J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*

AUTOMATIC REPEAT REQUEST

Automatic repeat request (*ARQ*) techniques are becoming pivotal not only for reliable data transmission, but also for real-time video and voice applications. ARQ lends itself to the data link layer (layer 2 of the Open Systems Interconnection (*OSI*) layered network architecture), to layer 4 of the transmission control protocol (*TCP*) of the Internet architecture (TP4 of OSI), and to certain application protocols (1).

In pure ARQ protocols, data are transmitted in frames, the last part of which carries certain parity check bits called cyclic redundancy check (*CRC*). These bits are algebraically related to the frame content. This relation is checked at the receiver [intermediate user if the ARQ is related to the data link layer (*DLC*), or final destination user if it is related to the TCP]. If this checking confirms the known algebraic relation, the frame is accepted as valid. If not, the user (ARQ entity) asks the transmitter to retransmit the frame (2).

The results of CRC are correct with high probability (3), and the probability of not detecting errors in the frame is very small. The length of the CRC field of the frame is typically 16 or 32 bits, which is very small compared to the total frame length. This implies very high bandwidth efficiency (typically higher than 95%). In contrast, forward error correction (*FEC*) techniques [3] have much lower bandwidth efficiencies (typically 1/16 to 7/8).

In pure FEC, one-way transmission from transmitter to receiver adds redundancy bits so as to enable error correction at the receiver. A return channel from receiver to transmitter is not necessary if pure FEC is employed.

If the probability of transmission errors on the link (channel) is very small, ARQ is efficient for concealing these error effects; otherwise, FEC may be more efficient.

Hybrids of ARQ and FEC are used extensively in many applications (4), and recently there has been growing interest in such hybrids for improving the quality of service (*QoS*) in multicast over the Internet.

Stop-and-Wait Automatic Repeat Request

In stop-and-wait (*SW*) ARQ, the transmitter (Fig. 1) sends an information frame of duration T_{f} and waits for a timeout T_{o} for positive acknowledgement before transmitting the frame again at time $T_{\mathrm{f}} + T_{\mathrm{o}}$. The propagation time is τ in each direction, and the acknowledgement frame duration is T_{A}.

The processing time [at the receiver to check the arriving frame, and at the transmitter to process the acknowledgment (*ACK*)] is given by $T_{\mathrm{p}} = T_{\mathrm{pf}} + T_{\mathrm{pa}}$. The transmitter timeout T_{o} is at least $T_{\mathrm{p}} + T_{\mathrm{A}} + 2\tau$. Under ideal conditions (no channel errors, no network congestion, etc.), the SW ARQ's efficiency or maximum throughput is given by (1)

$$\eta = \frac{T_{\mathrm{f}}}{T_{\mathrm{f}} + 2\tau + T_{\mathrm{p}} + T_{\mathrm{A}}} \tag{1}$$

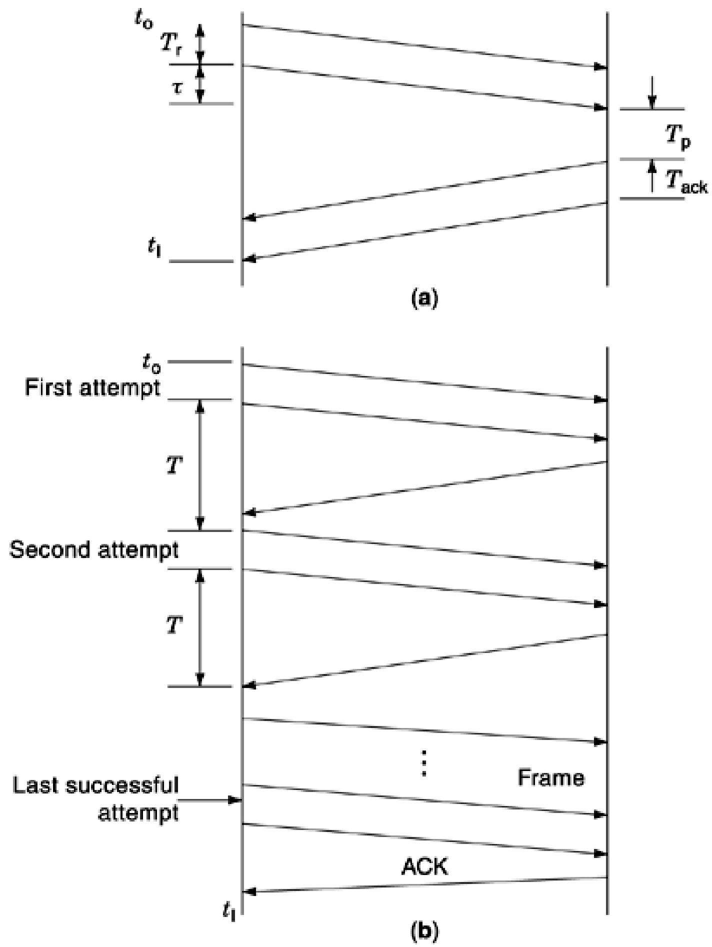

Fig. 1. Stop-and-wait protocols: (a) no errors, (b) with errors. Data transmitted by sender and ACK by receiver. Retransmission takes place if ACK is not received by timeout.

