Channel Coding 2

Dr.-Ing. Dirk Wübben
Institute for Telecommunications and High-Frequency Techniques
Department of Communications Engineering
Room: N2300, Phone: 0421/218-62385
wuebben@ant.uni-bremen.de

Lecture

Tuesday, 08:30 – 10:00 in N2420

Exercise

Wednesday, 14:00 – 16:00 in N2420

Dates for exercises will be announced during lectures.

Tutor

Matthias Hummert

Room: N2390

Phone 218-62419

hummert@ant.uni-bremen.de

www.ant.uni-bremen.de/courses/cc2/

Outline Channel Coding II

- 1. Concatenated Codes
  - Serial Concatenation
  - Parallel Concatenation (Turbo Codes)
  - Iterative Decoding with Soft-In/Soft-Out decoding algorithms
  - EXIT-Charts
- 2. Trelliscoded Modulation (TCM)
  - Motivation by information theory
  - TCM of Ungerböck, pragmatic approach by Viterbi, Multilevel codes
  - Distance properties and error rate performance
  - Applications (data transmission via modems)
- 3. Adaptive Error Control
  - Automatic Repeat reQuest (ARQ)
  - Performance for perfect and disturbed feedback channel
  - Hybrid FEC/ARQ schemes

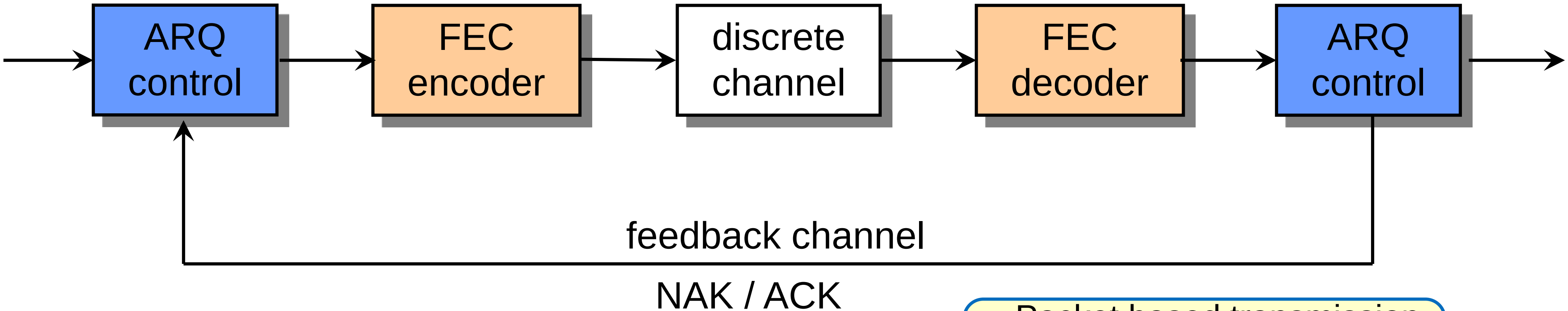
Chapter 3. Adaptive Error Control

- Introduction
- Efficiency and Reliability
- Classical ARQ-Schemes
  - Stop & Wait Strategy (SW)
  - *Go-Back-N* Strategy (GB-*N*)
  - *Selective-Repeat* Strategy (SR)
- Combined ARQ-Schemes
  - *Selective-Repeat* -Strategy with *Go-Back-N*
  - *Selective-Repeat* Strategy with *Stutter* Mode
  - Comparison for Ideal Feedback Channel
- Performance for Real Feedback Channel
  - Model
  - Reliability
  - Throughput of SW, GB-N, and SR
  - Comparison for real feedback channel
- Hybrid FEC/ARQ Systems

Adaptive Error Control

- So far: **FEC** coding (**F**orward **E**rror **C**orrection)
  - Throughput depends on code rate and is independent of channel quality (SNR)
  - Transmission quality (error probability) depends on transmission channel
- Now: **ARQ** strategies (**A**utomatic **R**epeat re**Q**uest)
  - Erroneous packets are repeated → adaptive retransmissions
  - Throughput depends on transmission channel
  - Quality (reliability) is independent of channel quality

ACK: Acknowledgement
NAK: Negative Acknowl.



- Packet based transmission
- Feedback channel
- Error detecting codes

PURE ARQ PROTOCOLS

ARQ Principles and Receiver Design

ARQ with repetition coding (same packet is retransmitted)

- Receiver without memory (**original approach**)
  - Erroneous packets are discarded and retransmission is requested
  - Each retransmission has same error probability for time-invariant channels

- Receiver with memory (**Chase Combining**)
  - Erroneous packet is stored in memory at receiver and retransmission is requested
  - Retransmission is optimally combined with previously received packets
  - SNR increases with each retransmission
  - Error probability is decreasing

ARQ with **Incremental Redundancy** (additional parity bits are transmitted)

  - Receiver with memory required
  - Coding scheme is improved with each retransmission (effective code rate decreases)

Cyclic Redundancy Check (CRC) Codes

- **CRC codes** are cyclic $(2^r\text{-}1,\ 2^r\text{-}r\text{-}2,\ 4)_2$ codes whose generator polynomial has the form $g(D) = (1{+}D)\cdot p(D)$ where $p(D)$ is a primitive polynomial of degree r.

- Decoding by calculation of the syndrome $s(D)$
  - $s(D) \neq 0$: error was detected; $s(D) = 0$: no detectable error or no error

- Properties of cyclic redundancy check codes
  - All error patterns with weight $w_H(\mathbf{e}) = 3$ are detected.
  - All error patterns with odd weight are detected.
  - All burst errors up to a length of $r + 1$ are detected.
  - Only a rate of 2^{-r} of errors with length $r + 2$ cannot be detected.
  - Only a rate of $2^{-(r+1)}$ of errors with length larger than $r + 2$ cannot be detected.

- Example: CRC code with 16 parity bits ($r = n\text{-}k\text{-}1 = 15$) detects
  - 100 % of burst errors with length $\leq$ 16.
  - 99,9969 % of burst errors with length 17.
  - 99,9985 % of burst errors with length $\geq$ 18.

Reliability and Efficiency

- Quality of ARQ schemes are characterized by **reliability** and **efficiency**
- **Reliability** of ARQ schemes with perfect feedback channel
  - ACK / NAK arrive at transmitter without any errors
  - P_c: probability that code word is received **c**orrectly
  - P_{ue}: probability of an **u**ndetected **e**rror
  - P_{ed}: probability of a detectable error (**e**rror **d**etected)
  - P_w: probability that ARQ strategy fails (i.e., transmission error is not detected)

$$P_c + P_{ed} + P_{ue} = 1$$

$$P_w = P_{ue} + P_{ed} \cdot P_{ue} + P_{ed}^2 P_{ue} + \cdots = P_{ue} \cdot \sum_{i=0}^{\infty} P_{ed}^i = \frac{P_{ue}}{1 - P_{ed}}$$

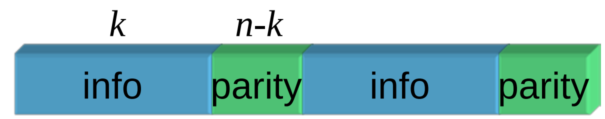
Accepted Packet Error Rate P_w is percentage of packets accepted by the receiver that contain one or more errors.

$$\sum_{i=0}^{\infty} a^i = \frac{1}{1-a}$$

  - Reliability of ARQ depends on capability of error detecting code (i.e., P_{ue} and P_{ed}) and not on channel or ARQ strategy!
  - Example: with $P_{ue} = 0$ we would achieve $P_w = 0$ (**genie code**)

Reliability and Efficiency

- **Efficiency → throughput**



$$R_c = \frac{k}{n}$$

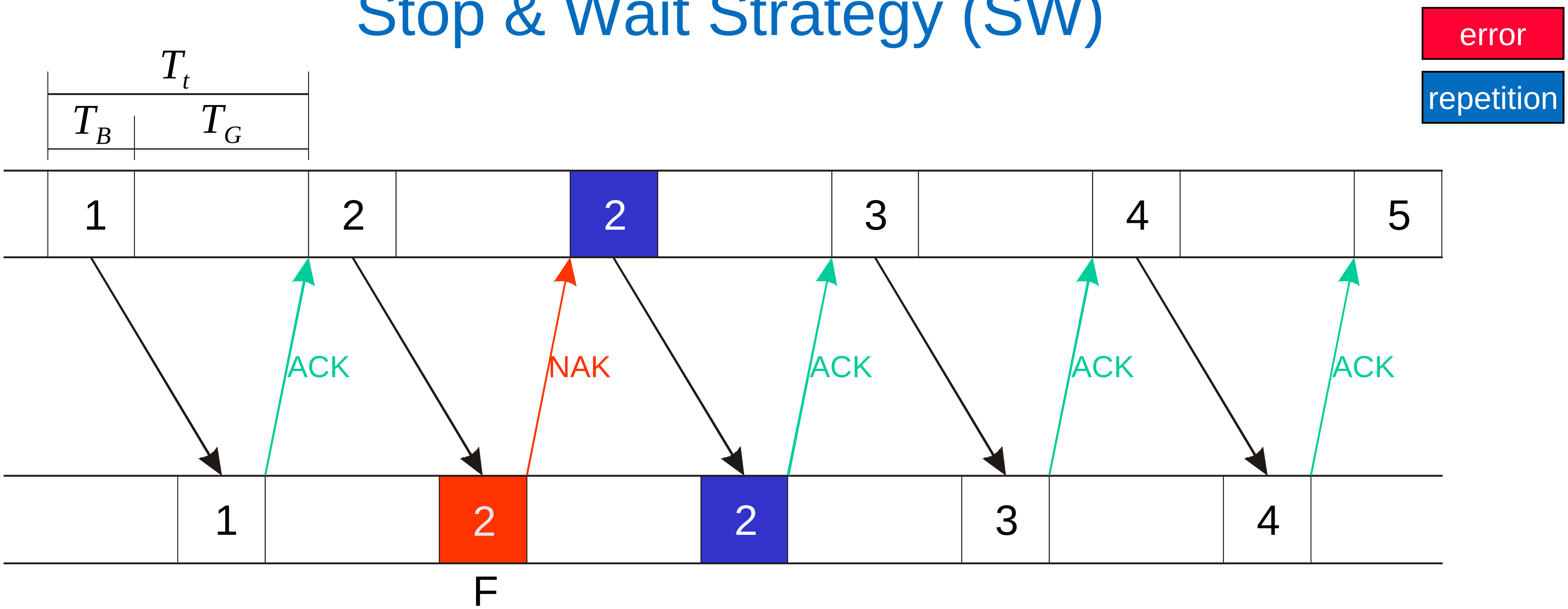
$$\eta = \frac{\text{number of correct transmitted info bits}}{\text{total number of transmitted bits}} = \frac{\text{number of correct transmitted blocks}}{\text{total number of transmitted blocks}} \cdot R_c$$

  - Throughput corresponds to code rate R_c of FEC systems, however varies and adapts to channel condition
  - Efficiency of ARQ is affected by channel properties and ARQ scheme!

For FEC systems the throughput is constant, but the reliability depends on the channel!
ARQ schemes guarantee a constant transmission reliability, but the throughput is affected by the channel condition!

  - For subsequent investigations a **genie code** is considered ($P_{ue}=0$ → $P_w=0$): Efficiency is not influenced by not detected errors as they lead to no retransmissions

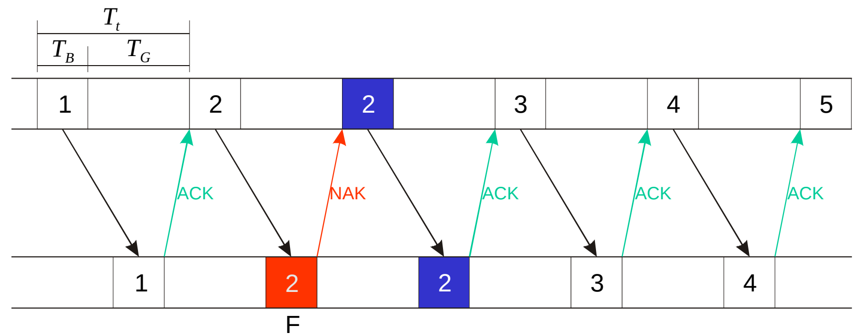
Stop & Wait Strategy (SW)



- After transmitting block of length T_B, the transmitter waits for ACK before the next block is transmitted. In case of NAK, the block is re-transmitted.
- **Idle** time T_G depends on delay of complete transmission system (**round-trip time**)
- Total transmission time per block $T_t = T_B + T_G$
- Advantage: easy implementation, no buffer at receiver necessary
- Drawback: low throughput due to high idle times

Stop & Wait Strategy (SW)

- Throughput for ideal feedback channel
  - Error-free $P_c = (1-P_{ed})$ T_t
  - 1 Repetition $P_c = P_{ed}\cdot(1-P_{ed})$ $2T_t$
  - 2 Repetitions $P_c = P_{ed}^2\cdot(1-P_{ed})$ $3T_t$
  - 3 Repetitions $P_c = P_{ed}^3\cdot(1-P_{ed})$ $4T_t$



- Average transmission time per block

$$T_{AV} = (1-P_{ed})\cdot T_t + P_{ed}(1-P_{ed})\cdot 2T_t + P_{ed}^2(1-P_{ed})\cdot 3T_t + \cdots$$

$$= (1-P_{ed})\cdot T_t \cdot \sum_{i=0}^{\infty}(i+1)\cdot P_{ed}^i = \frac{(1-P_{ed})\cdot T_t}{(1-P_{ed})^2}$$

$$\Rightarrow \quad T_{AV} = \frac{T_t}{1-P_{ed}}$$

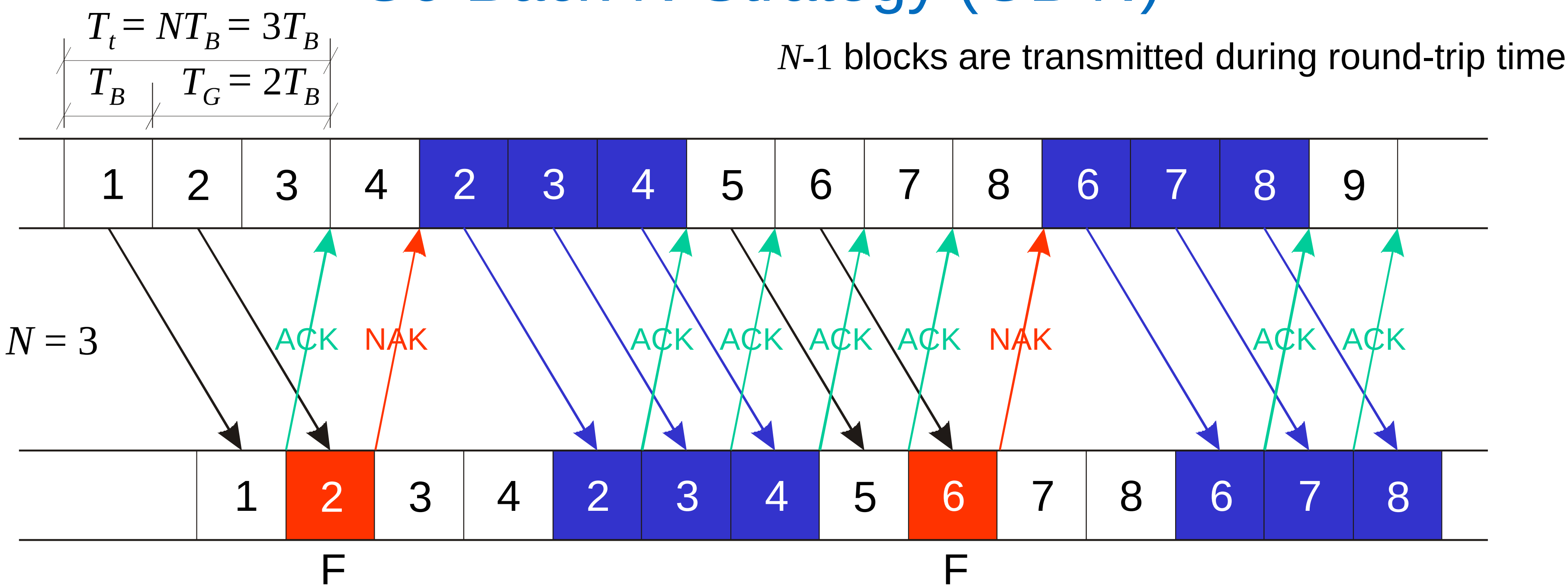
$$\sum_{i=0}^{\infty}(i+1)a^i = \frac{1}{(1-a)^2}$$
for $|a| < 1$

- Throughput: Ratio of duration per block T_B and average transmission time T_{AV} multiplied by code rate R_c

$$\eta_{SW} = \frac{T_B}{T_{AV}}\cdot R_c = \frac{T_B}{T_B+T_G}(1-P_{ed})\cdot R_c = \frac{1-P_{ed}}{1+T_G/T_B}\cdot R_c$$

For $P_{ed} \to 0$ efficiency becomes $\eta_{SW} = R_c/(1+T_G/T_B)$

Go-Back-N Strategy (GB-*N*)



- Continuous transmission of blocks (no idle time)
- In case of an error N blocks (erroneous and correct blocks) are re-transmitted
- Parameter N depends on round-trip time: $N = \lceil T_t / T_B \rceil = \lceil T_G / T_B \rceil + 1$
- Higher throughput than SW, but buffer of N blocks at transmitter is necessary

Go-Back-N Strategy (GB-*N*)

- Throughput for ideal feedback channel
  - Error free $P_c = (1-P_{ed})$ T_B
  - 1 Repetition $P_c = P_{ed}\cdot(1-P_{ed})$ $(N+1)T_B$ (repetition + original) block
  - 2 Repetitions $P_c = P_{ed}^2\cdot(1-P_{ed})$ $(2N+1)T_B$
  - 3 Repetitions $P_c = P_{ed}^3\cdot(1-P_{ed})$ $(3N+1)T_B$
- Average transmission time per block

$$T_{AV} = (1-P_{ed})\cdot T_B + P_{ed}(1-P_{ed})\cdot(N+1)T_B + P_{ed}^2(1-P_{ed})\cdot(2N+1)T_B + \cdots$$

$$= (1-P_{ed})T_B\cdot\sum_{i=0}^{\infty}(i\,N+1)\cdot P_{ed}^i = (1-P_{ed})T_B\cdot\frac{1+(N-1)\cdot P_{ed}}{(1-P_{ed})^2}$$

