SE-100 44, Stockholm, Sweden

Email: {oscar.flardh, kallej, mikael.johansson}@s3.kth.se

Abstract—Error correction mechanisms enable control and other real-time applications to be executed over unreliable packet-switched networks. By adding carefully adjusted redundancy to transmitted data at the sender, it is possible to recover lost data at the receiver and thereby improve effective throughput. We describe simple models for packet loss, which allow us to find the optimal redundancy as a function of packet loss probability. Two feedforward control mechanisms based on the packet loss probability are presented: one that is computed off-line and another one using an on-line algorithm. A drawback with these approaches is their dependency on accurate network state information and precise loss models. To cope with these issues, we propose a new feedback solution that tracks the optimum using gradient estimation. Simulations in ns-2 illustrate the behavior of the error correction schemes, demonstrating that the feedback solution outperforms the feedforward solution in presence of model errors.

I. INTRODUCTION

The emerging technology of network embedded control opens up vast possibilities for flexible automation solutions in application areas such as smart buildings, industrial automation and power distribution. At the same time, the desire to engineer viable networked systems raises many fundamental research challenges on predictability, reliability and scalability [1]. One important subproblem is on how to execute feedback control applications over unreliable communication channels. There are two natural approaches to address this problem. One is to modify each control algorithm to cope with the variations and uncertainties posed by the network [2]. Another approach is to develop generic interfaces between control applications and transport protocols that are suitable for a large class of control applications and network conditions. The latter approach is less studied in the literature, but considered in this paper. In general, these interfaces are aware of features of both control applications (e.g., control quality, stability) and networks (e.g., delay, jitter, packet loss, disconnection) [3], [4]. Here, the focus is on adaptation mechanism for packet losses.

The main contribution of the paper is a comparison of a set of end-to-end error correction algorithms that support control applications on top of an unreliable packet-switched network. By adapting the amount of transmitted redundancy, it is possible to cope with large changes in the packet loss probability. In particular, interesting features of feedforward and feedback control of error correction are discussed. The presented schemes are inspired by an adaptive forward error

correction (AFEC) method proposed by Park and Wang [5] for end-to-end transport of real-time traffic. Similar AFEC schemes have been applied to Internet telephony [6] and MPEG streaming [7]. We show that error correction control can be posed as an extremum control problem [8], in which the static nonlinear map corresponds to the normalized effective throughput. However, since the measurements are extremely noisy in our application, it seems hard to utilize more advanced extremum seeking methods. A simple scheme based directly on filtered derivatives of the redundancy and the effective throughput is shown to work well. In particular, it is illustrated how proper tuning of a single parameter makes the feedback loop generate sufficient excitation for appropriate estimation to stay close to the optimum. Simulations in ns-2 (the network simulator 2 [9]) with different loss models illustrate the results.

The outline of the paper is as follows. Section II presents the network model and the control problem. A feedforward control strategy for error correction based on estimates of loss probability is presented in Section III, whereas a more robust feedback control strategy based on a block recovery indicator is introduced in Section IV. The error correction schemes are evaluated through simulations described in Section V and the paper is concluded in Section VI.

II. PROBLEM FORMULATION

The considered error correction problem is defined next, by first introducing a simple network model followed by the control objective for error correction.

A. Network Model

Consider the following model for unreliable end-to-end communication over a packet-switched network. A sequence of data packets are transferred across the network using an unreliable transport protocol, such as UDP or RTP. Congestion and lossy links cause some packets to be dropped by the network. Let v_i be an indicator for successful transfer of data packet i , i.e.,

$$v_i = \begin{cases} 1, & \text{if packet } i \text{ was received successfully} \\ 0, & \text{otherwise} \end{cases}$$

We consider various models for packet loss. The simplest one assumes a fixed probability p for losing a packet, and that the packet losses are independent. To evaluate the scheme for correlated losses, we consider a two-state Markov model, also known as the Gilbert model [10]. The two states are “loss” and “no loss”, with the probability q_1 of going from “no loss” to “loss”, and q_2 of going back to “no loss”. The average packet loss probability is then equal to $q_1/(q_1 + q_2)$.