However, as in Fig. 1, the frame may have to be transmitted N_f times on average; $N_f - 1$ of these transmissions cost $T_f + T_o$, and the last, successful one costs $T_f + 2\tau + T_p + T_A$, leading to

$$\eta = \frac{T_f}{(T_o + T_f)(N_f - 1) + (T_f + 2\tau + T_p + T_A)} \tag{2}$$

To evaluate N_f, we assume that the probability of frame error and loss is given by P, and the transmission trials (Fig. 1) are repeated $i - 1$ times with probability P^{i-1}. The total number of transmissions, i, is a geometrically distributed random variable whose statistical average is given by

$$N_f = \sum_{i=1}^{\infty} i(P)^{i-1}(1 - P) = \frac{1}{1 - P} \tag{3}$$

Substituting Eq. (3) into Eq. (2) yields

$$\eta = \frac{T_f}{P(T_o + T_f)/(1-P) + (T_f + 2\tau + T_p + T_A)} \tag{4}$$

In the ideal case $P \approx 0$, $T_A \approx 0$, $T_p \approx 0$, we have

$$\eta = \frac{1}{1 + 2\tau/T_f} = \frac{1}{1+2a} \tag{5}$$

where

$$a = \tau/T_f. \tag{6}$$

Since $\tau = d/c$ and $T_f = L/R$, where d is the geographical distance in meters, c is the speed of light (3×10^8 m/s). L_f is the number of bits per frame, and R is the data rate in bits per second, η then becomes

$$\eta = \frac{1}{1 + (2dR/L_f c)} \tag{7}$$

From this one can see that for high values of R and d (typical of satellite links and long-haul networks), the SQ ARQ η is very low. Examples: Substituting $R = 100$ Mbit/s, $d = 60$ km, $L_f = 104$ bits in Eq. (7) yields $\eta = 0.2$, while $R = 10$ Mbit/s, $d = 9$ km, $L_f = 2000$ bits yields $\eta = 0.77$.

A one-digit sequence number is sufficient for numbering the frames for the SW ARQ protocol (010101...). Similarly, the acknowledgement frames have to be numbered according as they are lost or errored.

To detect transmission errors, each k information bits are encoded into n bits by means of CRC circuits (2). These have detecting code polynomials of 16 or 32 stages (CRC 16, CRC 32):

$$\begin{aligned} G(x) &= 1 + x^2 + x^{15} + x^{16} \quad \text{(CRC-16)} \\ G(x) &= 1 + x + x^2 + x^4 + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{12} \\ &\quad + x^{16} + x^{22} + x^{23} + x^{26} + x^{32} \quad \text{(CRC-32)} \end{aligned} \tag{8}$$

The probability that a frame in error is unchecked by CRC is small and given by (3).

$$P_{ud} \le 2^{-(n-k)}[1 - (1 - P_b)^n] \tag{9}$$

where P_b is the probability of bit transmission error over the channel, and the CRC code rate is

$$r = \frac{k}{n} = \frac{n - \text{CRC overhead}}{n}$$

If CRC checking reveals that the received frame does not contain any errors, then the accepted frame is actually errorless with high probability. However, to take the CRC checking time into consideration the efficiency η in Eqs. (1) to (7) should be multiplied by $r = k/n$; for example, Eq. (4) is replaced by

$$\eta = \frac{T_f}{P(T_o + T_f)/(1-P) + (T_f + 2\tau + T_p + T_A)} \cdot \frac{k}{n} \qquad (10a)$$

The assumption that all frames accepted by CRC are actually correct will be made throughout this article. The occurrence of occasional CRC errors leads to the necessity of defining a reliability measure as the probability of delivery of erroneous frames to higher layers :

$$P(E) = P_{ud} + P_{ud}P + P^2P_{ud} + P^3P_{ud} = P_{ud}\sum_{i=0}^{\infty} P^i = \frac{P_{ud}}{1-P} \qquad (10b)$$

Noting that frame loss is detected with probability 1 (by means of the timeout mechanism), P_{ud} becomes an upper bound on such loss detection (in reality $P_{ud} = 0$). On the other hand, for error detection, the probability is exactly P_{ud} (because of CRC decision errors). Combining the two events, one gets P_{ud} as the upper bound on the probability of making loss or detection errors, thus giving rise to Eq. (010Bb).

The SW ARQ is used among other techniques within the DLC layer. Negative ACK can be used instead of positive ACK, but lost frames can be then confused with errored frames. In most DLC standards (HDLC, SDLC, etc.) both positive and negative ACK are used to maximize the overall transmission efficiencies.

In all, SW ARQ is simple, does not need buffering, and works well in certain actual systems as mentioned above.

Selective Reject Automatic Repeat Request

Selective reject (SR) ARQ is the first of several sliding-window ARQ techniques where the information frames are continuously sent by the transmitter and acknowledged by the receiver (Fig. 2).

In Fig. 2, frame 3 is received in error, so no ACK is sent to frame 3 from the receiver, while frames 0, 1, 2, 4, and 5 are acknowledged. Depending on the application, the latter frames may be left stored in the DLC layer of the receiver and only be delivered to the user (higher layers) after frame 3 is subsequently received. Alternatively, frames 0, 1, 2 may be delivered to higher layers before receiving frame 3. The transmitter times out (ACK to subject frame not received after a time T_o following its transmission) and resends frame 3 following frame 7, then resumes with the transmission of frames 8, 9, and so on. NACK could also be used instead of the timeout mechanism; hence NACK3 will be sent in Fig. 2. Utilization of both NACK and timeout takes care of both errored and lost frames. Reassembly time is wasted in segmentation and reassembly (*SAR*), whereas link capacity is maximized by retransmissions of only lost or errored frames.

Upon each request for retransmission, one frame of length T_f is retransmitted. The total number of transmissions is a random variable with average $N_f = 1/(1-P)$, and the SR efficiency (maximum throughput) is

$$\eta = \frac{k}{n} \cdot \frac{T_f}{N_f T_f} = (1-P)\frac{k}{n} \qquad (11)$$

Sliding-window techniques are typically used with flow control (to stop the sender from flooding the receiver with too many frames). The transmitter can send up to W ($W \le 2^{n-1}$ to avoid retransmission ambiguities, where 2^n is the maximum frame sequence number) consecutive frames without receiving ACK; then it stops