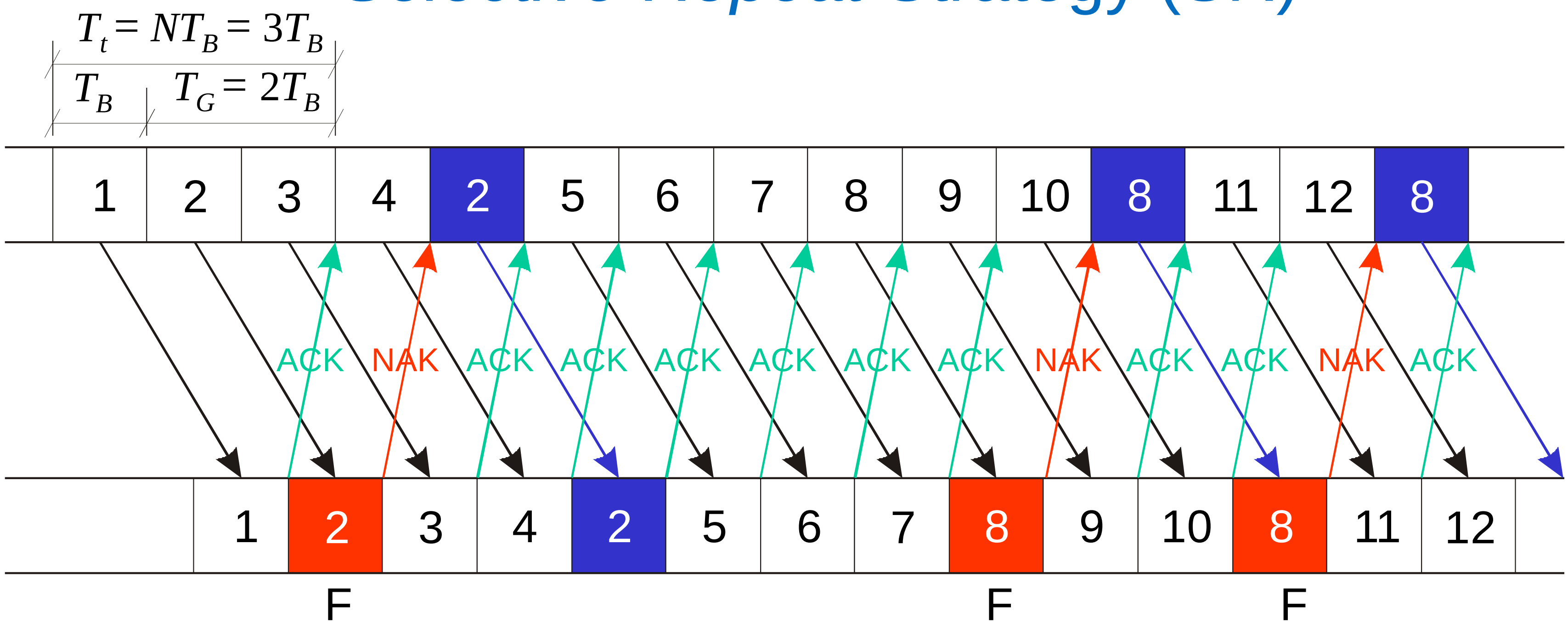
$$\Rightarrow\quad T_{AV} = \frac{1+(N-1)\cdot P_{ed}}{1-P_{ed}}\cdot T_B$$

- Throughput

$$\eta_{GB-N} = \frac{T_B}{T_{AV}}\cdot R_c = \frac{1-P_{ed}}{1+(N-1)\cdot P_{ed}}\cdot R_c$$

For $P_{ed}\to 0$ efficiency tends to code rate $\eta_{GB\text{-}N}\to R_c$

Selective-Repeat Strategy (SR)



- Highest throughput among all presented approaches → only erroneous blocks repeated
- Additional protocol effort: all blocks have to be labeled, to sort them again at the receiver and to retransmit only the erroneous blocks
- Not feasible in practice because an infinitely large buffer is required at receiver:
  - All interim blocks have to be stored in case of an error. For repeated errors, the required memory is duplicated → as memory is limited, buffer overrun (data loss) possible

Selective-Repeat Strategy (SR)

- Throughput for ideal feedback channel
  - Error free $P_c = (1-P_{ed})$ T_B
  - 1 Repetition $P_c = P_{ed} \cdot (1-P_{ed})$ $2T_B$
  - 2 Repetitions $P_c = P_{ed}^2 \cdot (1-P_{ed})$ $3T_B$
  - 3 Repetitions $P_c = P_{ed}^3 \cdot (1-P_{ed})$ $4T_B$
- Average transmission time per block

$$T_{AV} = (1-P_{ed}) \cdot T_B + P_{ed}(1-P_{ed}) \cdot 2T_B + P_{ed}^2(1-P_{ed}) \cdot 3T_B + \cdots$$

$$= (1-P_{ed})T_B \cdot \sum_{i=0}^{\infty}(i+1) \cdot P_{ed}^i = (1-P_{ed})T_B \cdot \frac{1}{(1-P_{ed})^2}$$

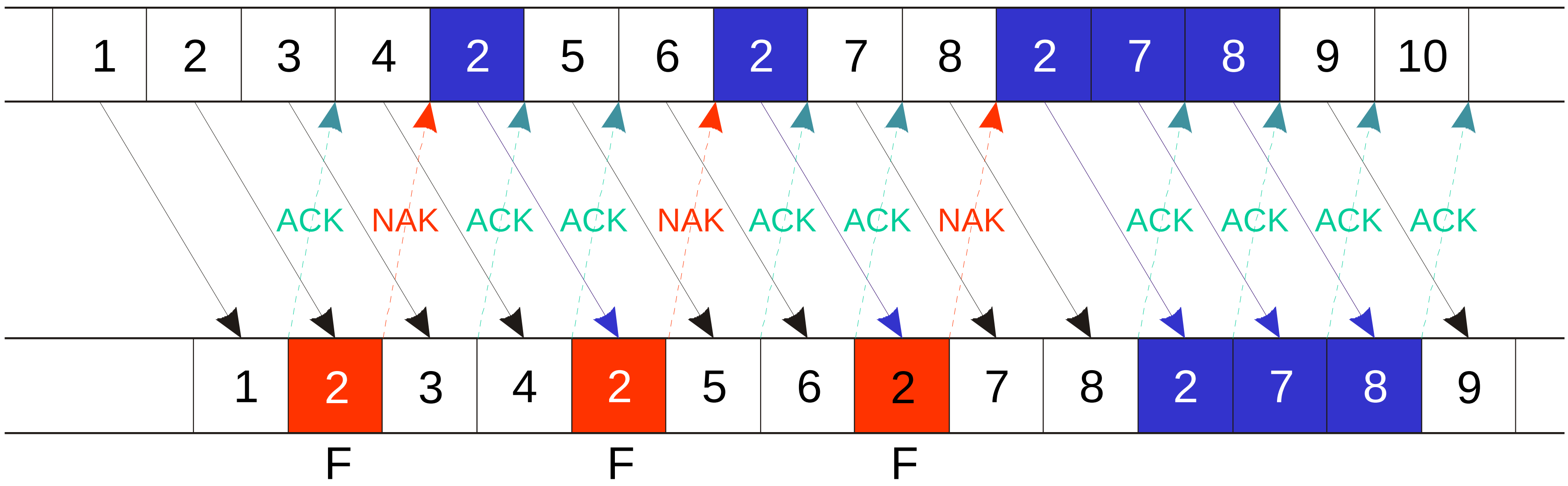
$$\Rightarrow \quad T_{AV} = \frac{T_B}{1-P_{ed}}$$

- Throghput

$$\eta_{SR} = (1-P_{ed}) \cdot R_c$$

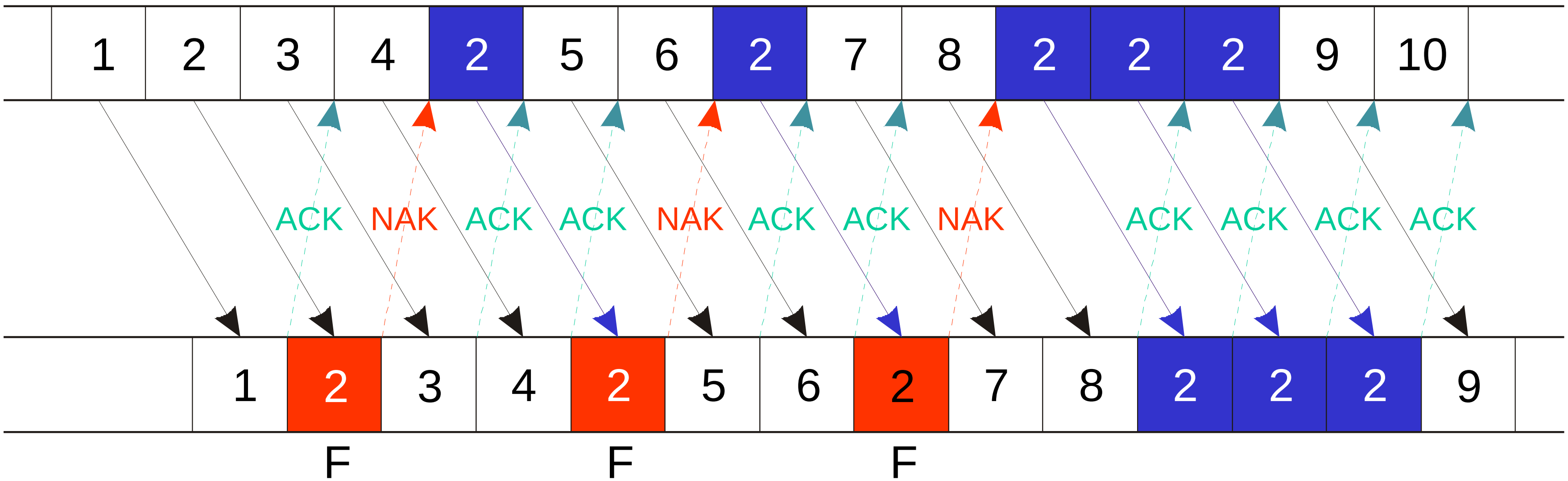
  - For $P_{ed} \rightarrow 0$ efficiency tends to code rate $\eta_{SR} \rightarrow R_c$

Selective-Repeat Strategy with *Go-Back-N*



- Transmission starts in *SR mode*
- After multiple erroneous transmissions switch from *SR mode* to *GB-N mode*
- Only buffer covering last N blocks necessary
- No buffer overrun possible

Selective-Repeat Strategy with *Stutter* Mode



- After multiple erroneous transmissions switch from *SR mode* into *stutter mode* → repeated transmission of erroneous block until correct reception
- No additional buffer necessary due to solely repeating erroneous block
- No buffer overrun possible

Comparison for Ideal Feedback Channel

- Investigation of basic strategies with respect to two exemplary applications, i.e., different distances and thus different *round-trip delays*
- Satellite link
  - Geostationary satellite (orbit is 36.000 km over equator)
  - Overall distance if earth-satellite-earth link is used for forward and reverse link

$$s = 4 \cdot 36.000\ \text{km} = 144.000\ \text{km}$$

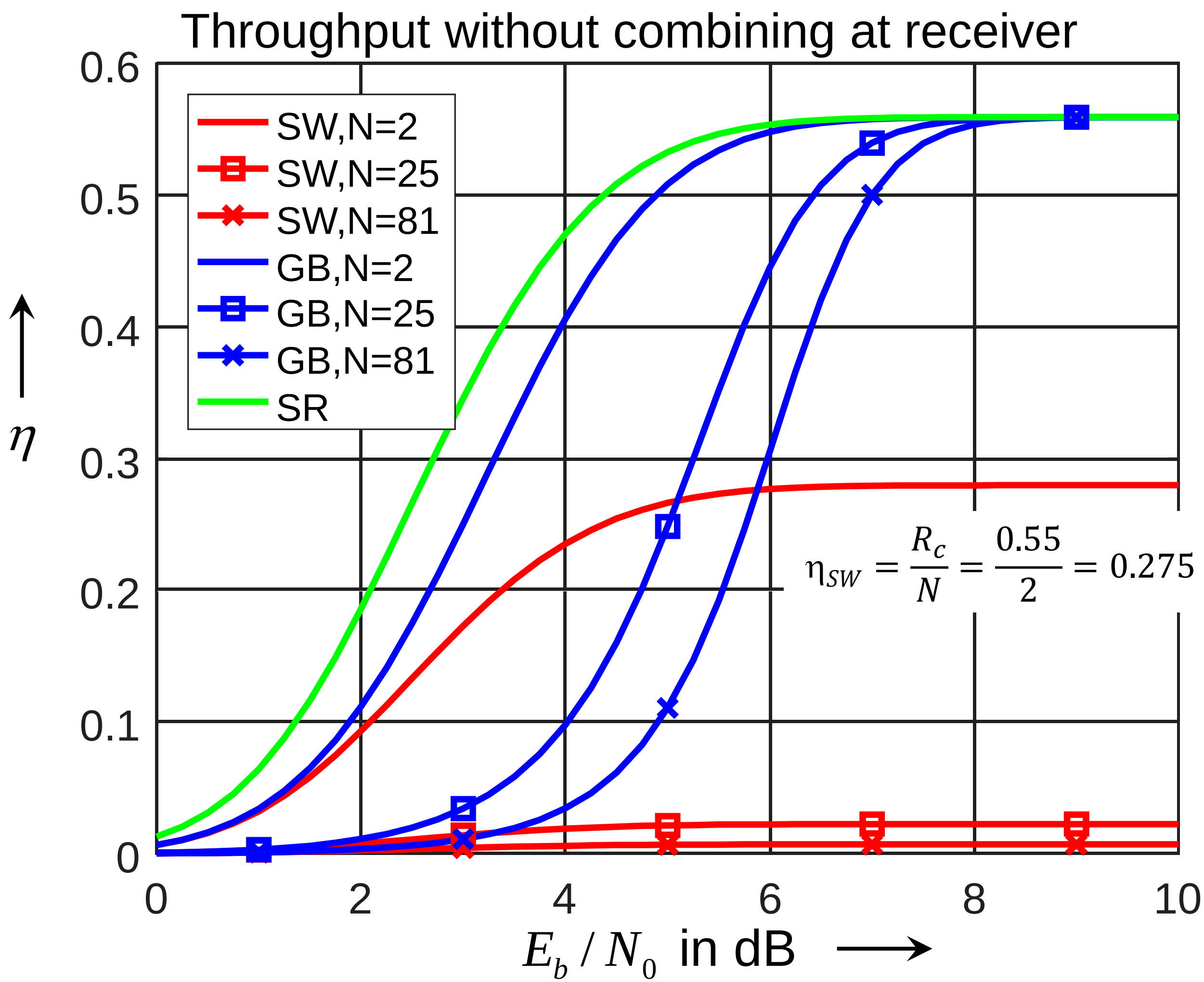
  - Round-Trip time

$$T_G = \frac{s}{c} = \frac{144 \cdot 10^6\ \text{m}}{3 \cdot 10^8\ \text{m/s}} = 0.48\,\text{s}$$

  - Parameter N of GB-N depends on ratio of round trip time T_G and packet length T_B
    - $T_B = 20$ ms (e.g. speech communications) $N = \lceil T_G / T_B \rceil + 1 = \lceil 0.48\,\text{s} / 0.02\,\text{s} \rceil + 1 = 25$
    - $T_B = 6$ ms (short packet) $N = \lceil T_G / T_B \rceil + 1 = \lceil 0.48\,\text{s} / 0.006\,\text{s} \rceil + 1 = 81$
- Beam radio (point-to-point radio system)
  - Almost no delay → $N = 2$

R_c=0.55

Comparison for Ideal Feedback Channel



- E_b/N_0 to P_{ed} mapping

$$P_b = \tfrac{1}{2}\mathrm{erfc}\left(\sqrt{E_b / N_0}\right)$$

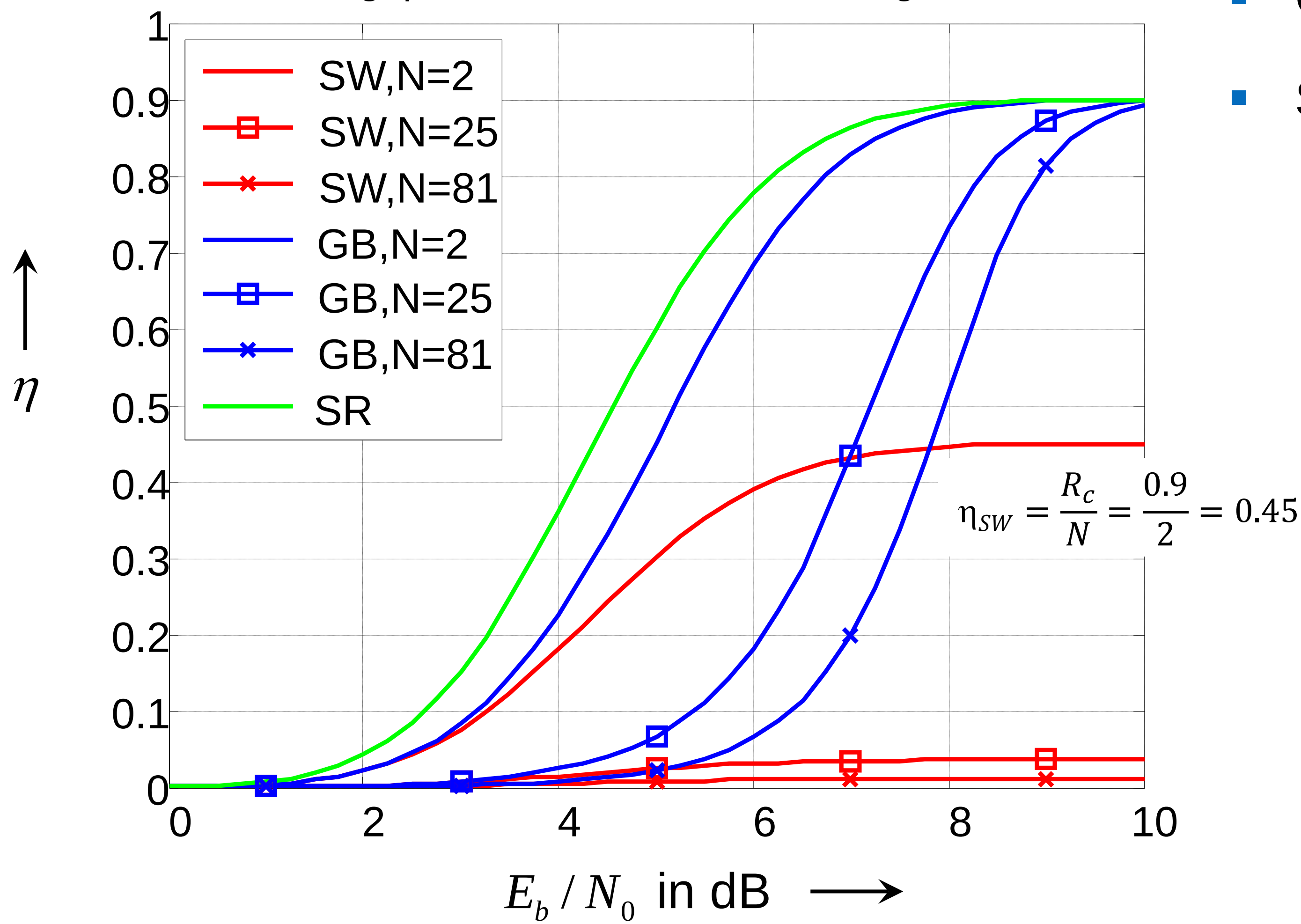
$$P_{ed} = 1-(1-P_b)^n$$

  - (127,71)-BCH → R_c=0.55
- In contrast to *SW* and *GB-N* the throughput of *SR* is independent of system delay
- *GB-N* achieves for N=2 almost SR performance
- Asymptotically SR and *GB-N* approach code rate
- Due to large redundancy, *SW* leads to very low throughput