This work was partially supported by the European Commission through the Integrated Project RUNES and the Network of Excellence HYCON, by the Swedish Research Council, and by the Swedish Foundation for Strategic Research through an Individual Grant for the Advancement of Research Leaders.

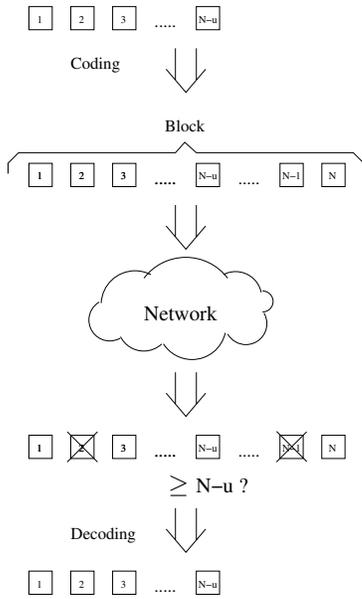


Fig. 1. Error correction. If at most u out of N sent packets in a block are lost, then it is possible to recover the original $N - u$ data packets of the block.

Consecutive packets are organized into blocks of fixed size N packets, and error correction is applied on block level as illustrated in Figure 1. Each block b contains a variable number $0 \leq u_b \leq N$ of redundancy packets and $N - u_b$ data packets (application information). It is assumed that the data in a block can be recovered if the number of packet losses is less than or equal to u_b . This is true for maximum distance separable codes (such as Reed Solomon codes), since the minimum distance for the code in this case is $u_b + 1$ [11]. Let I_b be the index set for data packets belonging to block b and introduce $x_b = \sum_{i \in I_b} v_i$ as the total number of received data packets in block b . Let z_b be the block recovery indicator for block b :

$$z_b = \begin{cases} 1 & \text{if } x_b \geq N - u_b \\ 0 & \text{otherwise} \end{cases}$$

For the implementation of the error correction mechanisms discussed below, it is important to properly choose the observation instances of the packet loss process. In order to decide when a packet is lost or not, a time-out interval between consecutive packets can be introduced similarly to what is done in TCP. This issue is not further discussed in the paper.

B. Control Objective

We focus on maximizing the normalized effective throughput, which we define as the average number of transmitted data packets per block over a horizon of B blocks

$$\frac{1}{BN} \sum_{b=1}^B (N - u_b) z_b$$

The control objective is to maximize the expected value of this quantity, which leads us to the following objective

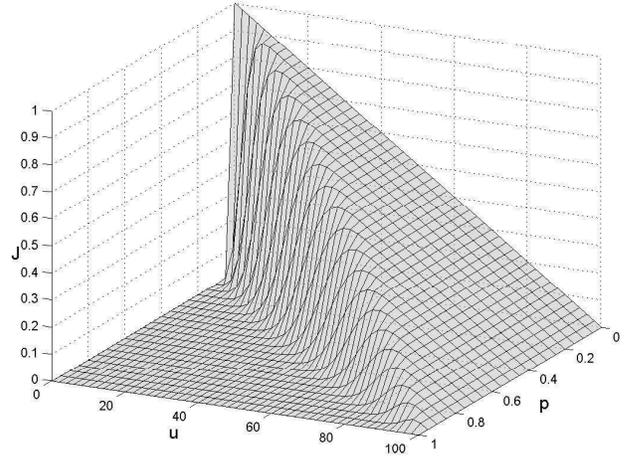


Fig. 2. Objective function J as a function of fixed redundancy u and independent packet loss with fixed probability p .