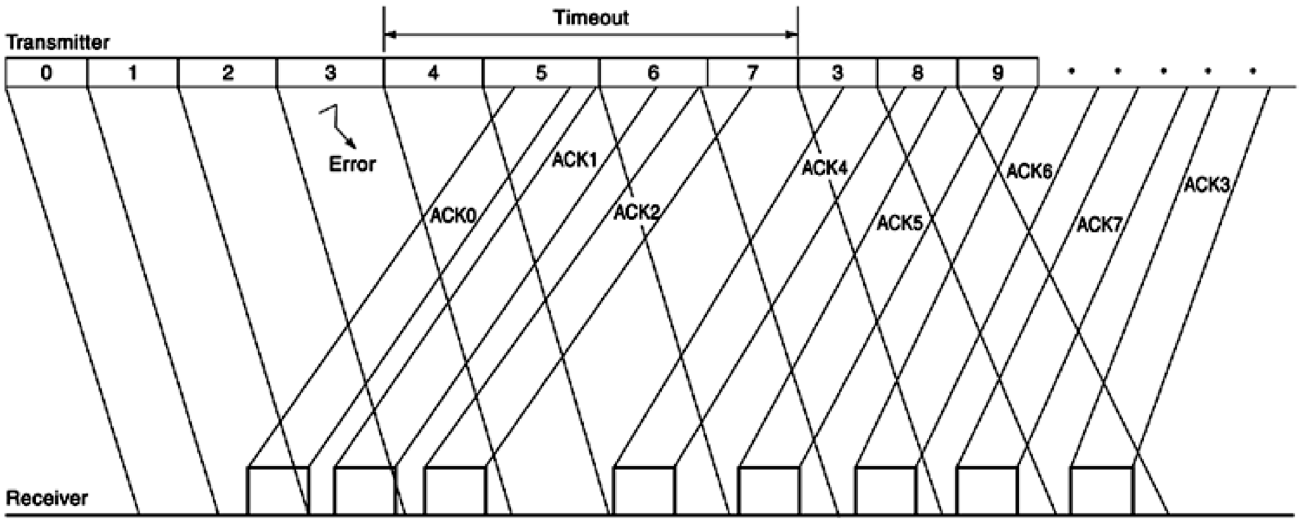

Fig. 2. Lost-information frame recovery using selective reject automatic repeat request. Frame 3 is received in error, and no ACK is transmitted to sender, so sender after timeout retransmits frame 3, after having transmitted frame 7.

till the receiver sends ACK to one or more frames (Fig. 3). In this case the SR maximum throughput (for error- and loss-free links) is given by

$$\eta = \begin{cases} 1 \cdot \dfrac{k}{n} & \text{if } WT_{\mathrm{f}} \geq 2\tau + T_{\mathrm{f}} \quad \text{[Fig. 3(a)]} \\ \dfrac{k}{n} \cdot \dfrac{WT_{\mathrm{f}}}{T_{\mathrm{f}} + 2\tau} & \text{if } WT_{\mathrm{f}} < 2\tau + T_{\mathrm{f}} \quad \text{[Fig. 3(b)]} \end{cases} \tag{12}$$

where processing and ACK times are assumed absorbed into 2τ.

In Fig. 3(a), ACK to the first frame comes before the expiry of the W-frame credit given to the transmitter, and the sender continuously transmits one frame after another, leading to a throughput of $1{\cdot}k/n$. In Fig. 3(b), the window W is small compared to the propagation delay, and the sender stops after transmitting one window (WT_{f}) till the ACK to the first packet comes (after time $T_{\mathrm{f}} + 2\tau$), giving rise to the second case of Eq. (010B).

When channel (link) errors and losses occur, each frame is transmitted a total of N_{f} times (on average) as before, where $N_{\mathrm{f}} = 1/(1 - P)$. Substituting this value into Eq. (010B) and replacing τ/T_{f} by a yields (5)

$$\eta = \begin{cases} (1 - P) \cdot \dfrac{k}{n} & \text{if } W \geq 2a + 1 \\ \dfrac{W(1 - P)}{1 + 2a} \cdot \dfrac{k}{n} & \text{if } W < 2a + 1 \end{cases} \tag{13}$$

The η of SR ARQ is one of the highest possible; however, the reassembly time implies a corresponding slowdown and loss of η at user level (higher layer of network architecture).

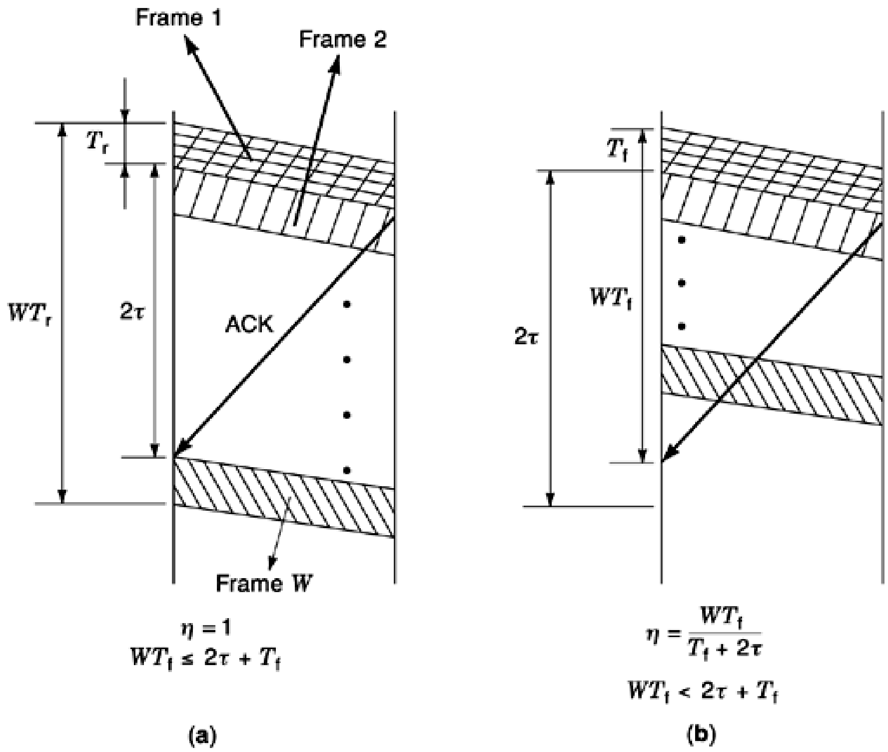

Fig. 3. Sliding-window protocol, error-free link. Continuous transmission is possible when W is large [case (a)]. In case (b), sender stops after transmitting W frames, lowering the efficiency.