R_c=0.9

Comparison for Ideal Feedback Channel

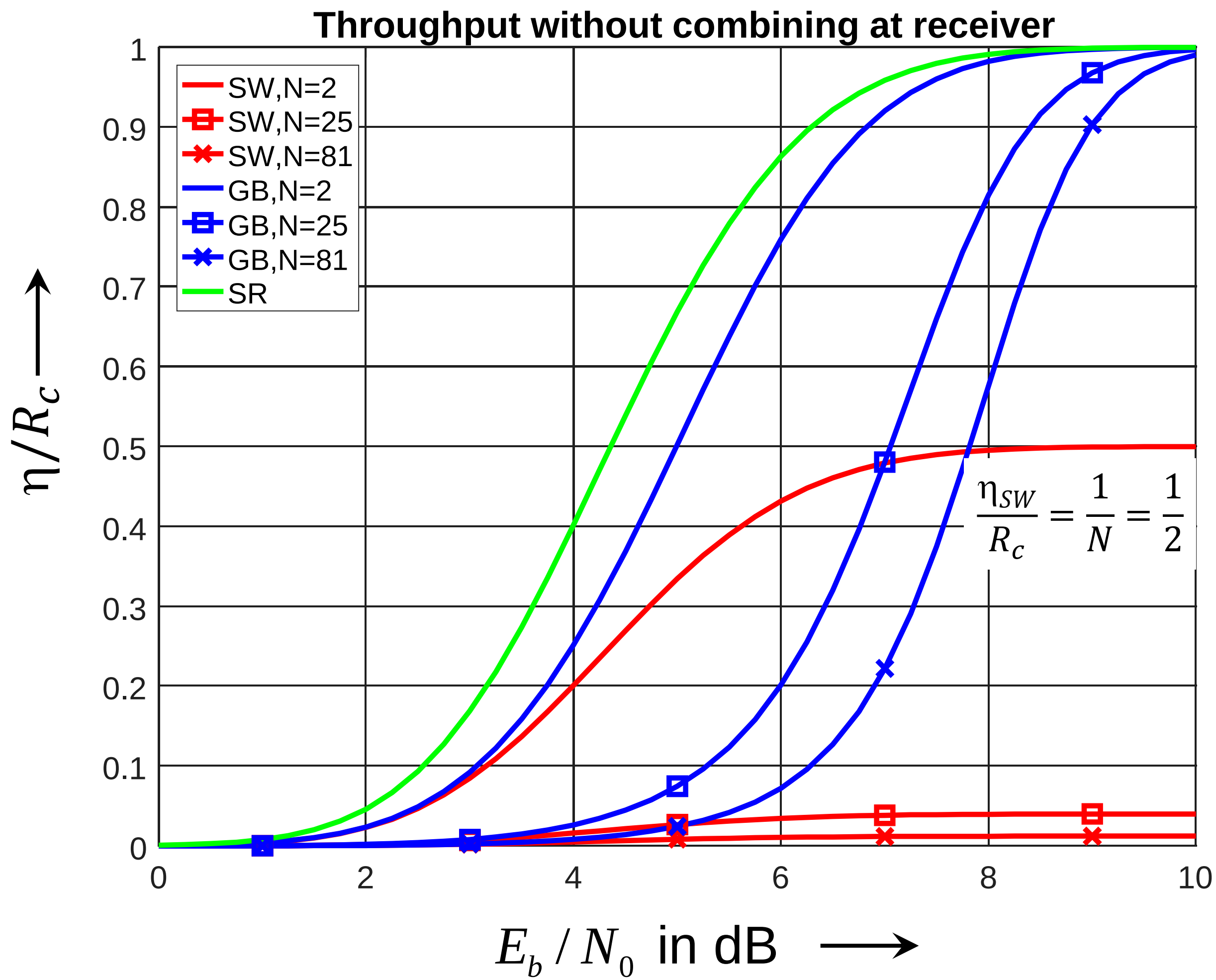
Throughput without Chase combining at receiver



- Code rate $R_c = 0.9$
- Systems:
  - $N = 2$: Beam radio
  - $N = 25$: Satellite link, (T_B = 20 ms)
  - $N = 81$: Satellite link, (T_B = 6 ms)

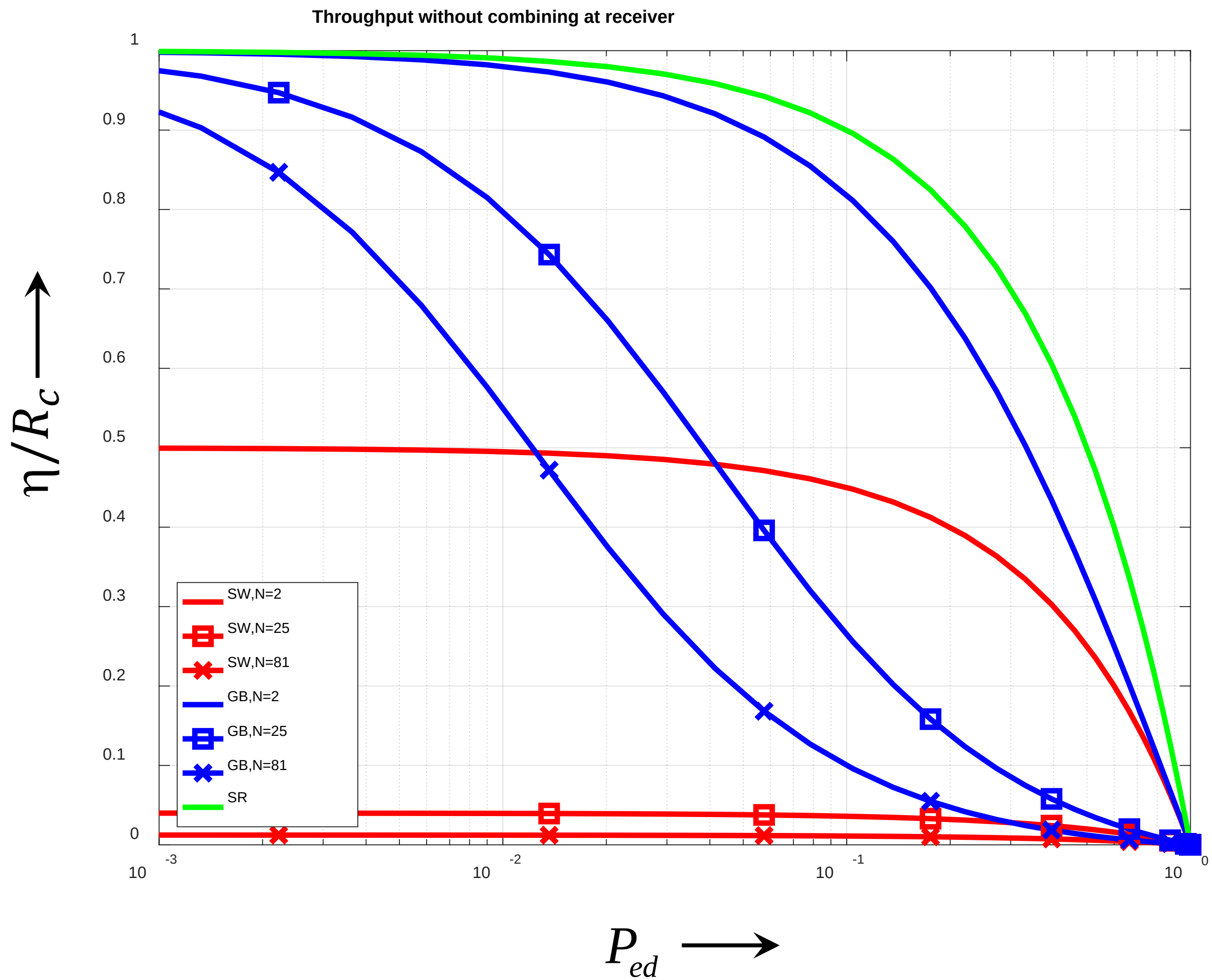
normalized

Comparison for Ideal Feedback Channel



- Normalized efficiency
- Systems:
  - $N = 2$: Beam radio
  - $N = 25$: Satellite link, ($T_B = 20$ ms)
  - $N = 81$: Satellite link, ($T_B = 6$ ms)
- In contrast to *SW* and *GB-N* the throughput of *SR* is independent of syst. delay
- *GB-N* achieves for N=2 almost SR performance
- Asymptotically SR and *GB-N* approach code rate
- Due to large redundancy, *SW* leads to very low throughput

Comparison for Ideal Feedback Channel

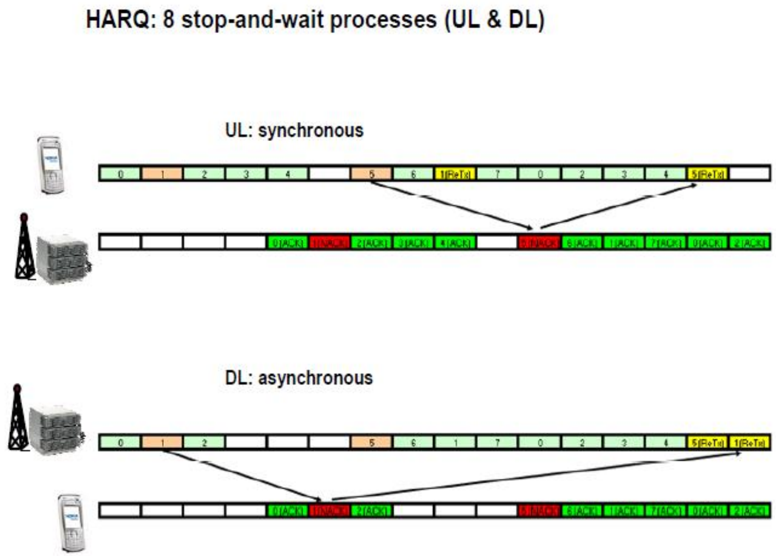


- Systems:
  - $N = 2$: Beam radio
  - $N = 25$: Satellite link, ($T_B = 20$ ms)
  - $N = 81$: Satellite link, ($T_B = 6$ ms)
- Direct impact wrt. P_{ed}

Stop & Wait for High Speed Channels in UMTS Standard

- High Speed Downlink Packet Access (HSDPA) and High Speed Uplink Packet Access (HSUPA) uses Stop & Wait ARQ strategies

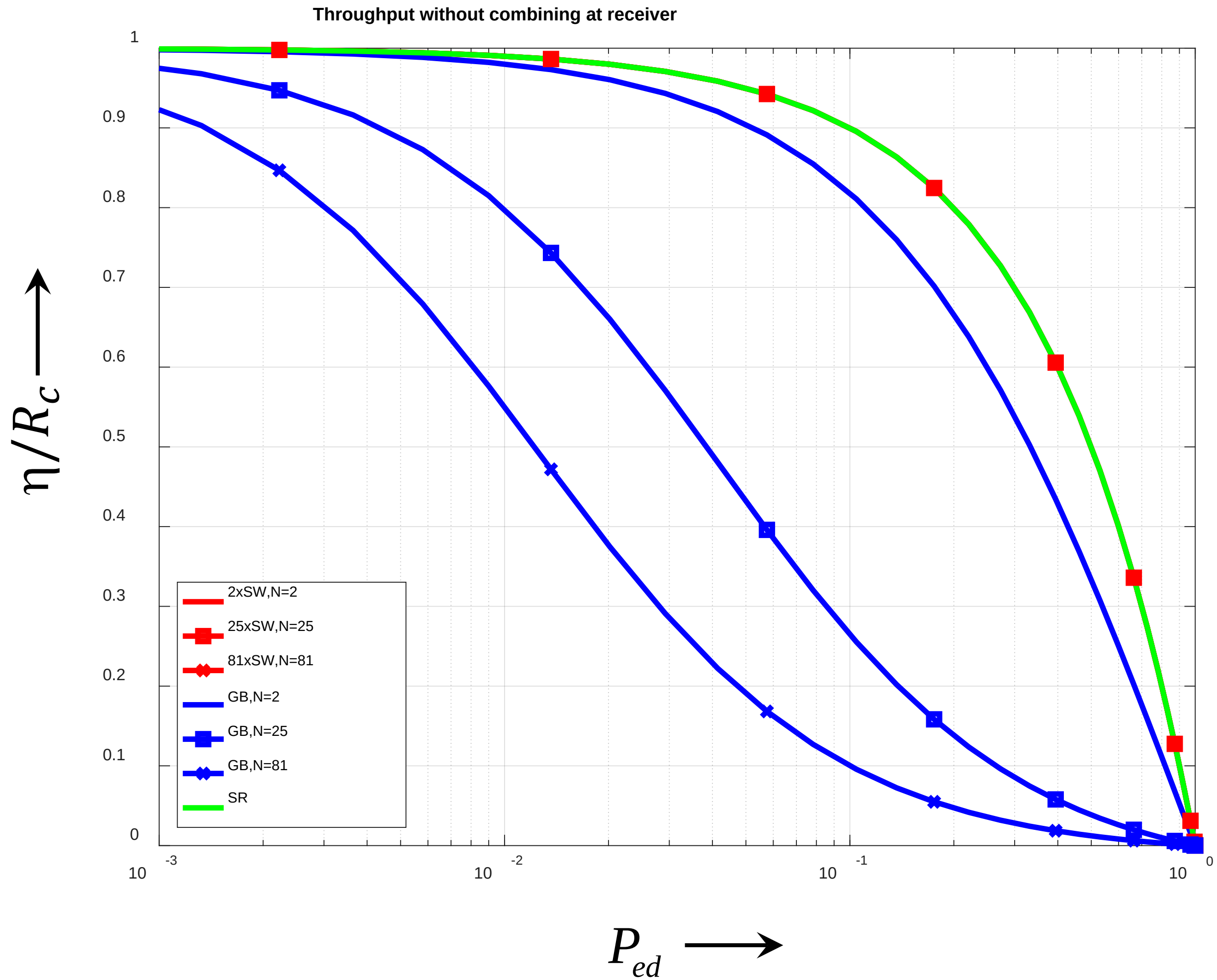
- To overcome S&W penalty, parallel ARQ processes are launched
  → idle time of S&W is used to start further ARQ streams



- Cell phones are categorized according to the number of ARQ streams they can handle

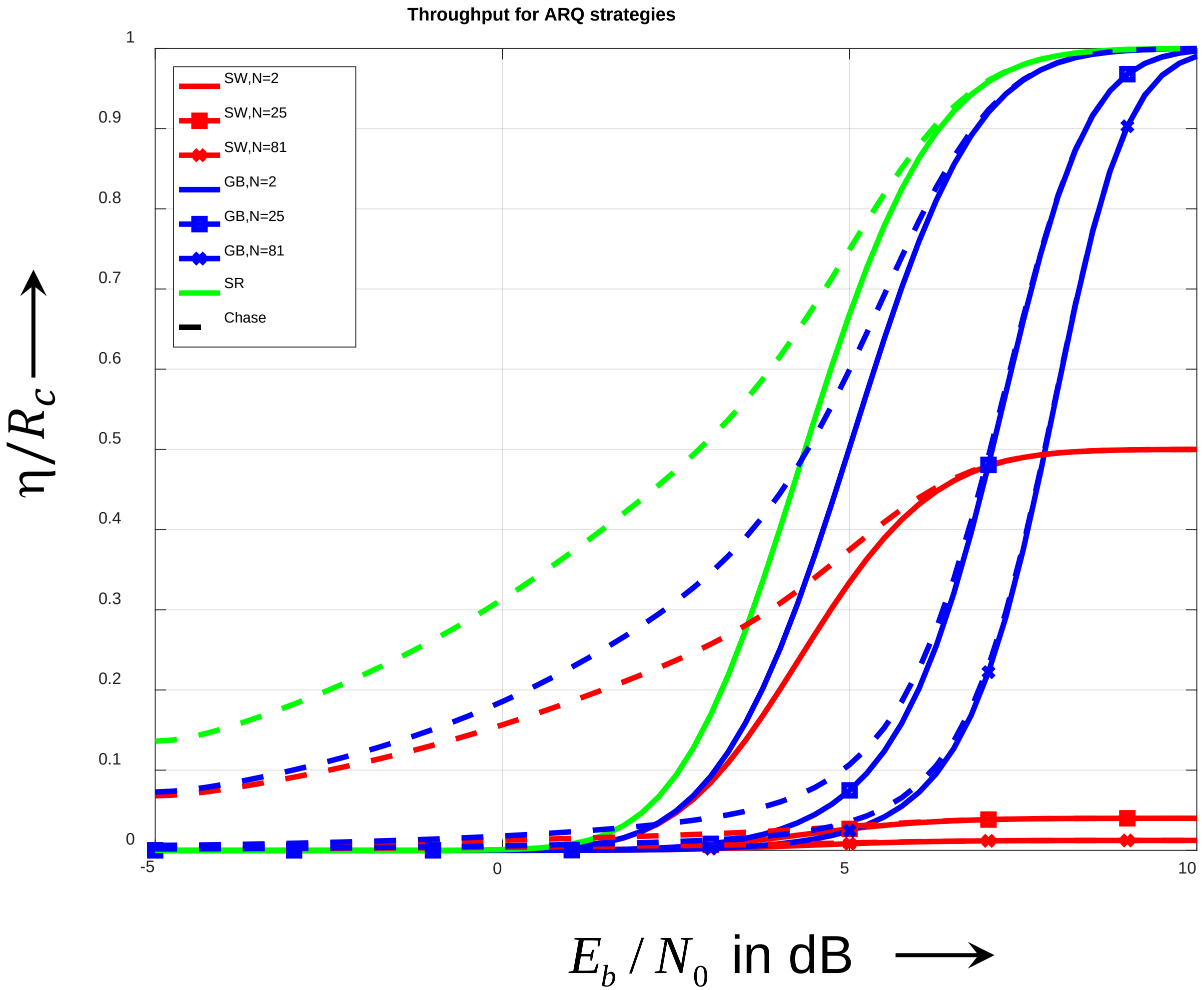
- For $N_{\mathrm{S\&W}}$ ARQ processes, throughput is increased by factor $N_{\mathrm{S\&W}}$