function

$$J(u) = \frac{1}{BN} \sum_{b=1}^B E \{ (N - u_b) z_b \} \quad (1)$$

It is interesting to qualitatively study how J is influenced by a fixed control $u_b = u$. A particular case is when the packet losses are independent and equal to p_b , say, for each packet in block b . Then, the probability of block recovery can be expressed as

$$\text{Prob}(z_b = 1) = \sum_{i=0}^{u_b} \binom{N}{i} p_b^i (1 - p_b)^{N-i} \quad (2)$$

Figure 2 shows J as a function of a fixed redundancy u for independent packet loss with fixed probability p . For large p and small u , the objective function is close to zero since the probability for block recovery is very low in this case. There is a threshold where the probability for block recovery increases rapidly with u to a value close to one, which corresponds to the ridge in J . Increasing u beyond this threshold increases the probability for block recovery very little, but decreases the number of data packets per block. In this situation, J thus decreases approximately linearly with u . For a given loss probability, there is a unique maximum and hence a unique optimal control. It is shown below that even if packet losses are correlated, it is still possible to argue locally in time and space for optimizing J .

C. Control Structure

The error correction control mechanisms we consider can all be framed in the control structure illustrated in Figure 3. The control signal u_b is computed in real-time based on feedforward of network state information and feedback information on the block recovery. The network state information corresponds to the packet loss probability p_b . This feedforward approach is further described in next section. The feedback information in Figure 3 is defined by $y_b = (N - u_b) z_b$, the number of recovered packets per block. Note that y_b can be seen as an observation of the objective function J . The feedback approach is based on

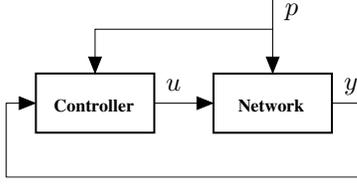


Fig. 3. Control structure for error correction with indication of feedforward and feedback information.

this measurement and is further described in Section IV. Obviously, the feedforward approach should respond faster but be more sensitive to modelling errors, while the feedback approach is expected to respond slower but be able to adapt to unmodelled changes.

From an implementation point of view, it should be noted that both the feedforward and the feedback signal are available at the receiver side: The packet loss probability p_b can be estimated and the feedback information on the block recovery y_b is known. Since the error correction is applied on the sender side, this information needs to be communicated to the sender over some possibly unreliable network. How the unreliable transfer of p_b and y_b affects control performance is not further discussed in the paper. Note, however, that the amount of data transferred in this direction is substantially smaller than the application data sent in the forward direction.

III. FEEDFORWARD CONTROL

We develop two control strategies in this section, based on feedforward information on the loss probability p . If p is known, the objective function (1) can be maximized. First we note that, if u_b is assumed to be fixed and thereby not stochastic,

$$E\{(N - u_b)z_b\} = (N - u_b)E\{z_b\}$$

and then using (2) for the probability of block recovery. Each term in the sum in (1) is given by

$$(N - u_b) \sum_{i=0}^{u_b} \binom{N}{i} p_b^i (1 - p_b)^{N-i}$$

Hence the optimal redundancy, if losses are independent, is given by

$$u_b^* = \arg \max_{u_b} (N - u_b) \sum_{i=0}^{u_b} \binom{N}{i} p_b^i (1 - p_b)^{N-i} \quad (3)$$

This optimization problem can be solved off-line, and then the control is a simple table look-up based on the current packet loss probability p . Hence we will call this control strategy the *off-line feedforward algorithm*. To efficiently derive the optimum redundancy (3), we consider the difference $\Delta L_u = L_u - L_{u-1}$, where

$$L_u = (N - u) \sum_{i=0}^u \binom{N}{i} p^i (1 - p)^{N-i}$$

It is easy to show that

$$\begin{aligned} \Delta L_u &= (N - u) \binom{N}{u} p^u (1 - p)^{N-u} \\ &\quad - \sum_{i=0}^{u-1} \binom{N}{i} p^i (1 - p)^{N-i} \end{aligned}$$

which gives a first-order condition for optimality: if $\Delta L_u > 0$ and $\Delta L_{u+1} < 0$, then u is the optimal redundancy.