Go-Back-*N* Automatic Repeat Request

In the go-back-N (*GBN*) protocol, the receiver insists on receiving frames in sequence. As with SR ARQ, frames are continuously transmitted by the sender and acknowledged by the receiver. However, upon occurrence of error or loss (Fig. 4) a NACK is transmitted to the sender. The sender then backs up N frames and retransmits the whole N-frame window that starts with the frame that was lost or received in error (frame 3 in Fig. 4). This way the frames are delivered in sequence to the user (higher layers) from the receiver, and no reassembly is necessary as in SR ARQ. Typically $N \le 2^n - 1$, where 2^n is the maximum frame sequence number, and N is closely related to the flow-control window credit, namely, $N \le W_s \le 2^n - 1$. Moreover, N is at least the round-trip propagation time plus the processing delay. Frame loss is detected at the receiver by its having received an out-of-sequence frame, whereas an errored frame is detected by CRC, and hence NACK is sent to the sender. Subsequent frames will be ignored by the receiver until the lost or errored frame is received.

Timeout by the transmitter is also necessary in case a NACK or ACK is lost or received in error or if a lost frame is the last one transmitted. Recall that no NACK will be generated if the lost frame is the last, since NACKs are generated by out-of-sequence frames coming after the last frame, which should not happen; so for the last frame ACK and timeout are used.

The maximum throughput of GBN ARQ in the error- and loss-free case is given by k/n for the case with no flow control (stream mode) and by Eq. (010B) for the windowed case. For operation in links with errors and loss, k/n will be replaced by $(k/n)(1/N_f)$ for the stream mode, and the efficiency of Eq. (010B) gets divided by the number of transmissions, N_f.

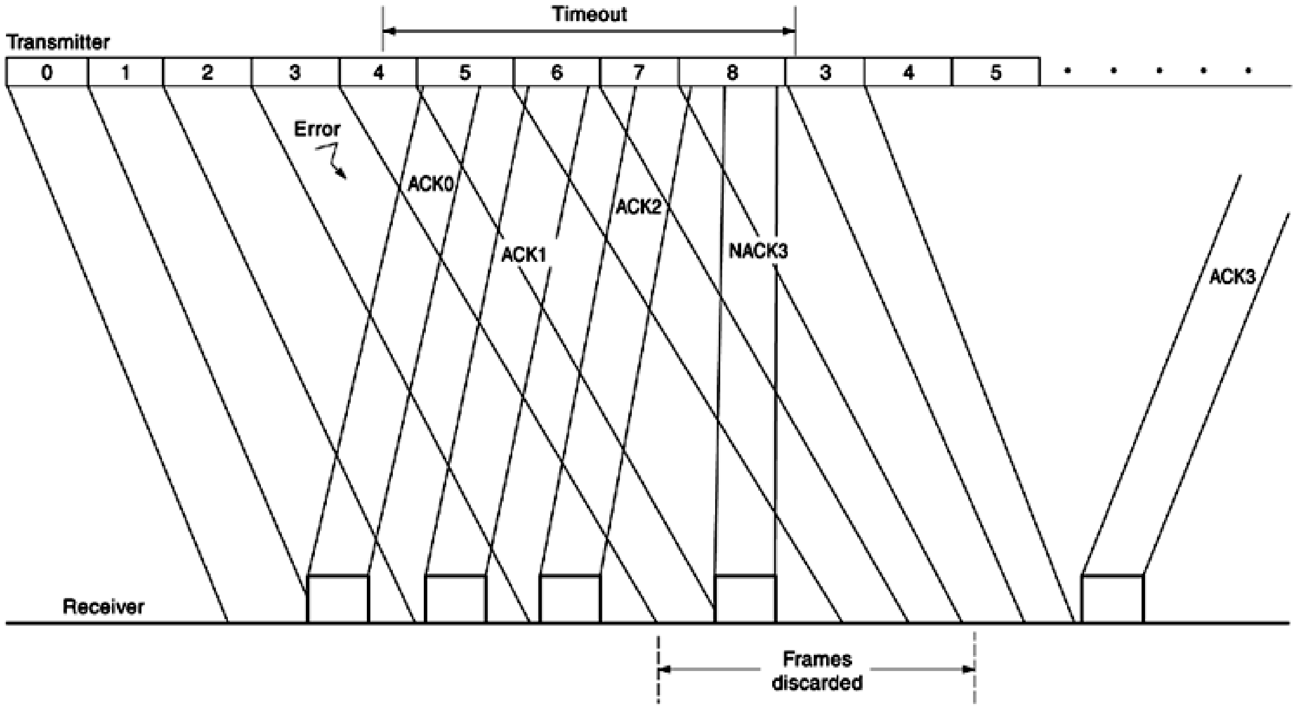

Fig. 4. Recovery in go-back-N ARQ through out-of-sequence frame reception. Frame 3 is received in error, so the sender, upon receiving NACK3, backs up to retransmit frames 3 and then 4, 5, ... in sequence, and the receiver discards all out-of-sequence frames till it receives sequenced frames 3, 4, 5 ....