Comparison for Ideal Feedback Channel



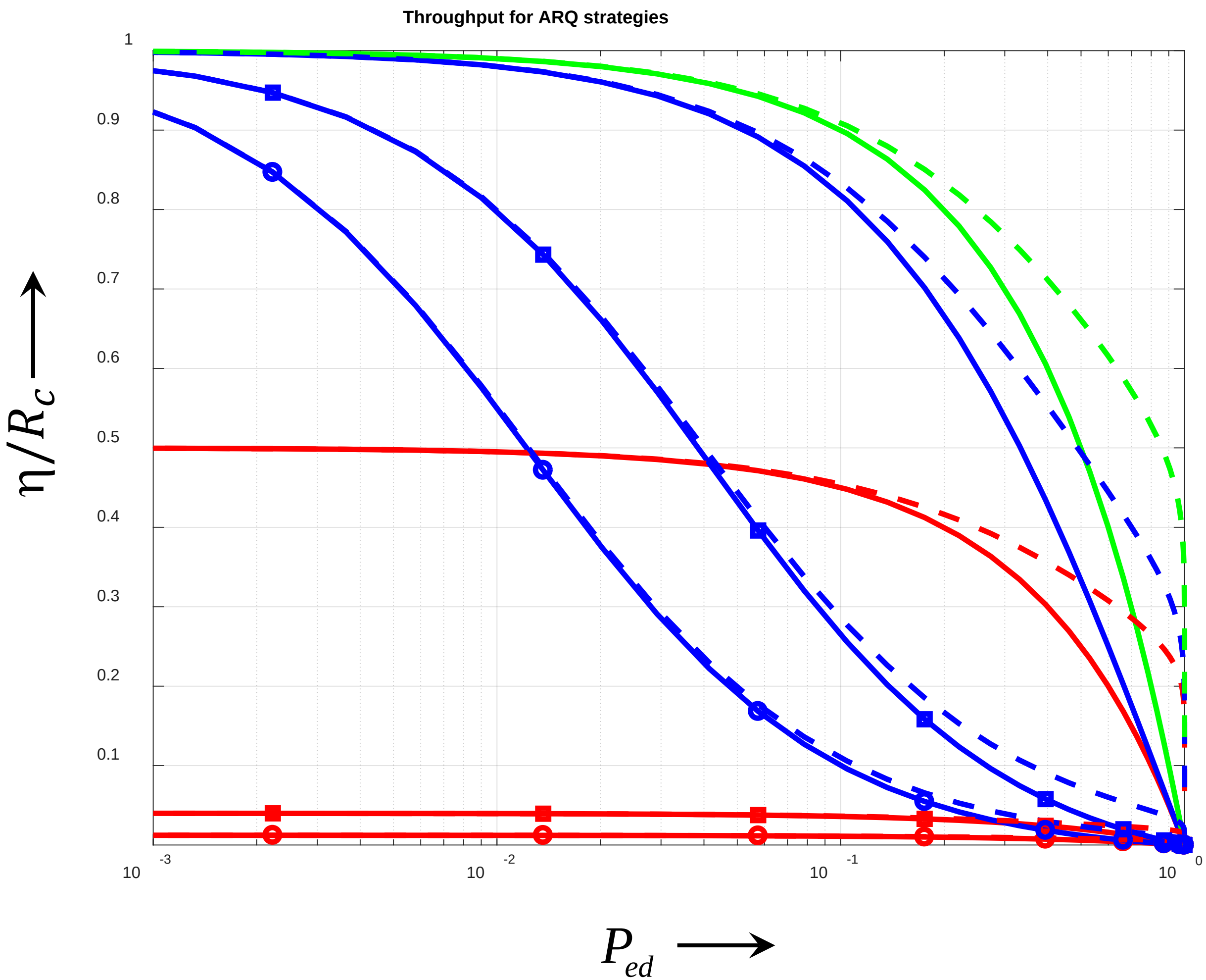
- $N_{\mathrm{S\&W}}$ parallel S&W processes
- Drawback of S&W is completely compensated for $N_{\mathrm{S\&W}} = N$

Comparison with Chase Combining for Ideal Feedback Channel



- Systems:
  - $N = 2$: Beam radio
  - $N = 25$: Satellite link, ($T_B = 20$ ms)
  - $N = 81$: Satellite link, ($T_B = 6$ ms)
- Remarkable gains by Chase Combining for low and medium SNR

Comparison with Chase Combining for Ideal Feedback Channel



- Systems:
  - $N = 2$: Beam radio
  - $N = 25$: Satellite link, ($T_B = 20$ ms)
  - $N = 81$: Satellite link, ($T_B = 6$ ms)
- Remarkable gains by Chase Combining for medium and high P_{ed}

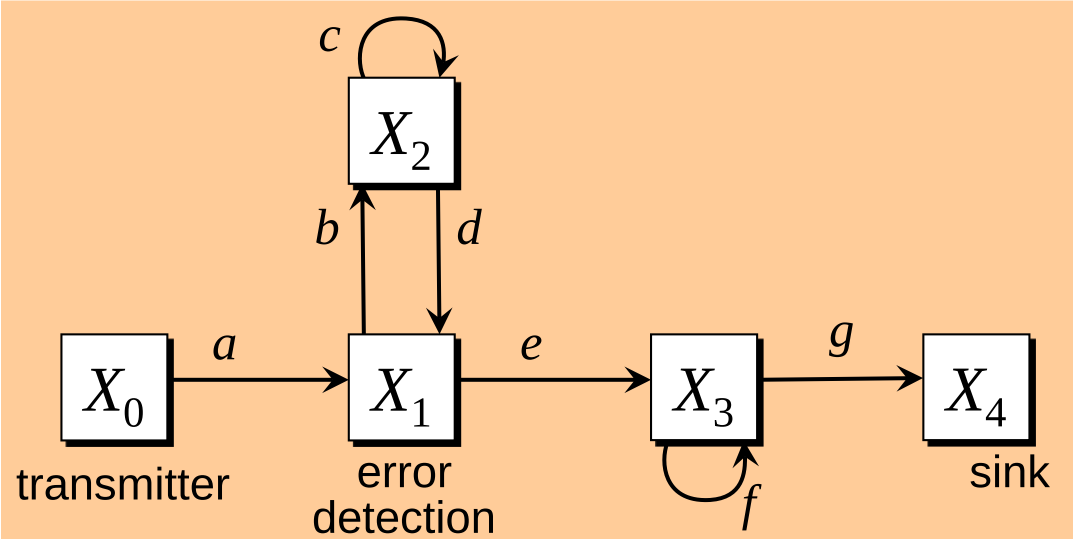
Performance for Real Feedback Channel

- Additional concerns when feedback channel is noisy
  - ACK may become NAK (ACK→NAK)
  - NAK may become ACK (NAK→ACK)
  - Response (ACK/NAK) may not reach its destination (transmitter)

- Additional protocols for noisy feedback channel
  - **Time reference at transmitter**: Each time the transmitter sends out a packet, a timer for that packet is started. If a response from receiver is not obtained for that packet after reasonable period of time, NAK is assumed → new transmission
  - **Time reference at receiver**: When receiver sends NAK, the receiver starts a timer. If new copy of packet is not received after reasonable period of time, NAK is transmitted again
  - If receiver obtains packet that has already been accepted, an ACK is sent to transmitter and packet is discarded

Performance for Real Feedback Channel

- Representation of transmission system as finite state diagram (*Mealy*): considers different events that can happen to a packet during transmission
  - No NAK/ACK signals are lost
- States:
  - X_0: Data is encoded
  - X_1: Error detection
  - X_2: Request of retransmission
  - X_3: Accepting packet
  - X_4: Transmission completed



- Actions (values depend on objective function, i.e., reliability or throughput)

  a: transmission of data block

  b: error detected → NAK to transmitter

  c: NAK → ACK: repeat NAK

  d: repeat block

  e: no error detected

  f: ACK → NAK: error-free block repeated

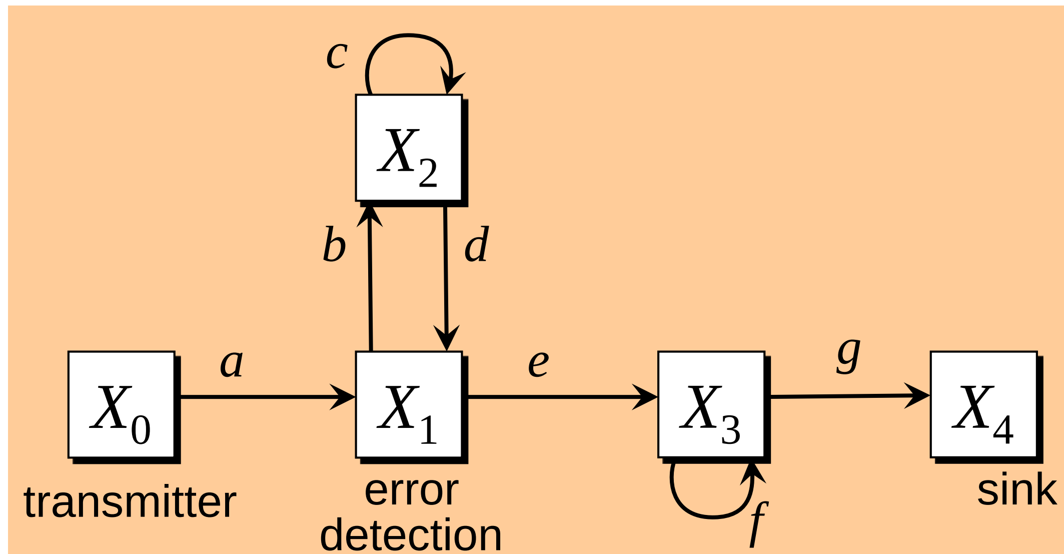
  g: block received at sink

Performance for Real Feedback Channel

- Representation of transmission system as finite state diagram
  - a: transmission of data block
  - b: error detected → NAK to transmitter
  - c: NAK → ACK: repeat NAK
  - d: repeat block
  - e: no error detected
  - f: ACK → NAK: error-free block repeated
  - g: block received at sink

- Linear equation system:

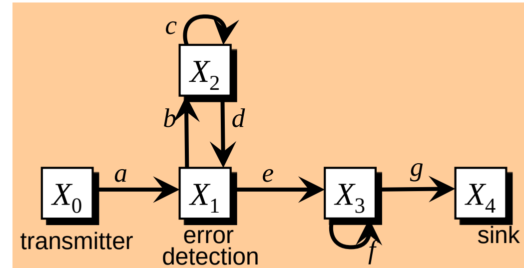
$$X_1 = aX_0 + dX_2$$
$$X_2 = bX_1 + cX_2$$
$$X_3 = eX_1 + fX_3$$
$$X_4 = gX_3$$



- Transfer function:

$$H = \frac{X_4}{X_0} = \frac{aeg(1-c)}{(1-f)(1-c-bd)}$$

Reliability by a Real Feedback Channel



- Parameters

  $a = 1$

  $b = P_{ed}$

  $c = P_{NA}$ Prob. for NAK $\longrightarrow$ ACK

  $d = 1 - P_{NA}$ Prob. for NAK $\longrightarrow$ NAK

  $e = P_{ue}$

  $f = P_{AN}$ Prob. for ACK $\longrightarrow$ NAK

  $g = 1 - P_{AN}$ Prob. For ACK $\longrightarrow$ ACK

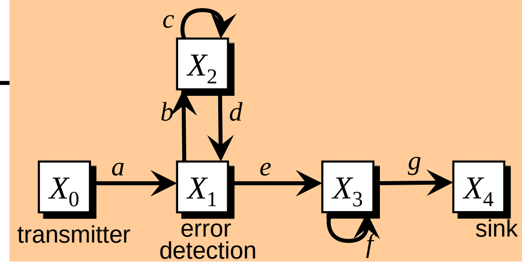
- Transfer function → reliability P_w

$$H = \frac{aeg(1-c)}{(1-f)(1-c-bd)} = \frac{P_{ue}(1-P_{AN})(1-P_{NA})}{(1-P_{AN})(1-P_{NA}-P_{ed}(1-P_{NA}))} = \frac{P_{ue}(1-P_{NA})}{(1-P_{ed})(1-P_{NA})}$$

$$= \frac{P_{ue}}{(1-P_{ed})}$$

  - The real feedback channel has no effect on the reliability, i.e., the error probability of the ARQ system

Throughput for Stop and Wait



- Average time for transmitting a block determines the throughput
  - Time size κ to denote the normalized duration for error-free transmission
  - Placeholder D → exponent denotes the time influence of one state transmission

$$\kappa = \frac{T_B + T_G}{T_B}$$

c: NAK→ACK for block #1: Source transmits block #2, but sink waits for repetition of block #1 and ignores block #2
→ feedback error occupies time of one block transmission

- Parameters

$a = D^{\kappa}$ $\quad d = (1\text{-} P_{NA})\, D^{\kappa}$ $\quad g = 1\text{-} P_{AN}$

$b = P_{ed}$ $\quad e = 1\text{-}P_{ed}$ → genie code

$c = P_{NA}\, D^{\kappa}$ $\quad f = P_{AN}\, D^{\kappa}$

$$H_{SW}(D) = \frac{(1-P_{ed})(1-P_{AN})(1-P_{NA}D^{\kappa})D^{\kappa}}{(1-P_{AN}D^{\kappa})(1-P_{NA}D^{\kappa} - P_{ed}(1-P_{NA})D^{\kappa})}$$

- Average transmission time per packet (info contained in exponent of D)

$$\frac{T_{AV}}{T_B} = \frac{\partial H_{SW}(D)}{\partial D}\bigg|_{D=1} = \frac{\kappa(1-P_{ed}P_{AN} - P_{NA} + P_{ed}P_{NA})}{(1-P_{ed})(1-P_{AN})(1-P_{NA})}$$

$P_{AN} = P_{NA}$: symmetric feedback

$$\eta = \frac{(1-P_{ed})(1-P_{AN})}{(1+T_G/T_B)} \cdot R_c = (1-P_{AN})\eta_{SW}$$

- Efficiency

$$\eta = \frac{T_B}{T_{AV}} \cdot R_c = \frac{(1-P_{ed})(1-P_{AN})(1-P_{NA})}{(1+T_G/T_B)(1-P_{ed}P_{AN} - P_{NA} + P_{ed}P_{NA})} \cdot R_c$$

$P_{AN} = P_{NA} = 0$: perfect feedback

$$\eta = \frac{1-P_{ed}}{(1+T_G/T_B)} \cdot R_c = \eta_{SW}$$

Throughput for GB-*N*

- Parameters: no idle time → normalized transmission time $\kappa = 1$

$a = D$ $d = (1-P_{NA})\,D^N$ $g = 1-P_{AN}$

$b = P_{ed}$ $e = 1-P_{ed}$

$c = P_{NA}D$ $f = P_{AN}D^N$

$$H_{GB-N}(D) = \frac{(1-P_{ed})(1-P_{AN})(1-P_{NA}D)D}{(1-P_{AN}D^N)(1-P_{NA}D-P_{ed}(1-P_{NA})D^N)}$$

- Average transmission time per packet

$$\frac{T_{AV}}{T_B} = \frac{1-P_{NA}-P_{ed}P_{AN}+P_{ed}P_{NA}+(N-1)(P_{AN}+P_{ed}-P_{AN}P_{NA}-P_{ed}P_{NA}-2P_{ed}P_{AN}+2P_{ed}P_{AN}P_{NA})}{(1-P_{ed})(1-P_{AN})(1-P_{NA})}$$

- Efficiency

$$\eta = \frac{(1-P_{ed})(1-P_{AN})(1-P_{NA})}{1-P_{NA}-P_{ed}P_{AN}+P_{ed}P_{NA}+(N-1)(P_{AN}+P_{ed}-P_{AN}P_{NA}-P_{ed}P_{NA}-2P_{ed}P_{AN}+2P_{ed}P_{AN}P_{NA}}\cdot R_c$$

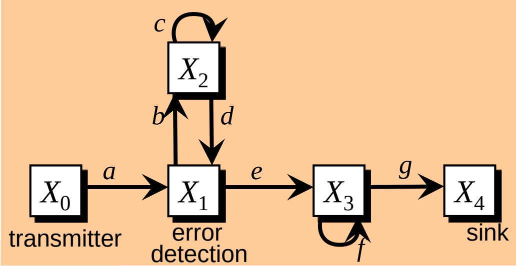
  - Special cases:

$P_{AN} = P_{NA}$:

$$\eta = \frac{(1-P_{ed})(1-P_{AN})}{1+(N-1)(P_{AN}+P_{ed}-2P_{AN}P_{ed})}\cdot R_c$$

$P_{AN} = P_{NA} = 0$:

$$\eta = \frac{1-P_{ed}}{1+(N-1)P_{ed}}\cdot R_c = \eta_{GB-N}$$



Throughput for Selective Repeat

- Parameters

  $a = D$ $\quad d = (1-P_{NA})\,D$ $\quad g = 1-P_{AN}$

  $b = P_{ed}$ $\quad e = 1-P_{ed}$

  $c = P_{NA}$ $\quad f = P_{AN}\,D$

$$H_{SR}(D) = \frac{(1-P_{ed})(1-P_{AN})(1-P_{NA}D)D}{(1-P_{AN}D)(1-P_{NA}-P_{ed}(1-P_{NA})D)}$$