Alternatively, we can attempt to find the optimum redundancy (3) using the following recurrence scheme

$$u_{b+1} = \begin{cases} u_b + \alpha_1 (J(u_b^*) - J(u_b)), & \text{if } u_b \leq u_b^* \\ \alpha_2 u_b, & \text{otherwise} \end{cases} \quad (4)$$

where $\alpha_1 > 0$ and $0 < \alpha_2 < 1$ are tuning parameters. We will call this scheme the *on-line feedforward algorithm*. If the redundancy is too low, it is increased by an amount proportional to the difference between the current objective and the optimum, whereas an exponential backoff is applied when the redundancy is too high. This control scheme is similar in spirit to the AFEC scheme proposed by Park and Wang [5]. However, since they consider another objective function and do not seek the maximum of that function, their solution is slightly different from (4).

The two control strategies described in this section adapt fast to changes in p , under the assumption that a good estimate of p is available. If the underlying model or the estimate of p is not accurate, the control strategies might give unsatisfactory performance. Next, we present a feedback strategy that tries to remedy these drawbacks.

IV. FEEDBACK CONTROL

The *feedback control algorithm* attempts to move the system towards the peak of J without relying on a detailed analytical model. Although it is hard to derive a closed-form expression for J under a general packet loss process, it is still possible to qualitatively study the shape of J . We may note that J will necessarily tend to zero as u approaches N . Moreover, if p is not zero J will be small for $u_b = 0$ and will initially increase as the redundancy increases, at least for independent losses. The detailed behavior of J for intermediate values is hard to specify, but we will make the assumptions that J is unimodal and monotone on each side of the peak.

As discussed in Section II-C, the feedback signal y_b gives information about J for the current redundancy level. Since the location of the peak and even J itself are uncertain, we simply estimate on which side of the peak the state is, and then adjust u so that the state moves closer to the peak. The estimation is based on the derivatives of y_b and u_b . However, since the underlying signals are very noisy, we need to filter them carefully to obtain a reliable estimate. To this end, we compute y_b^d and u_b^d by first averaging the signals y_b and u_b over a sequence of blocks, and then filtering the signals through a low-pass filter with a derivative term. By comparing the signs of y_b^d and u_b^d we can decide on which side of the optimum the current control value correspond to, cf., Figure 2. A control algorithm based on this argument is

$$u_{b+1} = u_b + \beta \operatorname{sgn}(y_b^d u_b^d) \quad (5)$$

where $\beta > 0$ is a tuning parameter.

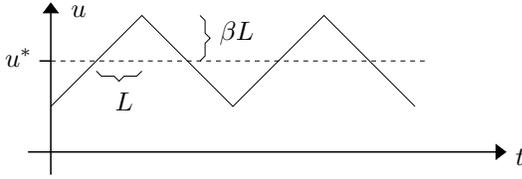


Fig. 4. Principal sketch of the solution to (6). The number of redundant packets will vary around u^* with period $4L$ and amplitude $2\beta L$.

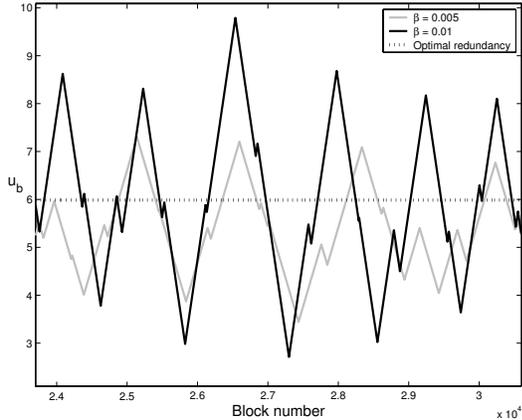


Fig. 5. Simulation of control algorithm (5) using independent packet losses.