To evaluate N_f, we note that with probability $1 - P$, the frame is error-free and the time spent is T_f, that is, there is one transmission. With probability $P(1 - P)$, two frames are transmitted: the first is in error and costs N frames; the second is successful. Enumerating over all these cases, one obtains

$$\begin{aligned} N_f &= 1 \cdot (1-P) + (N+1)P(1-P) + (N+2)P^2(1-P) \\ &\quad + (2N+1)P^3(1-P) + \cdots \\ &= \sum_{i=0}^{\infty} (iN+1)P^i(1-P) = 1 + \frac{NP}{1-P} \end{aligned} \tag{14}$$

Dividing k/n by this N_f, one obtains (5)

$$\eta = \frac{1-P}{NP + (1-P)} \cdot \frac{k}{n} \tag{15a}$$

for the stream mode. For the windowed mode we obtain

$$\eta = \frac{1-P}{1+2aP} \cdot \frac{k}{n} \qquad \text{for } W \geq 2a+1 \tag{15b}$$

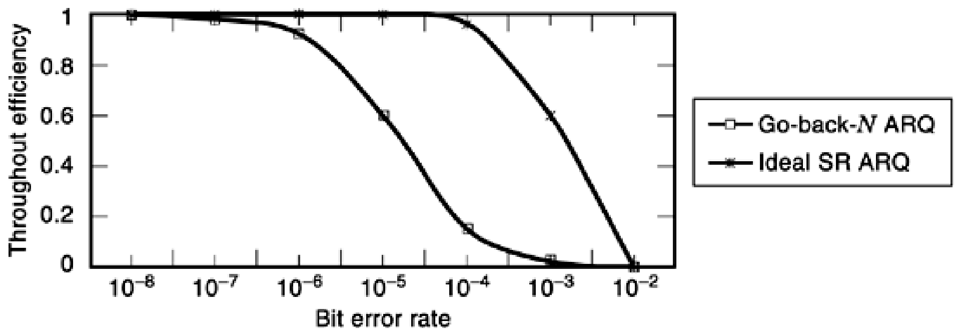

Fig. 5. Comparison of the throughput efficiencies of the ideal SR and the GBN ARQ. At low bit error rates, the throughput performance is the same. For higher rates, SR ARQ yields better efficiency.

where we see that the substitution $N = 1 + 2a$ has been made (recall that for the case $W \geq 2a + 1$, after time 2τ the sender receives an ACK or NACK). Thus $NT_{\rm f} = T_{\rm f} + 2\tau$, leading to $N = 1 + 2a$ and

$$N_{\rm f} = \frac{1 + 2aP}{1 - P} \tag{16}$$

Figure 5 shows a comparison of the efficiencies η provided by SR and GBN ARQ policies as a function of the probability $P_{\rm b}$ of bit errors over the channel. SR provides better efficiency η, but GBN ARQ guarantees in-sequence delivery of frames to users (upper layers).

Hybrid of Forward Error Correction and Automatic Repeat Request: Type I

FEC is used to reduce the frequency of retransmission by trying to correct the most probable random bit errors first (4). Information bits are encoded first at the typical high rate $k/n = (n - 16)/n$ (CRC code C1). The CRC-encoded bits are then fed to an FEC code 2 of rate $r' = n/n'$. At the decoder, the n' bits of the frame are FEC-decoded by the C2 decoder to yield n bits. Error detection on the n bits by CRC code C1 yields the original k data bits. If the CRC detects errors, then the receiver will ask for retransmission of the frame.

In principle SW, GBN, and SR modes will all work with hybrid type 1. All the efficiency expressions in Eqs. () to (21) apply herein, but with

$$P = 1 - (1 - Pb)^n$$

replaced by

$$P' = 1 - \sum_{i=0}^{j=[d_{\min}/2]-1} \binom{j}{i} (Pb)^i (1 - P_{\rm b})^{j-i} \tag{17}$$

because retransmission takes place only if the number of bit errors on the link (channel) is larger than the error correction capability $[d_{\min}/2] - 1$ of the FEC decoder.

Independent errors are assumed here; otherwise, for example, interleaving is used to break the error bursts of the fading channels.

The reliability $P(E)$ of Eq. (010Bb) also will be changed, to

$$P(E) = \frac{P_{\mathrm{ud}}}{1 - P'} \tag{18}$$

The throughput and reliability [η and $P(E)$] of the type I hybrid ARQ have been computed (4) assuming the (24, 12, 8) extended binary Golay code and SR mode.

Figure 6 shows the comparison between the $P(E)$ of pure SR-ARQ [Eq. (010Bb)] and that of the hybrid ARQ type 1 with SR mode [Eq. (015B)]. The x axis is the logarithm of the probability of bit errors, $\log P_{\mathrm{b}}$. Figure 6(b) shows a comparison of η for the two systems. The η of type 1 hybrid ARQ outperforms that of pure ARQ at the expense of less reliability [in Fig. 6(a) we see that the $P(E)$ of hybrid ARQ is higher than that of pure ARQ].

Similar pure-ARQ results for Reed–Solomon (*RS*) FEC codes are shown in (6) for Raleigh faded channels.

Hybrid of Forward Error Correction and Automatic Repeat Request: Type II

One celebrated version of this technique is the Wang–Lin system (7). This uses the usual high-rate CRC error-detecting code C1(n, k) and the systematic invertible FEC code C2(2n, n). The first code, C1, adds the usual 16 or 32 parity check bits of CRC 16 or 32 to the k-bit message to formulate the n-data-bit frame P1, which in turn serves as the message bits of the next FEC code, C2(2n, n). The n parity bits of C2 (i.e., P2) are stored at the beginning of transmitting a frame, and only the n-bit frame of C1 is sent. If no errors are detected by the CRC C1 error-detecting code (ED1), then the corresponding k data bits of C1 are delivered to the user (higher layer). If C1 detects errors, then retransmission takes place. Now only the n-bit parity part of C2 (i.e., P2) is retransmitted, not the whole 2n-bit frame of C2. Because C2 is invertible, it is possible to create the n-bit frame C1 from the P2 parity bits of C2. The inverted word is checked for errors (ED2). If the inverted version has errors, P2 is then appended to P1 to form a new C2 message. FEC decoding is applied again to the 2n-bit word [P1 P2]. The resulting message is checked again for errors (by ED3). If errors still exist, the process continues, with the transmitter alternating transmission between P1 and P2 until one of the three error-detecting schemes reports no errors detected.