- Average transmission time per packet

$$\frac{T_{AV}}{T_B} = \frac{1-P_{ed}P_{AN}}{(1-P_{ed})(1-P_{AN})}$$

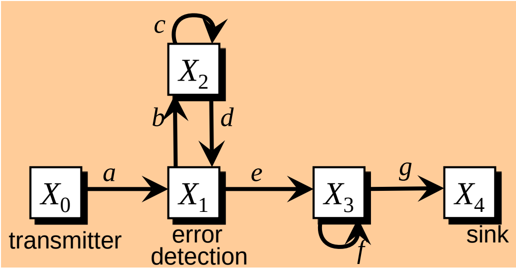
- Efficiency

$$\eta = \frac{(1-P_{ed})(1-P_{AN})}{1-P_{ed}P_{AN}} \cdot R_c$$

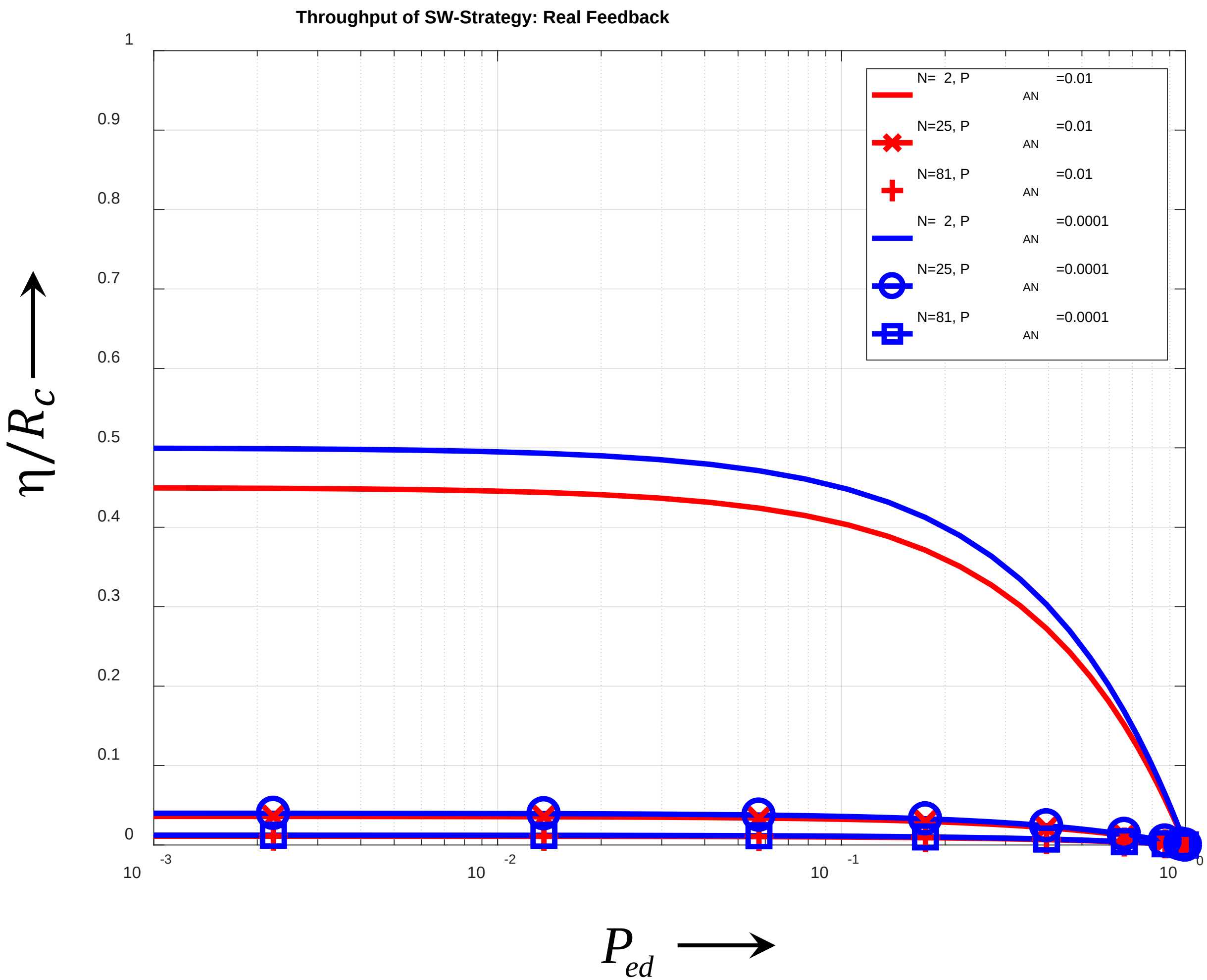
  - Efficiency is independent of P_{NA} (NAK→ACK)

Difference w.r.t. GB-N:
If NAK→ACK occurs in case of an error, the transmission is erroneously continued without repetition. This does not lead to an additional delay, as no packages that are already received correctly are discarded (*c*). Only the affected package is repeated (*d*).

The erroneous repetition of an already correctly received block results in a delay of one only (*f*).

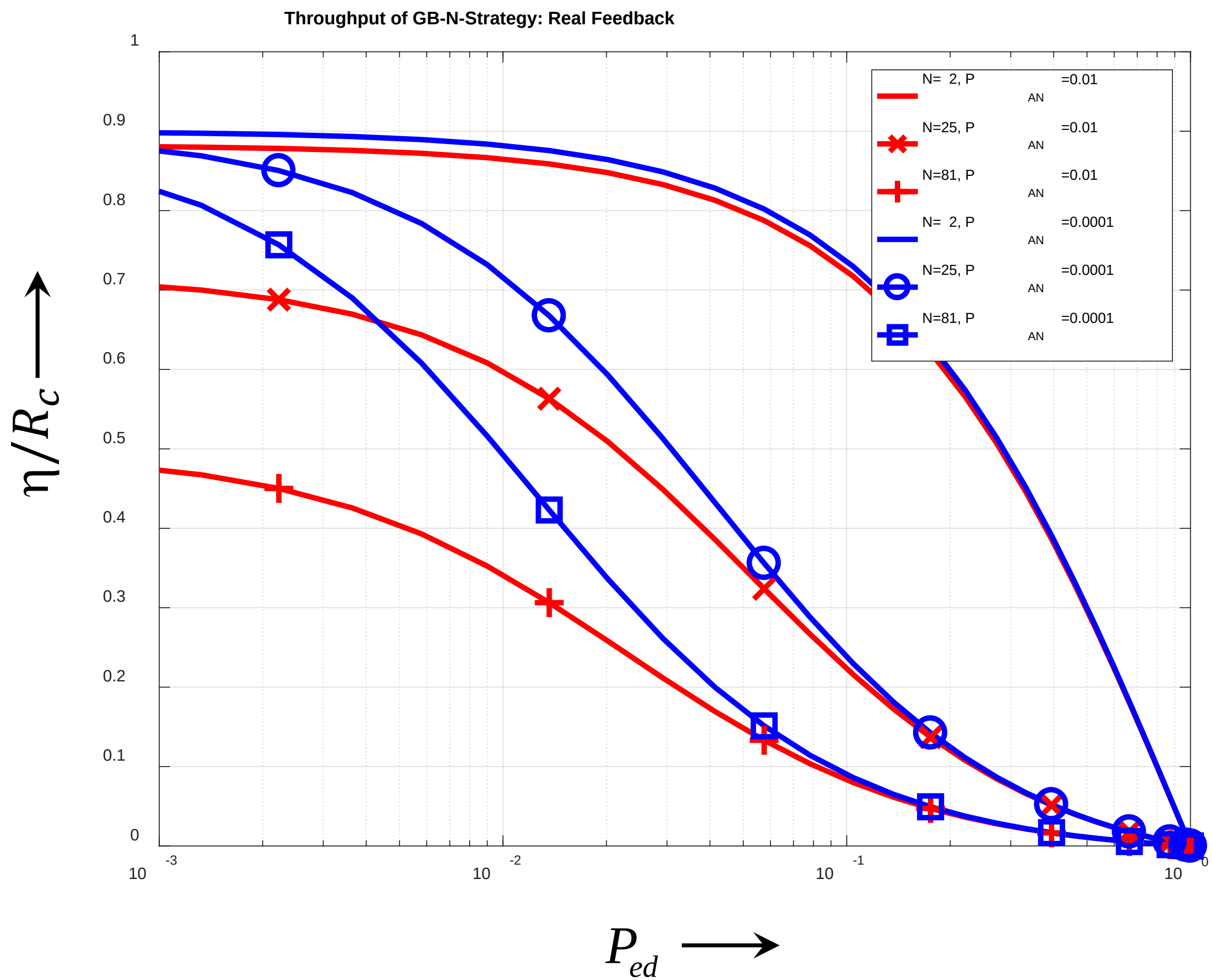


Comparison for Real Feedback Channel (1)



- Systems:
  - Systems:
  - $N = 2$: Beam radio
  - $N = 25$: Satellite link, ($T_B = 20$ ms)
  - $N = 81$: Satellite link, ($T_B = 6$ ms)
- Results
  - Only small efficiency with SW
  - Only small differences for large idle time (round-trip delay / N)

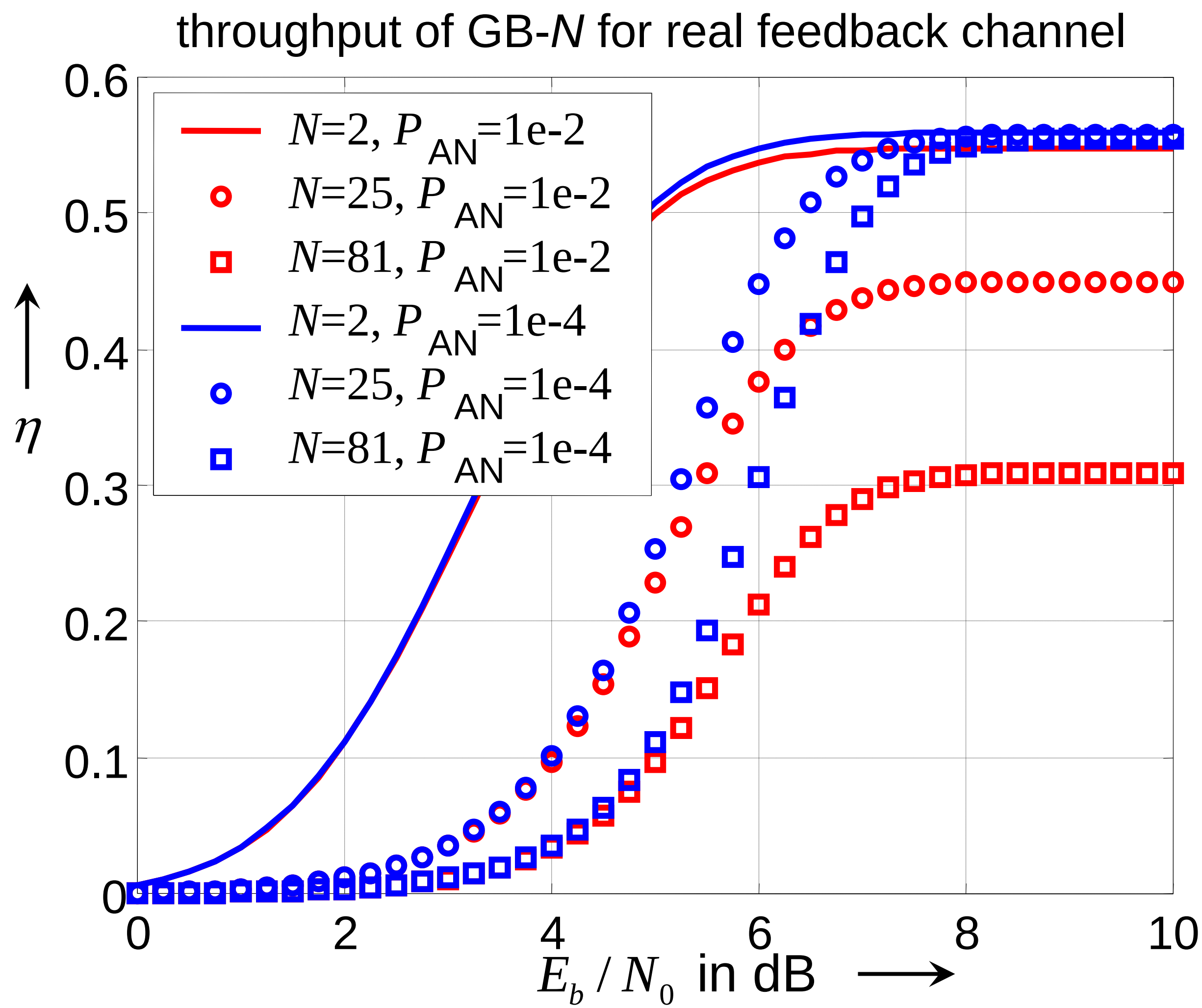
Comparison for Real Feedback Channel (2)



- Systems:
  - $N = 2$: Beam radio
  - $N = 25$: Satellite link, ($T_B = 20$ ms)
  - $N = 81$: Satellite link, ($T_B = 6$ ms)
- Results
  - For more reliable feedback channel efficiency for different N approach same value
  - For bad feedback channel the efficiency is strongly reduced for larger N

R_c=0.55

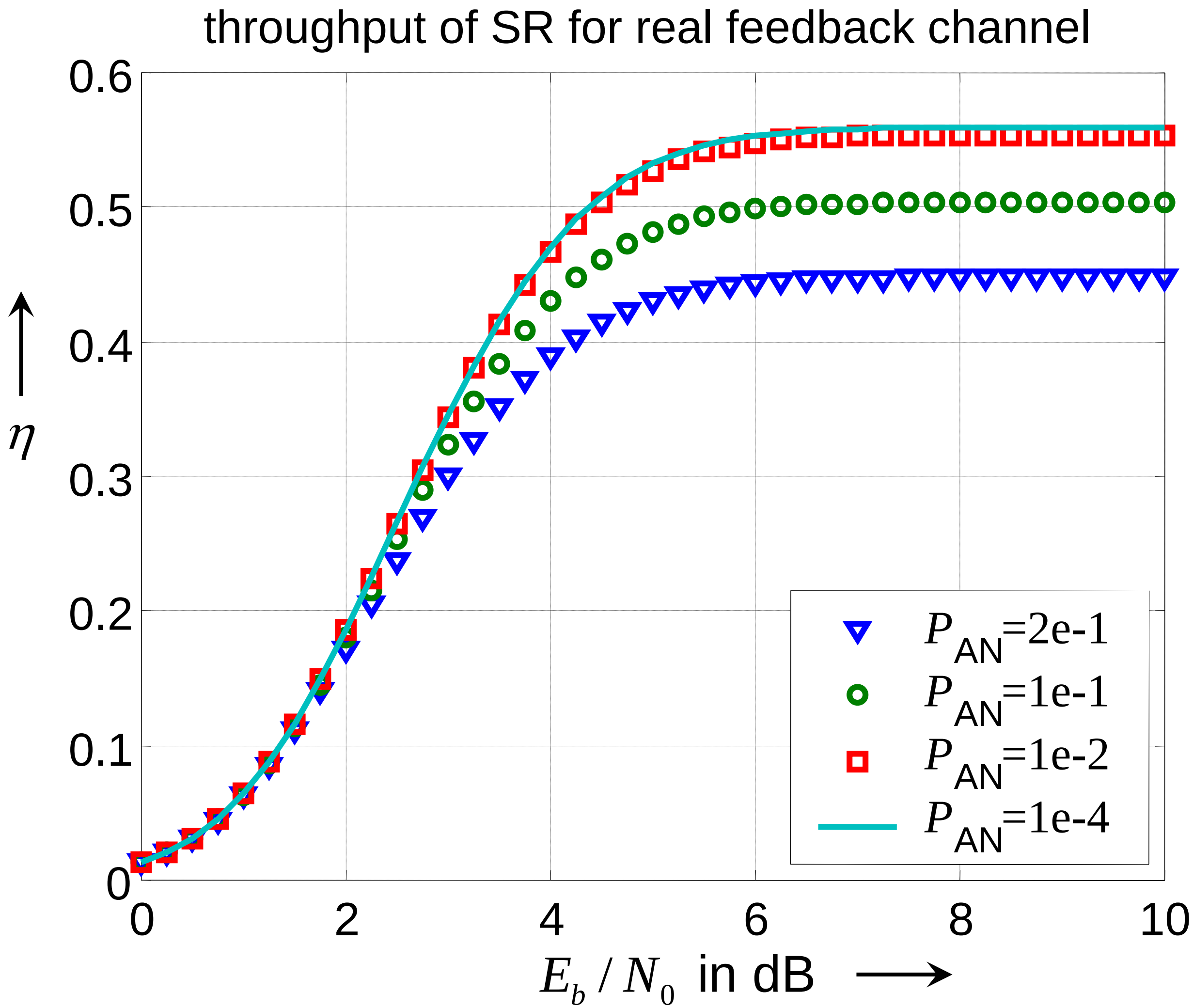
Comparison for Real Feedback Channel (2)



- Systems:
  - $N = 2$: Beam radio
  - $N = 25$: Satellite link, ($T_B = 20$ ms)
  - $N = 81$: Satellite link, ($T_B = 6$ ms)
- Results
  - For more reliable feedback channel efficiency for different N approach same value
  - For bad feedback channel the efficiency is strongly reduced for larger N

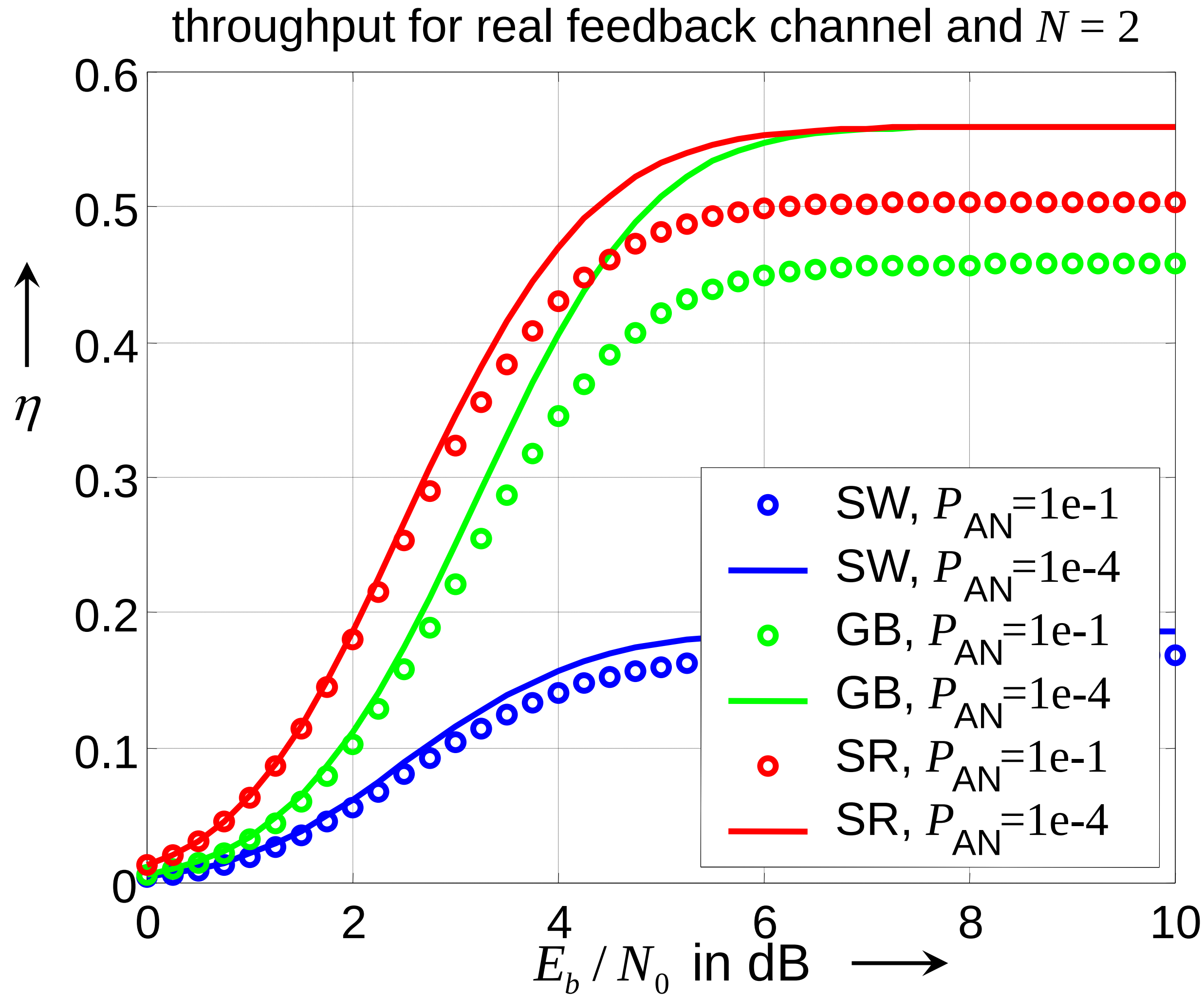
R_c=0.55

Comparison for Real Feedback Channel (3)