To determine good values of β and to investigate closed-loop system characteristics, we consider a fluid model of (5):

$$\dot{u} = \beta \operatorname{sgn}(y^d u^d) \quad (6)$$

The control signal u is hence linearly increasing or decreasing with slope β . This controller fits into the class of switching extremum controllers, which have been traditionally used in other control applications [12]. The control signal will increase until the filtered signals indicate that the peak has been reached. Given that there is a delay L after the peak is passed until it is actually discovered, the solution of (6) will be a sawtooth signal with period $4L$ and amplitude $2\beta L$, see Figure 4. Simulations with independent packet losses confirm this behavior. Figure 5 show simulations for two different values of β . The signals are sawtooth-shaped, but there is discrepancy due to noise caused by the loss process. For $\beta = 0.01$ we have a period of about 1200 blocks, so our analysis predicts the amplitude $2\beta \approx 6$, which corresponds well with the simulation result. For $\beta = 0.05$, we have period 1300 and thus an amplitude just above 3, which also agrees well with the simulations.

Note that u will vary around u^* with an amplitude depending linearly on β . The smaller the amplitude, the higher the effective throughput. With u varying in an interval, the effective throughput will be the mean value of the function in that interval. This analysis suggests that control performance improves as β decreases. However, the value of β influences the gain in the control loop, and a higher gain gives faster response during transients. Furthermore, if the value of β is too small, the controller will not excite the system enough to detect changes in network conditions and might fail to find the optimal redundancy. To illustrate this, we investigate how the effective throughput varies with β during a step in packet

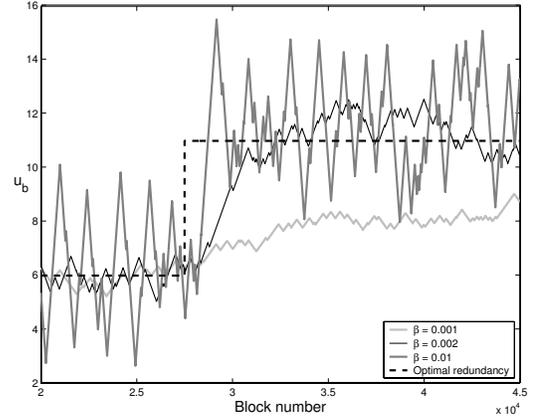


Fig. 6. Transients of the control signal u with three different values of β .

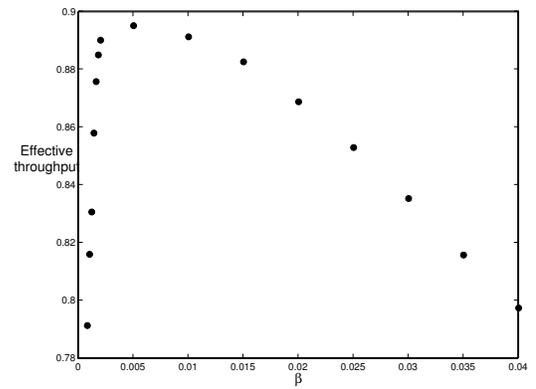


Fig. 7. The effective throughput as a function of β during a step in packet loss probability.

loss probability. Responses for three different values of β are shown in Figure 6. Here it is clearly illustrated that a small β gives small oscillations but slow response to changes in p , whereas a large β gives faster response but large oscillations. To generalize this observation, one can find the effective throughput as a function of β for this particular step response. This is presented in Figure 7, where both the threshold for enough excitation (small β) and the performance degradation due to large oscillations (large β) is shown.

V. SIMULATIONS

The control algorithms have been evaluated via extensive simulations in ns-2. We report results for independent packet losses as well as correlated losses generated by the Gilbert model described in Section II. In both cases, we evaluate the performances for a scenario with two sudden changes in the average packet loss probability. This behavior of the packet loss process is motivated from observations of real Internet traffic [13]. To avoid effects of inaccurate filtering, the two feedforward controllers are assumed to know the correct value for the packet loss probability in all simulations.