The details of analysis of the maximum throughput reliability $P(E)$ of type II hybrid ARQ can be found in (6) and 7. Figure 7 shows that η of type II outperforms η of type I for a good range of bit signal-to-noise ratios ($E_{\mathrm{b}}/N_{\mathrm{o}}$).

Mixed-Mode Automatic Repeat Request

Miller and Lin (8) introduced a technique that alternates between SR ARQ and GBN ARQ. The transmitter starts in SR ARQ and monitors the states (i.e., the numbers of retransmission requests) of frames in the transmitter buffer (Fig. 8). Once the state of one frame (frame 7 in Fig. 8) equals V (i.e., V retransmissions have been requested for that frame by the receiver), the transmitter switches to GBN ARQ mode. The receiver, upon asking for the Vth time for a certain frame, erases $N - 1$ frames that were received after the erroneous subject frame in preparation for the GBN transmission that will follow. The transmitter switches back to Sr-ARQ (frame 11) once the frame (frame 7) that caused the switch to GBN has been successfully acknowledged. If more than one frame in transmitter buffer reaches state V (i.e., has suffered V retransmission attempts in

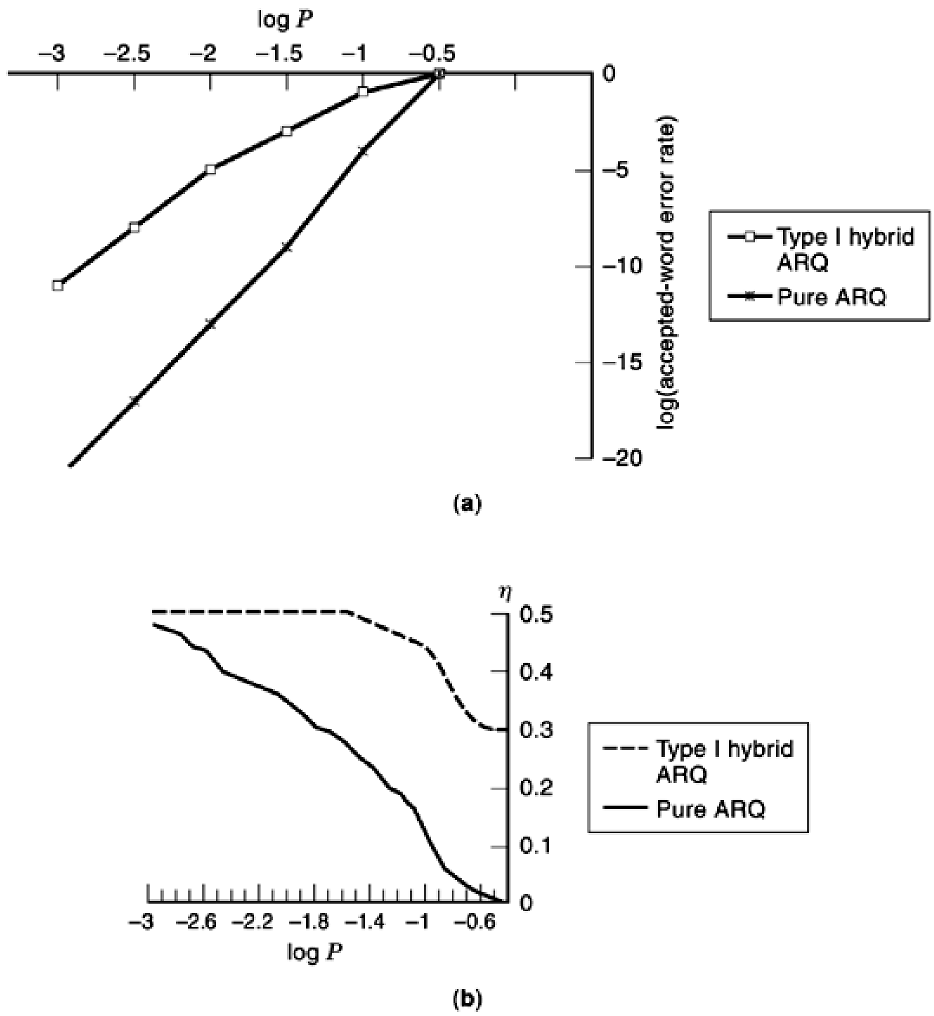

Fig. 6. (a) Reliability comparison of pure ARQ and Type I hybrid ARQ. Pure ARQ has a better accepted-word error rate (5). (b) Performance comparison of pure ARQ and type I hybrid ARQ throughput versus log P. Pure ARQ is inferior to type I hybrid ARQ (5).

SR mode), then corresponding and independent GBN cycles, one for each, will commence, starting with the earliest frame in the buffer, and so on.

The throughput of the above SR + GBN ARQ has been evaluated (8):

$$\begin{aligned} &\text{for} \quad V = 1, \quad \eta_1 = \frac{1-P}{1+(N-1)P^2} \cdot \frac{k}{n} \\ &\text{for} \quad V > 1, \quad \eta_2 = \frac{1-P}{1+(N-1)P^{V+1}} \cdot \frac{k}{n} \end{aligned} \tag{19}$$

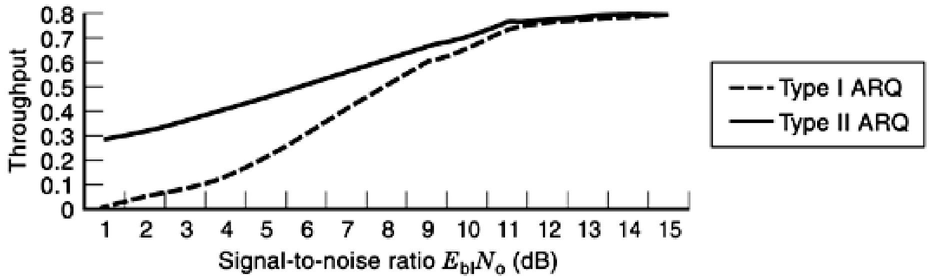

Fig. 7. Throughput performance comparison of type I and type II hybrid ARQ protocols. Type II hybrid ARQ yields higher throughput efficiency (5).