- Systems:
  - Efficiency is independent of N
- Results
  - Robust w.r.t. errors on feedback channel → instead of N only 1 package is repeated in case of an error
  - Efficiency is independent of P_{NA}
  - Note: SR can not be implemented in this direct form

R_c=0.55 Comparison for Real Feedback Channel (4)

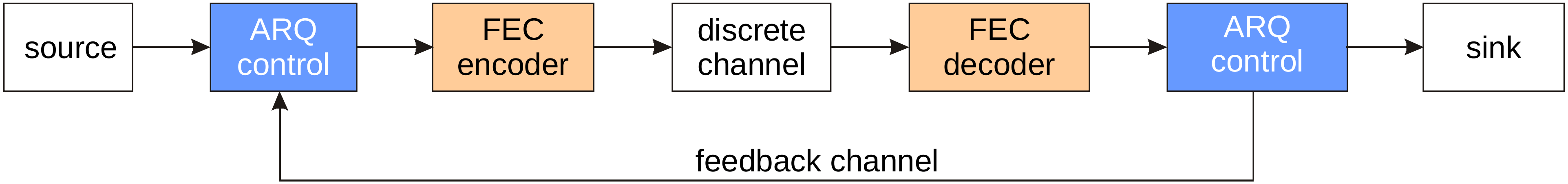


- $N = 2$: Beam radio
- Results
  - SR achieves largest efficiency in general
  - Differences increase for feedback channels of low quality
  - For larger N this difference would be even larger

HYBRID FEC/ARQ SYSTEMS

Hybrid FEC/ARQ Systems

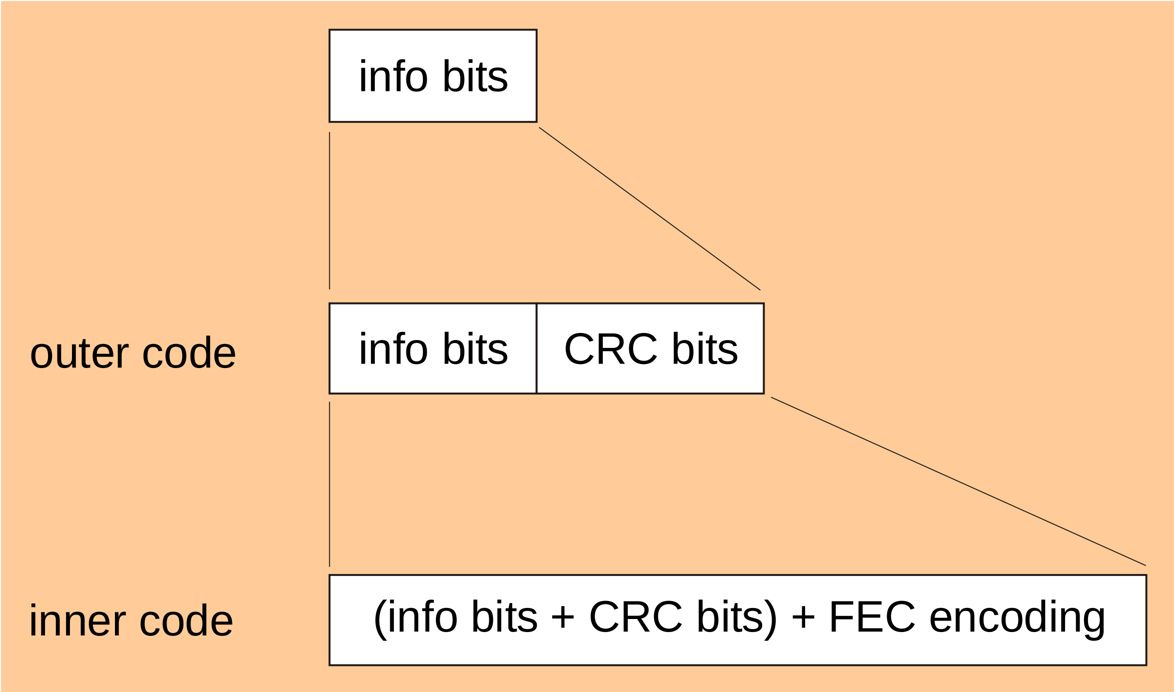
- Both FEC and ARQ show some drawbacks:
  - FEC is able to achieve very low error rates for sufficiently large SNR, but can not guarantee error-free transmission in case of bad channel conditions. For good channel conditions more redundancy than necessary is added.
  - ARQ achieves almost error-free transmission in case of a strong CRC-code, but the throughput may tend to zero. Large throughput is achieved for good channel conditions, as CRC requires less redundancy. For bad channels the repetitions decrease the throughput.



- Combine both strategies by serial concatenation to exploit advantages of FEC and ARQ and avoid their drawbacks
  - Good to medium SNR: FEC achieves almost error-free transmission → no repetition
  - Bad SNR: ARQ generates retransmissions

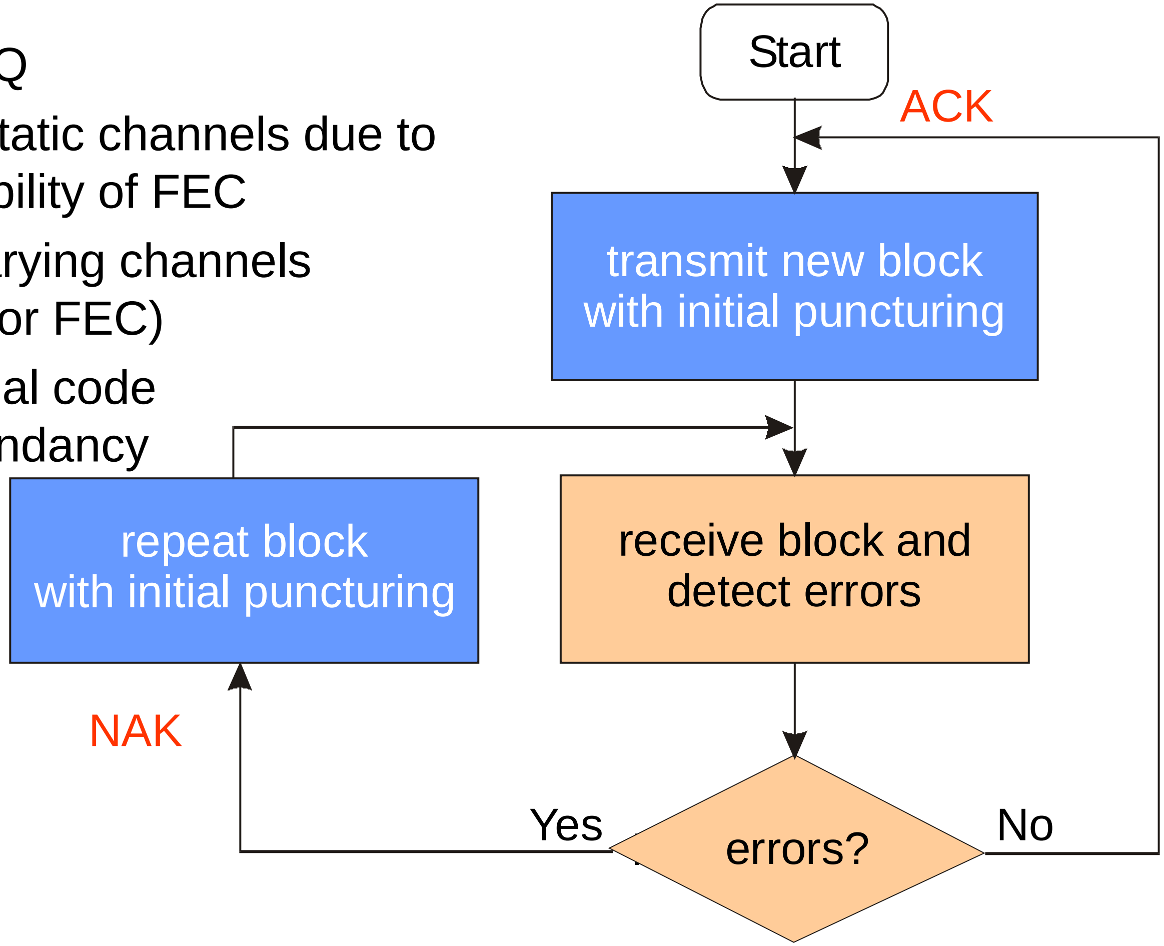
Type-I Hybrid ARQ System

- Easiest combination of FEC and ARQ leading to good performance for almost constant transmission conditions (quasi-static channel)
- Outer error detecting code is concatenated with inner error correcting code



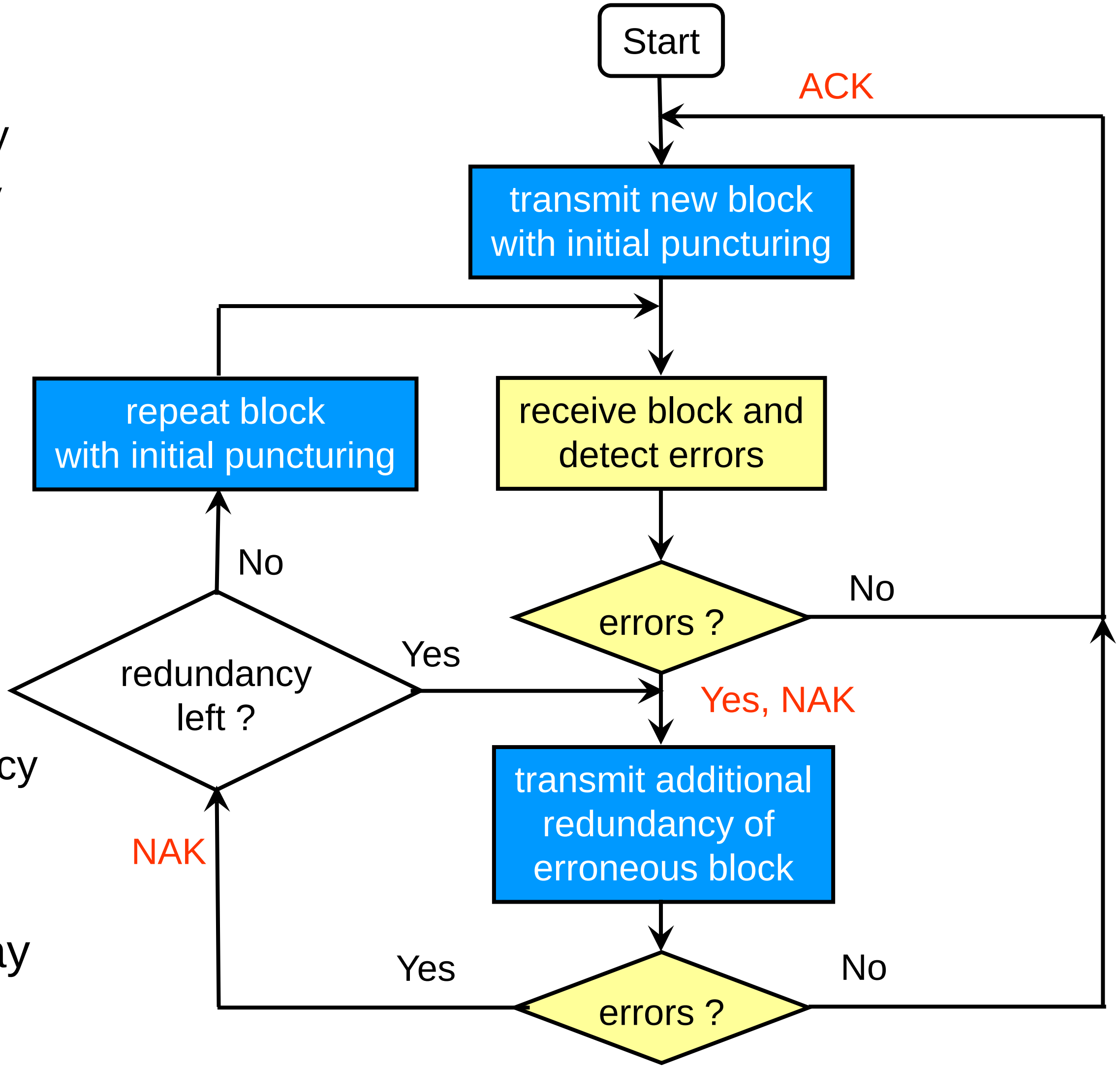
Type-I Hybrid ARQ System (Repetition Coding)

- Properties
  - Data flow identical to pure ARQ
  - Good performance for quasi-static channels due to adapted error correction capability of FEC
  - But poor adaptation to time-varying channels (requiring adaptive code rate for FEC)
  - Fixed rate of inner convolutional code often introduces to much redundancy
- Adaptive code rate of error correcting code desirable



Hybrid ARQ System with Rate-Compatible Punctured Convolutional Codes

- Goal: **adaptive FEC**
  - Good channel → less redundancy
  - Bad channel → more redundancy
- Solution:
  - Successive transmission of redundancy
- Principle:
  - In case of error detection not whole block is repeated but only punctured redundancy
  - Re-decoding of block with buffered first part and additionally transmitted redundancy
  - Less redundancy required
  - Higher efficiency
- Drawback: erroneous first part may lead to decoding failure



Rate-Compatible Punctured Convolutional Codes (1)

- Rate-Compatible Punctured Convolutional Codes (**RCPC-Codes**) [**Hagenauer**]
- Mother code is given by ordinary convolutional code of code rate $R_c=1/4$
- Construction of several puncturing matrices $\mathbf{P}_\ell$ to achieve distinct code rates

$$R_c^{(\ell)} = \frac{L_P}{L_P + \ell}$$

Puncturing period L_P
Parameter $\ell = 1,\ldots,(n\text{-}1)L_P$

$\ell = (n\text{-}1)L_P$ → $R_c = 1/4$ (no puncturing)
$\ell = 1$ → $R_c = 8/9$

- By using different puncturing matrices
  - additional redundancy (parity bits) can be transmitted successively, if non-correctable error was detected
  - the effective code rate can easily be adjusted due to current channel condition
- Rate-Compatibility: Puncturing matrices satisfy

$$p_{i,j}(\ell_0)=1 \quad \Rightarrow \quad p_{i,j}(\ell)=1 \;\; \forall \ell \ge \ell_0 \ge 1$$
$$p_{i,j}(\ell_0)=0 \quad \Rightarrow \quad p_{i,j}(\ell)=0 \;\; \forall \ell_0 \le \ell \le (n-1)L_P - 1$$

w.r.t. some reference index ℓ_0

→

More redundancy: All redundancy bits transmitted by $\mathbf{P}_\ell$, are also transmitted by $\mathbf{P}_{\ell+1}$. Only parity bits are added that have been punctured so far.
Less redundancy:
Only parity bits transmitted so far, are punctured. Already punctured code bits are still punctured.

Rate-Compatible Punctured Convolutional Codes (2)

- Generator polynomials of mother code (L_c = 5, R_c = 1/4) [Hagenauer]

$$g_0 = 1 + D + D^4 \qquad g_2 = 1 + D + D^2 + D^4$$
$$g_1 = 1 + D^2 + D^3 + D^4 \qquad g_3 = 1 + D + D^3 + D^4$$

$$R_c^{(\ell)} = \frac{L_P}{L_P + \ell}$$

- Puncturing matrices (L_p = 8)

$$R_c = \frac{1}{4} \quad \rightarrow \quad \mathbf{P}_0 = \left(\begin{array}{cccc|cccc} 1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1 \end{array}\right)$$

$$R_c = \frac{4}{15} \quad \rightarrow \quad \mathbf{P}_1 = \left(\begin{array}{cccc|cccc} 1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1 \\ 1&1&1&0&1&1&1&0 \end{array}\right)$$

$$R_c = \frac{2}{7} \rightarrow \mathbf{P}_2 = \left(\begin{array}{cccc|cccc} 1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1 \\ 1&0&1&0&1&0&1&0 \end{array}\right)$$

$$R_c = \frac{4}{13} \quad \rightarrow \quad \mathbf{P}_3 = \left(\begin{array}{cccc|cccc} 1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1 \\ 1&0&0&0&1&0&0&0 \end{array}\right)$$

$$R_c = \frac{1}{3} \quad \rightarrow \quad \mathbf{P}_4 = \left(\begin{array}{cccc|cccc} 1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0 \end{array}\right)$$

$$R_c = \frac{4}{11} \quad \rightarrow \quad \mathbf{P}_5 = \left(\begin{array}{cccc|cccc} 1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1 \\ 1&1&1&0&1&1&1&0 \\ 0&0&0&0&0&0&0&0 \end{array}\right)$$

Rate-Compatible Punctured Convolutional Codes (3)

$$R_c = \frac{4}{10} \quad \rightarrow \quad \mathbf{P}_6 = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$R_c = \frac{1}{2} \quad \rightarrow \quad \mathbf{P}_8 = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$R_c = \frac{2}{3} \quad \rightarrow \quad \mathbf{P}_{10} = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$R_c = \frac{8}{9} \quad \rightarrow \quad \mathbf{P}_{12} = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$R_c = \frac{4}{9} \quad \rightarrow \quad \mathbf{P}_7 = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

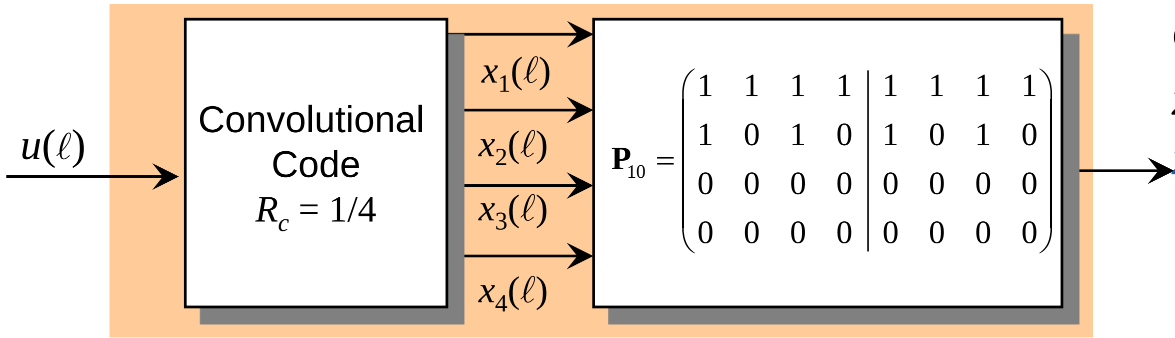
$$R_c = \frac{4}{7} \quad \rightarrow \quad \mathbf{P}_9 = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$R_c = \frac{4}{5} \quad \rightarrow \quad \mathbf{P}_{11} = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