Figure 8 shows a simulation for the case of independent losses and Figure 9 shows the associated redundancy for the three control algorithms. From the detailed view of the transient shown in Figure 10 we see that the feedforward algorithms react quickly to changes in packet loss probability.

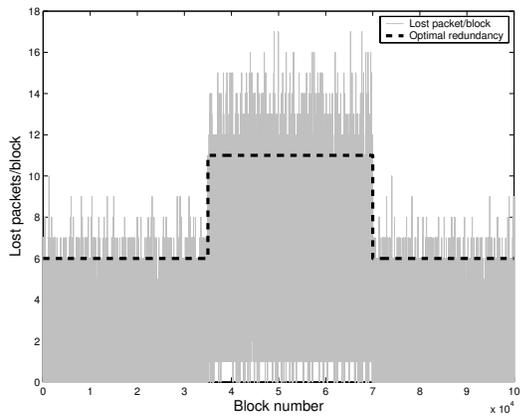


Fig. 8. Lost packets per block and optimal redundancy for the simulation with independent packet losses.

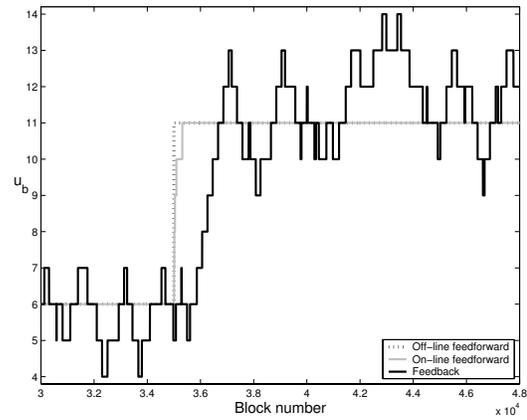


Fig. 10. Transients of the control signal during a sudden increase in packet loss probability. Simulation with independent losses.

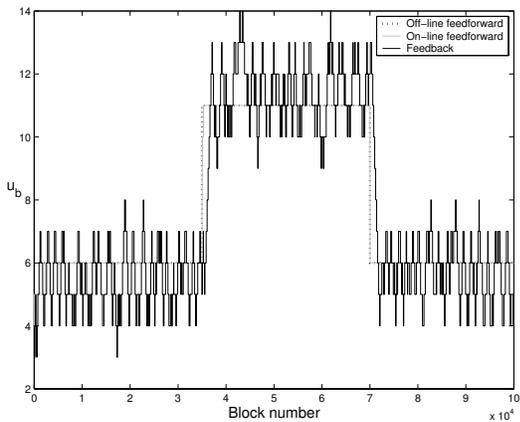


Fig. 9. Control signals during sudden change in packet loss. Simulation with independent losses. Note the overlap between the two feedforward algorithms.

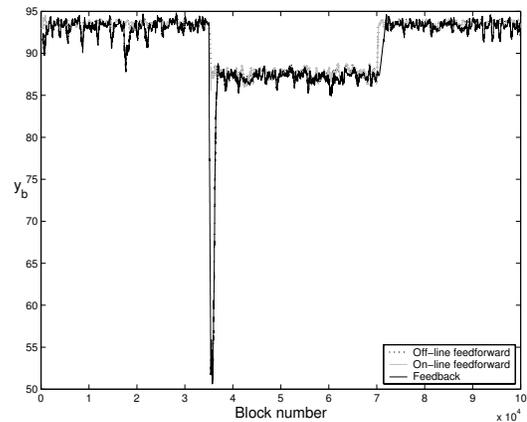


Fig. 11. Output signals during sudden change in packet loss. The signals are filtered. Simulation with independent losses.

This is natural since they have direct knowledge of the network state. The feedback algorithm is a bit slower, but it still manages to find the optimal redundancy, thereby achieving good steady state performance. The effects of this slower adaption is also clearly shown in Figure 11. The reason for the large dip during the increase in packet loss probability is that it is much worse to use too little redundancy than too much, if the discrepancy is more than a few packets, *cf.*, Figure 2. From the output signals, we can also conclude that the oscillations in the control signal have a minor effect on the effective throughput during steady state. Numerical values for the effective throughput and block recovery rates achieved by the different strategies are below.