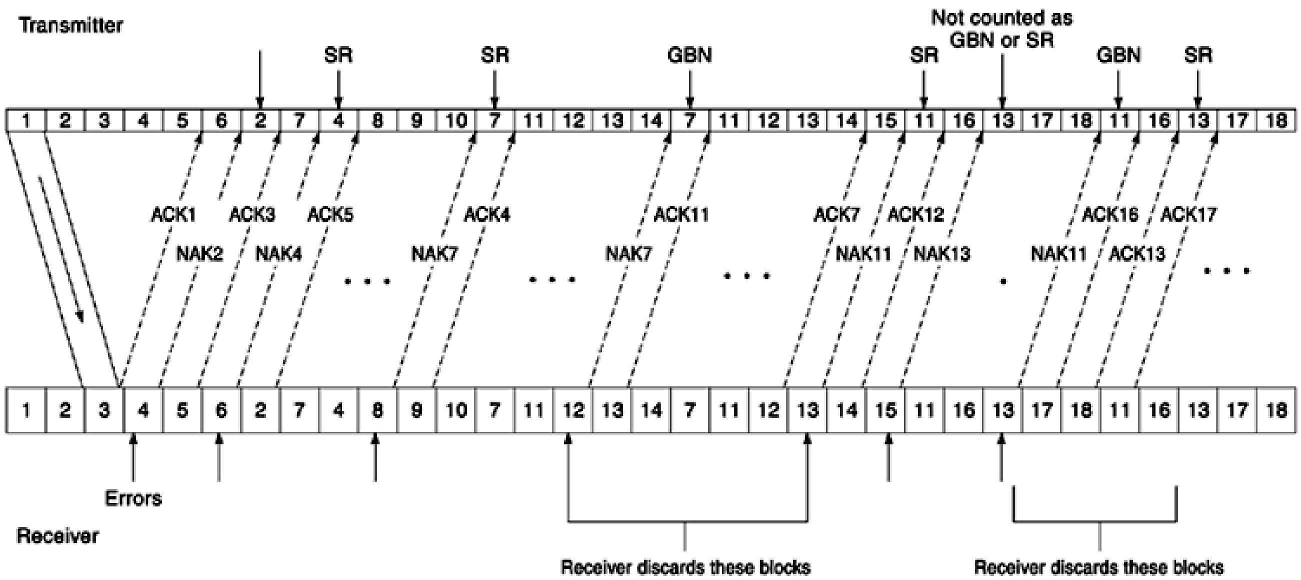

Fig. 8. Timing diagram of the SR + GBN ARQ for $v = 1$ and $N = 5$ (8).

For $V = 0$ we revert to the classic GBN,

$$\eta = \frac{1-P}{NP + (1-P)} \cdot \frac{k}{n} \tag{20}$$

which is the same as Eq. (015Aa).

Figure 9 shows that the SR-GBN η outperforms the η of GBN but is slightly inferior to the η of SR ARQ, which suffers from the reassembly delay problem.

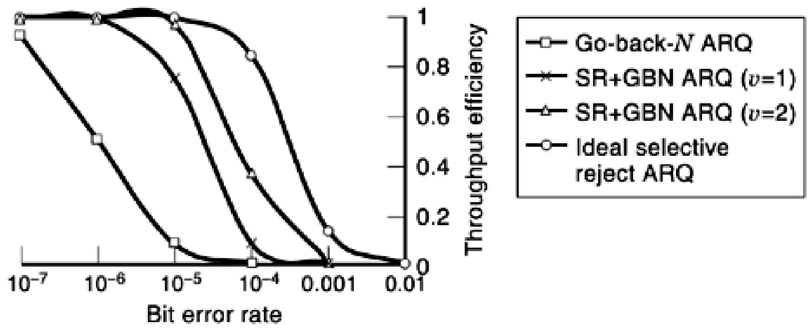

Fig. 9. Comparison of the throughput efficiencies of the SR + GBN with $v = 1$ and $v = 2$, for $n = 2024$ (7). Ideal selective reject yields highest efficiency and GBN gives lowest efficiency, for all bit error rates on the channel.

Automatic Repeat Request Techniques for Quality of Service Multicast Applications Over the Internet

The type I hybrid ARQ scheme is an example of a layered approach where FEC operates separately below ARQ. FEC tries to correct the most probable errors, after which ARQ asks for retransmissions. Better efficiency can be reached by integrating FEC and ARQ in one technique. This integrated approach (9) has recently found theoretical acceptance as a error- and loss-concealment technique in Internet multicast (video, voice, applications, etc.).

Other techniques for error and loss concealment on the Internet use pure ARQ (10), where whole data frames [called transmission groups (TGs) in multicast] are retransmitted. In contrast, in the integrated approach, RS codes are used (9), and only parts of a TG are retransmitted on request.

The RS encoder accepts k data packets and generates parity packets P_1, P_2,..., P_{n-k} (each of l bits). Together the data and parity packets form the frame or TG. RS decoders can recover the whole RS-encoded word (and subsequently the k data and $n - k$ parity packets) if they receive at least any k out the n packets. The RS decoder inserts erasure packets (unknown packets) in place of the lost or errored packets and tries to guess their values in an orderly manner. Once the RS decoder receives or recovers all n packets, it will transform these to the k data packets.

For integrated FEC and ARQ in multicast application, the protocol proceeds as follows:

(1) The sender transmits $k + a$ packets, where $a \leq n - k$, in broadcast mode to all receivers of the sender.
(2) If at least k of those were received, then all other packets of the frame can be estimated by the RS erasure decoding capability and no retransmission will be asked.
(3) If more than $n - k$ packets are lost, the applicable receiver will ask for more packets till it accumulates at least k different data packets.