Example

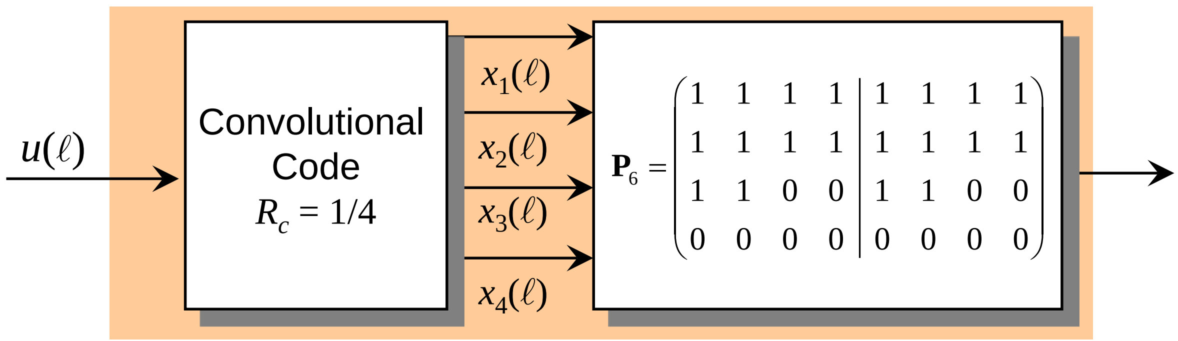
- Encoder output

$$x(\ell) = x_1(0), x_2(0), x_3(0), x_4(0), x_1(1), x_2(1), x_3(1), x_4(1), x_1(2), x_2(2), x_3(2), x_4(2),$$
$$x_1(3), x_2(3), x_3(3), x_4(3), x_1(4), x_2(4), x_3(4), x_4(4), x_1(5), x_2(5), x_3(5), x_4(5), \ldots$$



Coded sequence with $\mathbf{P}_{10}$, i.e. $R_c = 2/3$

$$x(\ell) = x_1(0), x_2(0), x_1(1), x_1(2), x_2(2), x_1(3), x_1(4), x_2(4), x_1(5), x_1(6), x_2(6), x_1(7), \ldots$$



Coded sequence with $\mathbf{P}_6$, i.e. $R_c = 4/10$

$$x(\ell) = x_1(0), x_2(0), x_3(0), x_1(1), x_2(1), x_3(1), x_1(2), x_2(2), x_1(3), x_2(3), x_1(4), x_2(4), x_3(4), x_1(5), x_2(5), x_3(5), x_1(6), x_2(6), x_1(7), x_2(7), \ldots$$

Rate-Compatible Punctured Convolutional Codes (4)

- Generator polynomials of mother code ($L_c = 7$, $R_c = 1/3$) [Hagenauer]

$$g_0 = 1 + D + D^3 + D^4 + D^6 \qquad g_2 = 1 + D^3 + D^4 + D^5 + D^6$$

$$g_1 = 1 + D^2 + D^5 + D^6$$

- Puncturing matrices ($L_p = 8$)

$$R_c = \frac{1}{3} \to \mathbf{P}_0 = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right)$$

$$R_c = \frac{4}{11} \to \mathbf{P}_1 = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{array}\right)$$

$$R_c = \frac{4}{10} \to \mathbf{P}_2 = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{array}\right)$$

$$R_c = \frac{4}{9} \to \mathbf{P}_3 = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right)$$

$$R_c = \frac{1}{2} \to \mathbf{P}_4 = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$R_c = \frac{4}{7} \to \mathbf{P}_5 = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

Rate-Compatible Punctured Convolutional Codes (5)

$$R_c = \frac{2}{3} \rightarrow \mathbf{P}_6 = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

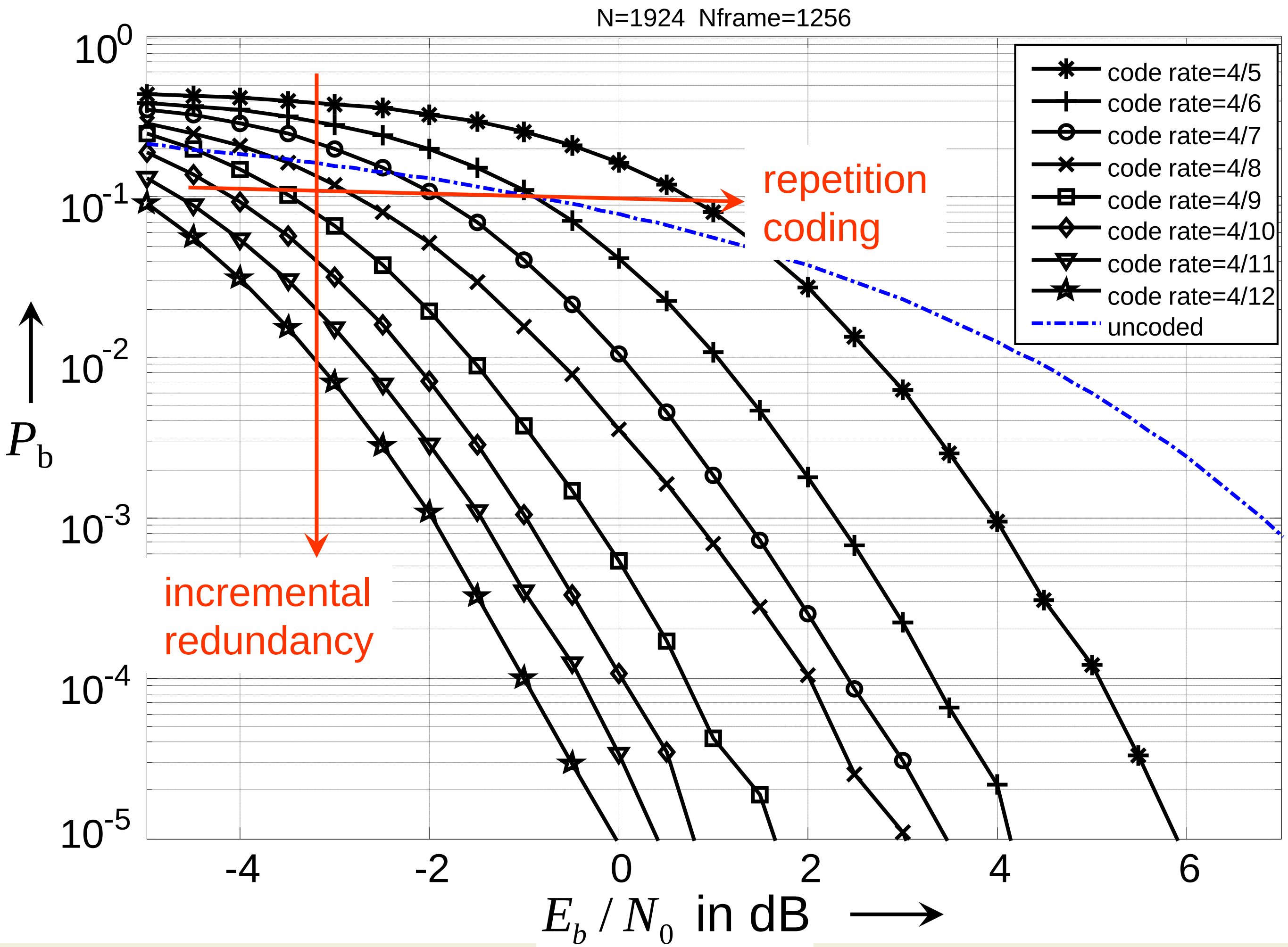
$$R_c = \frac{4}{5} \rightarrow \mathbf{P}_7 = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$R_c = \frac{8}{9} \rightarrow \mathbf{P}_8 = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

- Drawback of Type-I Hybrid System:

  In case of an error the first transmission block is used for each subsequent decoding step (with added redundancy)
  → if this packet is heavily disturbed, the success is unlikely
  → Type-II Hybrid Systems avoid this drawback

Bit Error Rates for RCPC Family and AWGN Channel



Convolutional code

- memory 4
- generators: $(31, 27, 35)_8$

- Repetition coding: stay on curve and go to right (increase SNR)

- Incremental redundancy: hop from curve to curve at same SNR

Throughput Analysis for Finite Number of Retransmissions

- Probability of detected error equals frame error rate: $\mathrm{FER} = P_{\mathrm{ed}}$
- Average transmission time of packet with at most *r* repetitions

$$T_{AV} = T_B \cdot \left[1 - P_{ed} + 2P_{ed}(1-P_{ed}) + \ldots + (r+1) P_{ed}^{r} (1-P_{ed}) + (r+1) P_{ed}^{r+1} \right]$$

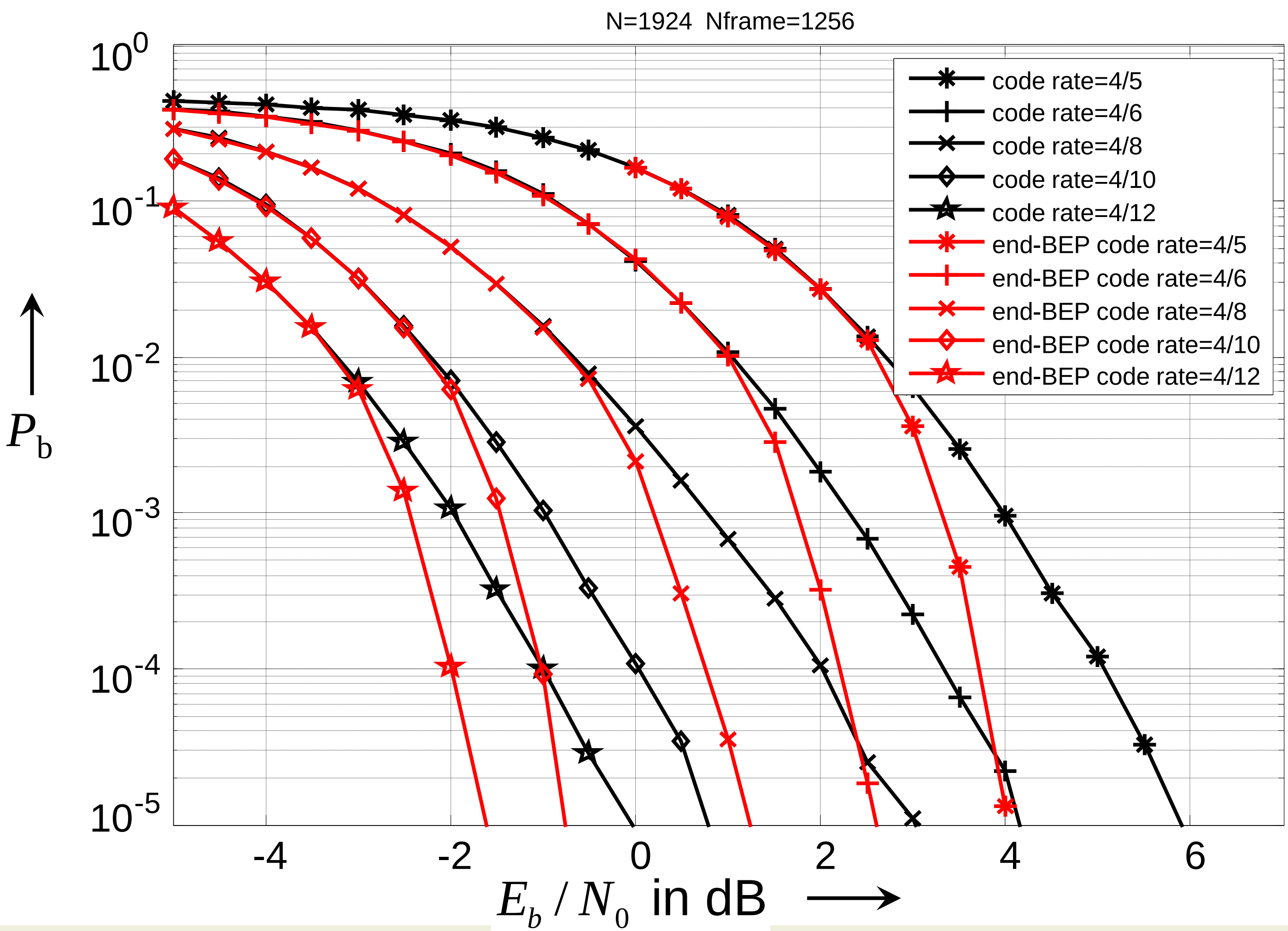
$$= T_B \cdot \left[(1-P_{ed}) \cdot \sum_{i=0}^{r} (i+1) \cdot P_{ed}^{i} + (r+1) P_{ed}^{r+1} \right]$$

$$= T_B \cdot \left[\frac{1-(r+2) P_{ed}^{r+1} + (r+1) P_{ed}^{r+2}}{(1-P_{ed})} + (r+1) P_{ed}^{r+1} \right] = T_B \cdot \frac{1-P_{ed}^{r+1}}{1-P_{ed}}$$

- Outage (failure after *r* repetitions) probability: $P_{out} = \mathrm{FER}^{r+1}$
- Throughput without Chase combining (only successful transmissions considered)

$$\eta_{GB-N} = \frac{T_B}{T_{AV}} \cdot R_c \cdot (1-P_{out}) = \frac{1-P_{ed}}{1-P_{ed}^{r+1}} \cdot R_c \cdot \left(1-P_{ed}^{r+1}\right) = (1-P_{ed}) \cdot R_c$$

Bit Error Rates for Type-I-ARQ without Chase Combining



Convolutional code

- memory 4
- generators: $(31, 27, 35)_8$

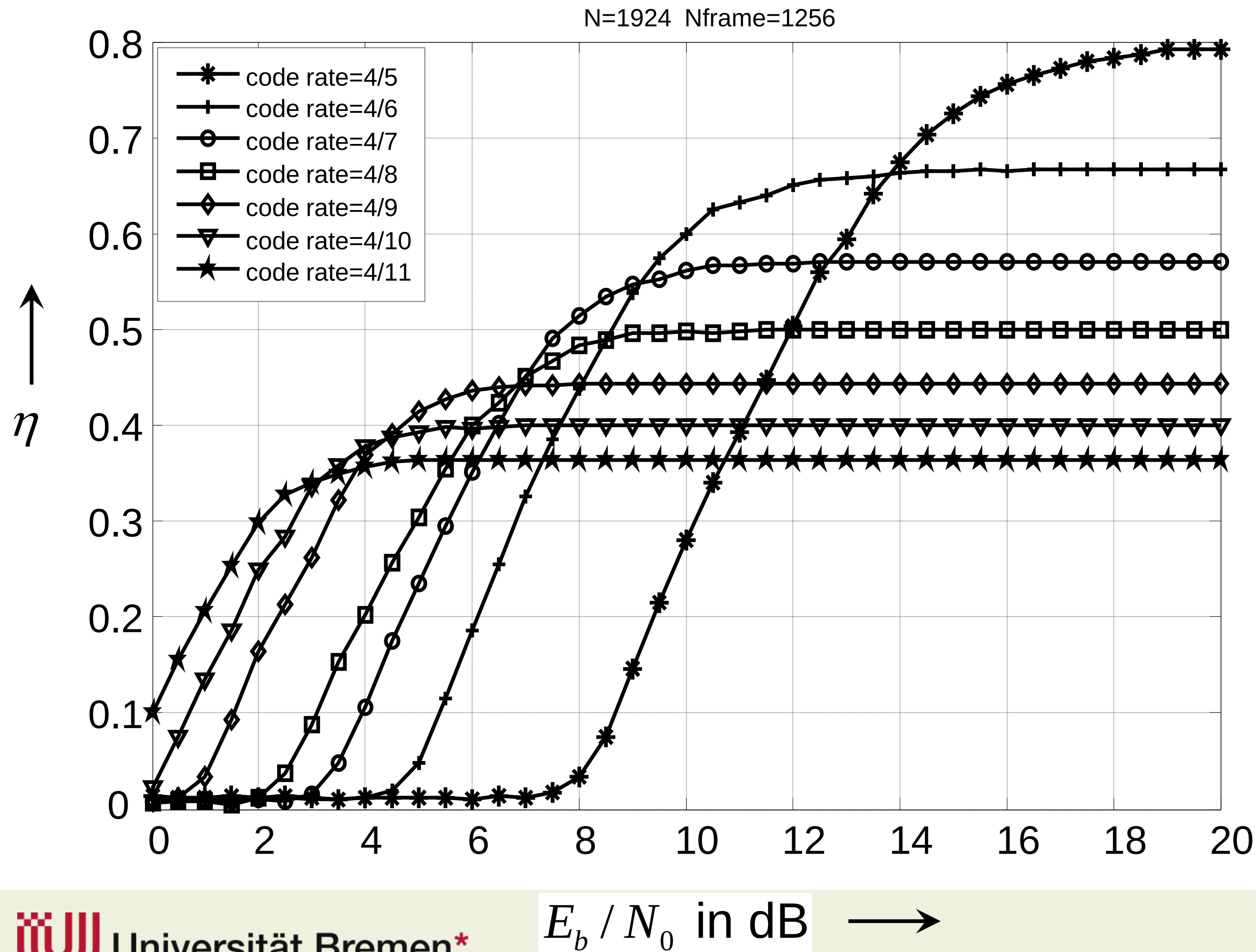
AWGN Channel

ARQ

- max. $r = 3$ retransmissions
- no Chase combining

$$\eta = (1 - P_{ed}) \cdot R_c$$

Throughput for Type-I-ARQ without Chase Combining



Convolutional code

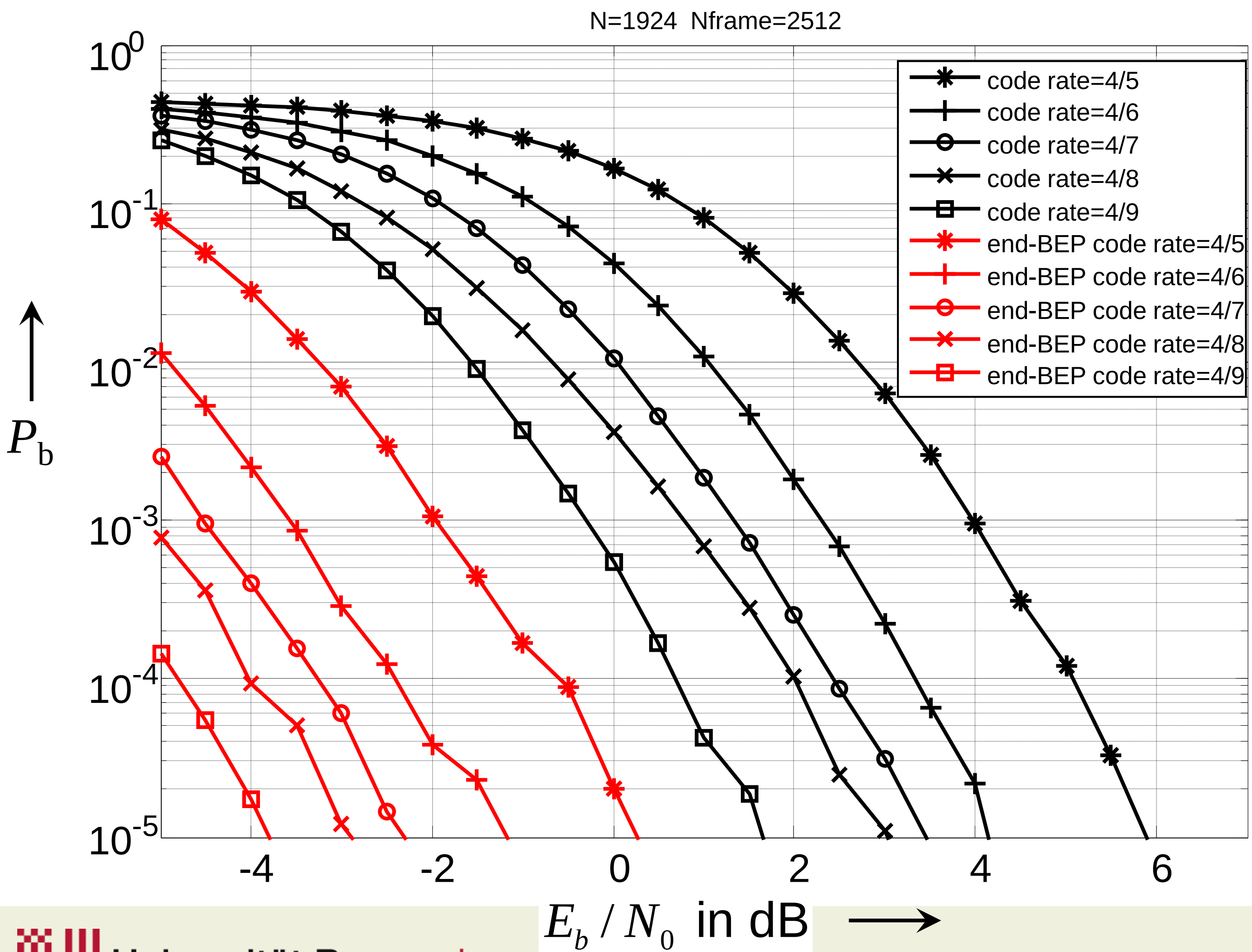
- memory 4
- Rate 1/3
- generators: $(31, 27, 35)_8$
- different puncturing patterns