	J	Recovery rate
Off-line feedforward	0.915	99.2 %
On-line feedforward	0.915	99.1 %
Feedback	0.906	98.1 %

Even though the on-line feedforward algorithm has slightly slower transients than the off-line solution using optimal redundancy all the time, it performs nearly as well. The feedback algorithm, with its slower transients and sustained oscillations around the optimal redundancy, performs slightly worse.

The second set of simulations investigates the performance of the different schemes under correlated losses generated by the Gilbert model. The control signals of the three control algorithms are shown in Figures 12 and 13. In this case, the feedforward algorithms are computed under the assumption of independent losses and thus underestimate the optimal redundancy. The feedback algorithm, on the other hand, stabilizes on a higher level of redundancy than the other schemes and achieves the best performance of the three:

	J	Recovery rate
Off-line feedforward	0.861	93.3 %
On-line feedforward	0.861	93.3 %
Feedback	0.872	97.3 %

The output signals are shown in Figure 14, where it is seen that the feedback controller performs worse during the transients, but finds better steady state values.

The underestimation of the redundancy by the feedforward algorithms is natural since, if packets are likely to be lost in bursts, the probability for losing a block increases even though the average packet loss probability is the same. Note also that the correlation of losses causes a general performance degradation. This is a well-known phenomenon in forward error correction [14], [15], and is due to the

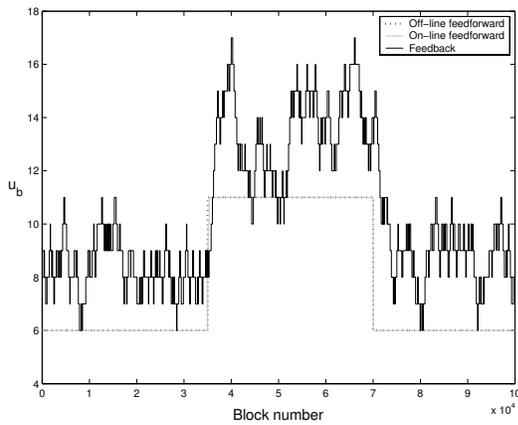


Fig. 12. Control signals during sudden change in packet loss. Simulation with correlated losses. Note the overlap between the two feedforward algorithms.

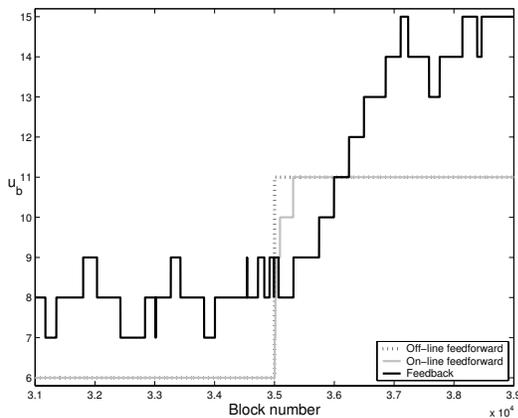


Fig. 13. Transients during a sudden increase in packet loss probability. Simulation with correlated losses.

increased block loss probability: Keeping the same level of redundancy as if the losses were independent will decrease the effective throughput (since more blocks are lost); increasing the redundancy gives higher block recovery probability but less application data is sent and the effective throughput is decreased.

VI. CONCLUSIONS

This paper has studied several control structures for error correction. The first two were feedforward controllers, one on-line and one off-line, based on packet loss probability for the connection. A drawback with this approach is the dependency on accurate network state information and the reliance on uncertain models. To cope with these issues, we proposed a new feedback solution that tracks the optimum using gradient estimation. The controllers are evaluated with simulations in ns-2, demonstrating that the feedforward controller performs well when there is no model error, whereas the feedback controller performs better in the presence of model errors.