To evaluate η for this protocol, the discrete distribution of the number of additional packet transmissions required by a random receiver, R_ν, is given by

$$P(R_\nu = 0) = \sum_{i=0}^{a} \binom{k+a}{j} P^j (1-P)^{k+a-j} \tag{21}$$

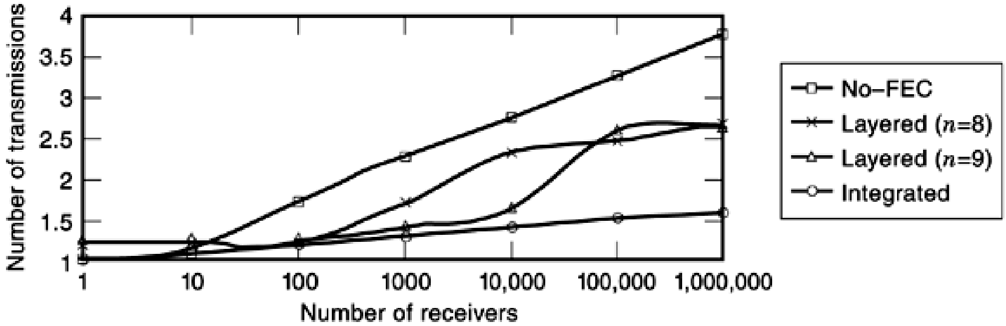

Fig. 10. Comparison of non-FEC, layered FEC ($n = 8, 9$), and integrated FEC ($n = \infty$) for $k = 7$ and $P = 10^{-2}$ (9) versus number of receivers. Non-FEC yields the largest number of transmissions, and integrated ARQ + FEC yields best results.

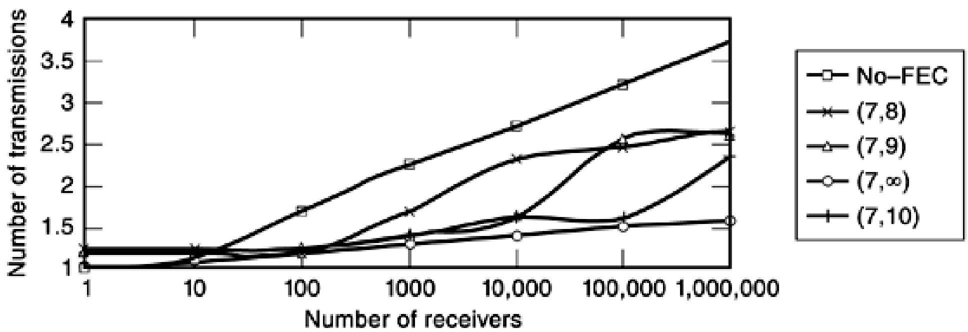

Fig. 11. Integrated FEC with $k = 7$ and $P = 10^{-2}$ for $n = 8, 9, 10, \infty$ (9) versus number of receivers. Non-FEC yields worst results, and $k = 7$ and $n = 10$ yields much better results.

where P is the probability of packet error or loss on the link (channel), and

$$P(R_v = l) = \binom{k+a+l-1}{k-1} P^{l+a}(1-P)^k, \quad l = 1, 2, \ldots \tag{22}$$

Defining L as the maximum number of retransmissions needed by the requests of R receivers, we have

$$P(L \le l) = P(R_v \le l)^R, \quad l = 0, 1, 2, \ldots \tag{23}$$

where

$$P(R_v \le l) = \sum_{j=1}^{l} P(R_v = j) \tag{24}$$

The expected number of retransmissions is

$$E(L) = \sum_{l=0}^{\infty} l \cdot P(L = l) = \sum_{l=0}^{\infty} [1 - P(L \leq l)] \tag{25}$$

and the average total number of packet transmissions per packet is given by

$$E(T) = \frac{E(L) + k + a}{k} \tag{26}$$

Then

$$\eta = \text{maximum throughput} \approx 1/E(T) \tag{27}$$

Figures 10 and 11 show $E(T)$ versus the number of receivers, R, for both the integrated and the layered approach (9). η can be easily evaluated as in Eq. (23) from these results.

BIBLIOGRAPHY

1. M. Schwartz *Telecommunication Networks, Protocols, Modeling and Analysis*, Reading MA: Addison-Wesley, 1987.
2. J. D. Spragins J. L. Hamond K. Pawlinkowski *Telecommunications Protocols and Design*, Reading, MA: Addison-Wesley, 1991.
3. J. B. Anderson S. Mohan *Source and Channel Coding: An Algorithmic Approach*, Norwell, MA: Kluwer, 1991.
4. S. B. Vicker *Error Control Systems for Digital Communications and Storage*, Upper Saddle River, NJ: Prentice-Hall, 1995.
5. T. N. Saadawi M. H. Ammar A. K. Elhakeem *Fundamentals of Telecommunication Networks*, New York: Wiley-Interscience, 1994.
6. S. B. Vicker Reed–Solomon error control coding for data transmission over rayleigh fading channels with feedback, *IEEE Trans. Veh. Technol.*, **41**: 124–133, 1992.
7. S. Lin D. J Costellow *Error Correcting Coding: Fundamentals and Applications*, Englewoood Cliffs, NJ: Prentice-Hall, 1983.
8. M. J. Miller S. Lin The analysis of some selective-repeat-ARQ schemes with finite receiver buffer, *IEEE Trans. Commun.*, **COM-29**: 1307–1315, 1981.
9. J. Nonnenmacher E. W. Biersack D. Towsley Parity-based loss recovery for reliable multicast transmission, *IEEE/ACM Trans. Networking*, **6**: 349–361, 1998.
10. S. Paul K. K. Sabnani J. C. Lin S. Bhattacharyya Reliable multicast transport protocol (RMTP), *IEEE J. Select. Areas Commun.*, **15**: 407–421, 1997.

A. K. ELHAKEEM
Concordia University