There are several issues that warrants further investigations. One is how to best estimate the gradients despite the high noise level in signals, another one is to develop

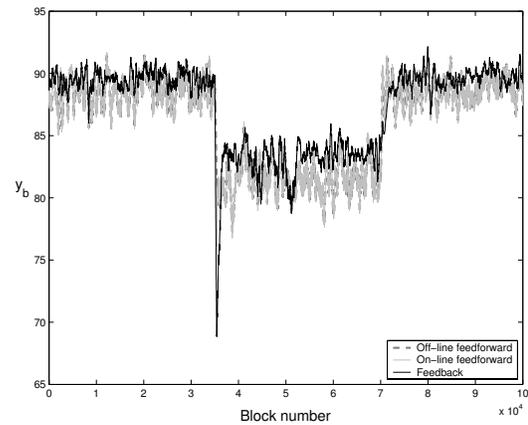


Fig. 14. Output signals during sudden changes in packet loss. The signals are filtered. Simulation with correlated losses.

systematic ways for tuning the feedback algorithm accounting for both stability and performance. It would also be interesting to apply other extremum control approaches to the forward error correction problem, as well as control structures combining feedforward and feedback components. Moreover, the sensitivity with respect to lost packets in the feedback signal needs further studies.

VII. ACKNOWLEDGMENTS

The authors wish to thank the reviewers for their useful comments. We are also indebted to Inés Cabrera Molero for her help with the ns-2 simulations.

REFERENCES

- [1] D. Estrin, Ed., *Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers*. U.S. National Research Council, 2001.
- [2] P. Antsaklis and J. Baillieul, "Special issue on networked control systems," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, 2004.
- [3] S. Graham, G. Baliga, and P. R. Kumar, "Issues in the convergence of control with communication and computing: Proliferation, architecture, design, services, and middleware," in *IEEE CDC*, 2004.
- [4] J. Nilsson, B. Bernhardsson, and B. Wittenmark, "Stochastic analysis and control of real-time systems with random time delays," *Automatica*, vol. 34, Jan. 1998.
- [5] K. Park and W. Wang, "AFEC: an adaptive forward error-correction protocol and its analysis," Technical Report CSD-TR97-038, Department of Computer Sciences, Purdue University, Tech. Rep., 1997.
- [6] J.-C. Bolot, S. Fosse-Parisis, and D. F. Towsley, "Adaptive FEC-based error control for internet telephony," in *INFOCOM (3)*, 1999, pp. 1453–1460.
- [7] K. Park and W. Wang, "QoS-sensitive transport of real-time MPEG video using adaptive forward error correction," in *ICMCS, Vol. 2*, 1999, pp. 426–432.
- [8] K. B. Ariyur and M. Krstić, *Real-Time Optimization by Extremum Seeking Control*. John Wiley & Sons Inc, 2003.
- [9] "The network simulator 2," <http://www.isi.edu/nsnam/ns/>.
- [10] W. Jiang and H. Schulzrinne, "Modeling of packet loss and delay and their effect on real-time multimedia service quality," in *Proc. NOSSDAV*, 2000.
- [11] R. E. Blahut, *Theory and Practice of Error Control Codes*. Addison-Wesley, 1983.
- [12] J. Sternby, "A review of extremum control," Department of Automatic Control, Lund Institute of Technology, Tech. Rep., April 1979.
- [13] K. Jacobsson, N. Möller, K. H. Johansson, and H. Hjalmarsson, "Some modeling and estimation issues in traffic control of heterogeneous networks," in *International Symposium on Mathematical Theory of Networks and Systems*, Leuven, Belgium, 2004.
- [14] I. Cidon, A. Khamisy, and M. Sidi, "Analysis of packet loss processes in high-speed networks," *IEEE Transactions on Information Theory*, vol. 39, no. 1, 1993.
- [15] N. Shacham and P. McKenny, "Packet recovery in high-speed networks using coding and buffer management," in *IEEE INFOCOM*, 1990.