AWGN Channel

ARQ

- max. $r = 3$ retransmissions
- no Chase combining

Bit Error Rates for Type-I-ARQ with Chase Combining



Convolutional code

- memory 4
- Rate 1/3
- generators: $(31, 27, 35)_8$

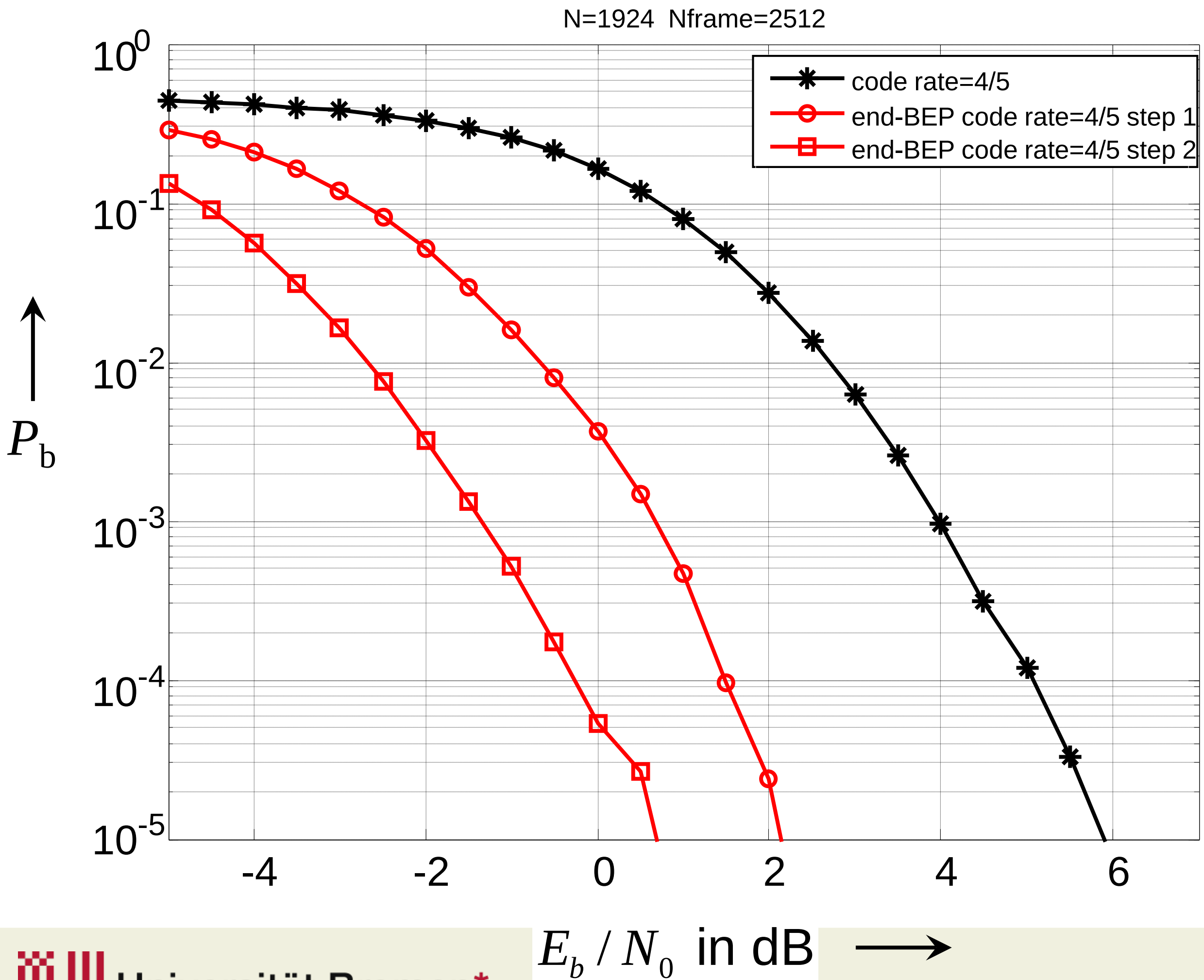
AWGN Channel

ARQ

- repetition coding
- max. $r = 3$ retransmissions
- Chase Combining

→ Curves shifted to left

Bit Error Rates for Type-I-ARQ (Incremental Redundancy)



Convolutional code

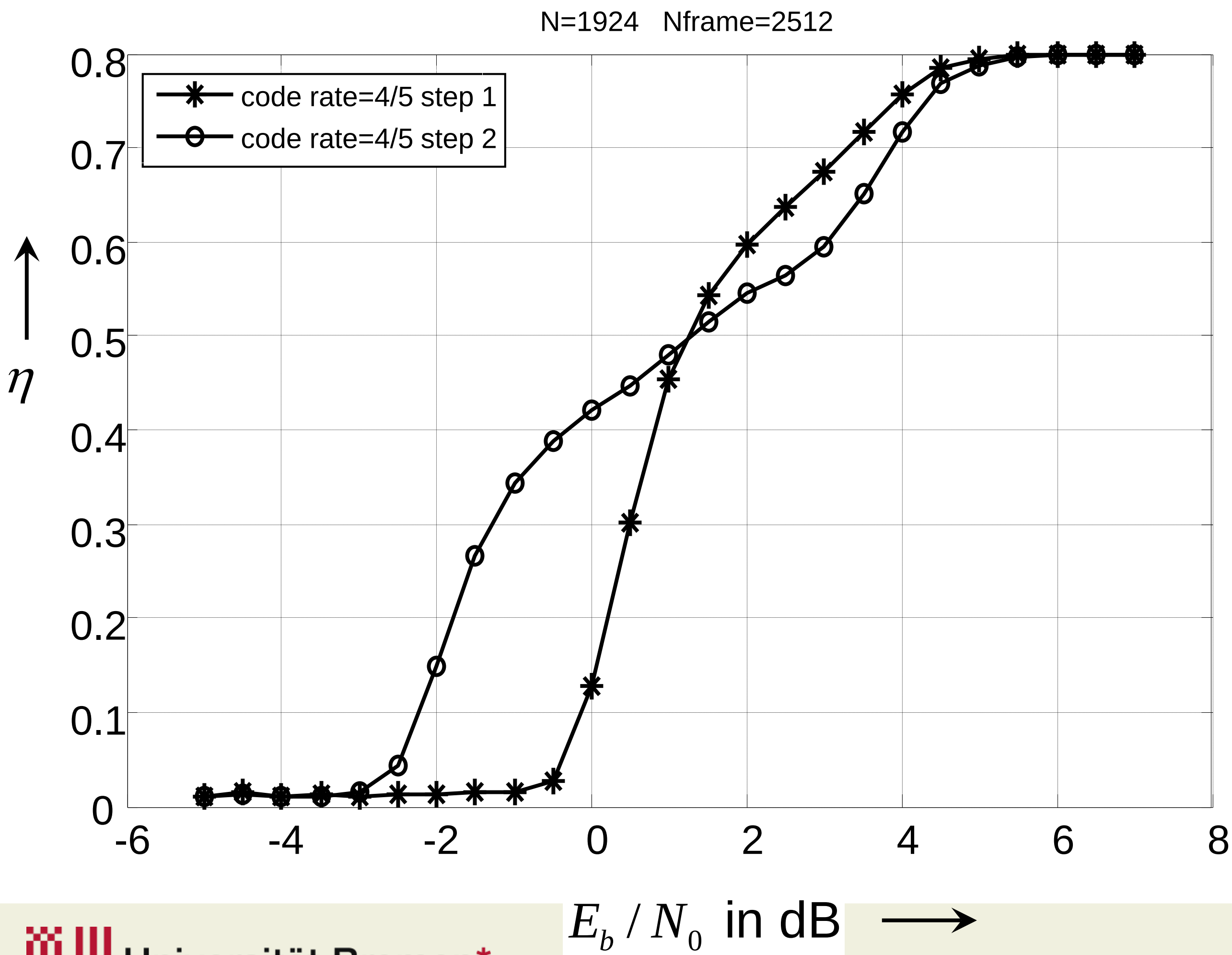
- memory 4
- Rate 1/3
- generators: $(31, 27, 35)_8$

AWGN Channel

ARQ

- incremental redundancy with different step sizes
- max. $r = 3$ retransmissions

Throughput for Type-I-ARQ (Incremental Redundancy)



Convolutional code

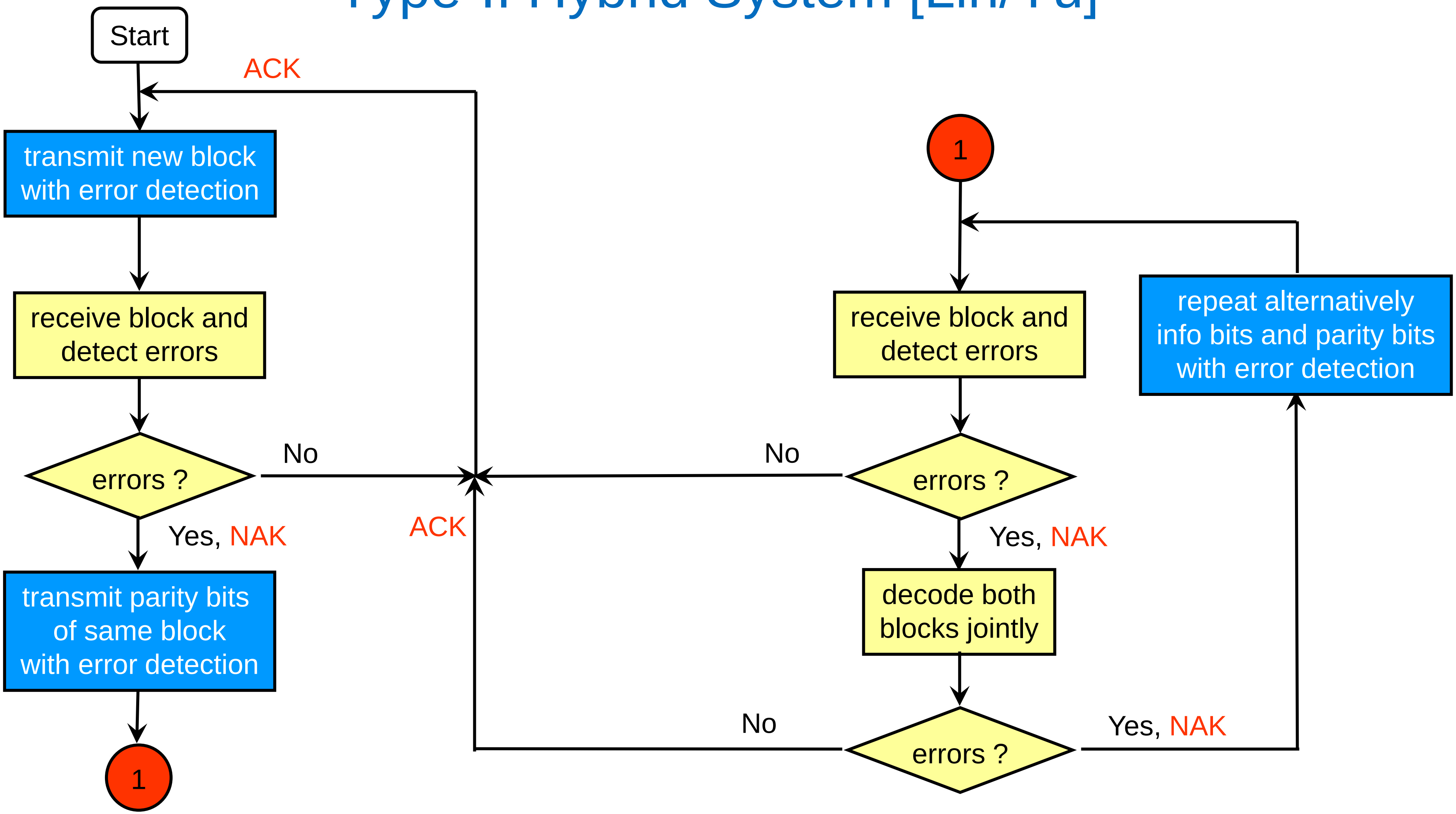
- memory 4
- Rate 1/3
- generators: $(31, 27, 35)_8$

AWGN Channel

ARQ

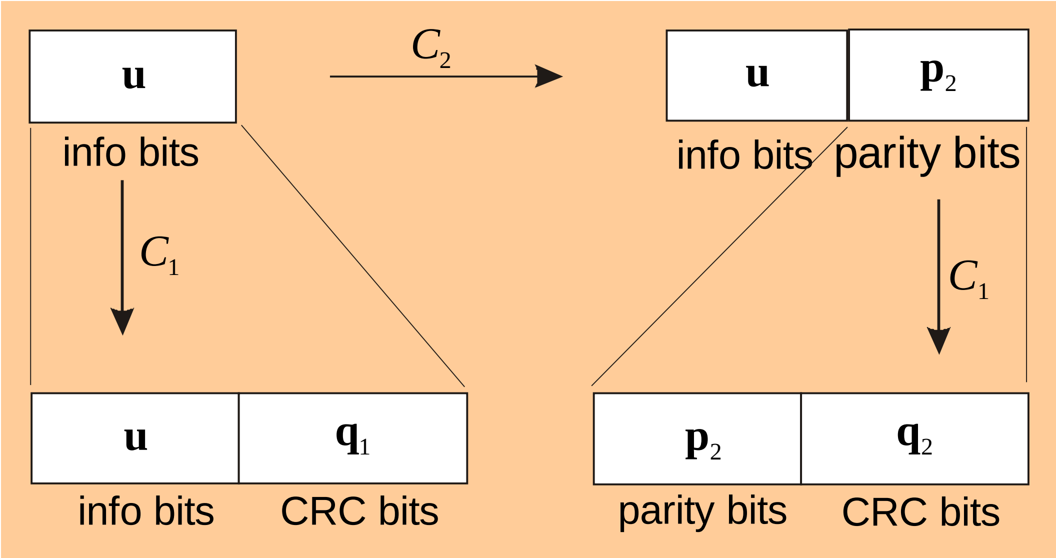
- Incremental redundancy with different step sizes
- max. $r = 3$ retransmissions

Type-II Hybrid System [Lin/Yu]



Encoding for Type-II Hybrid Systems

- High rate error detecting code C_1
- **Systematic invertible error correcting code C_2**
  - Bits $\mathbf{u}$ can be determined uniquely from parity bits $\mathbf{p}_2$
  - As many parity bits $\mathbf{p}_2$ as information bits $\mathbf{u}$ required ($R_c \leq 1/2$)



- Process
  - Information vector $\mathbf{u}$ is encoded by C_1 and codeword $\mathbf{c}_1 = (\mathbf{u},\mathbf{q}_1)$ is transmitted. In the same time $\mathbf{u}$ is encoded by C_2 to generate the parity bits $\mathbf{p}_2$ (not transmitted)
  - In case of an error NAK is sent → encode bits $\mathbf{p}_2$ by C_1 and transmit $\mathbf{c}_2 = (\mathbf{p}_2,\mathbf{q}_2)$
  - If this transmission was correct, $\mathbf{u}$ is given by $\mathbf{p}_2$ as C_2 is invertible, otherwise $(\mathbf{u},\mathbf{p}_2)$ is decoded
  - If this leads again to an error, retransmission of $\mathbf{c}_1 = (\mathbf{u},\mathbf{q}_1)$ and decoding; in case of error combined decoding with $\mathbf{p}_2$, ….

Type-II Hybrid System by Lugand/Costello

- CRC code C_1 with generator $g(D)$
- Use half-rate convolutional code C_2 with generators $g_0(D)$ and $g_1(D)$
  - Shift register structure of $g_i(D)$ equals FIR filter → can be inverted by IIR filter $1/g_i(D)$
- Process
  - Generate code word $c_0(D) = u(D) \cdot g_0(D) \cdot g(D)$
    - Equals half of convolutional code → no redundancy added by C_2, only the information bits are now correlated by convolution with $g_0(D)$
  - For error-free transmission, filter $u(D)\ g_0(D)$ with IIR $1/g_0(D)$ to achieve $u(D)$
  - If transmission error occurs (NAK), transmit $c_1(D) = u(D) \cdot g_1(D) \cdot g(D)$
    - 2nd part of the code word
  - For error-free transmission, filter $u(D)\ g_1(D)$ with IIR 1/$g_1(D)$ to achieve $u(D)$
  - Otherwise, perform joint decoding of ($c_0(D)$, $c_1(D)$) with Viterbi Algorithm
  - If this leads again to an error, retransmission of $c_0(D)$ and decoding; in case of error combined decoding with $c_1(D)$, ….

Type-III Hybrid System

- Combination of rate-compatible convolutional code and Type-II hybrid system

- Complementary puncturing by multiple puncturing matrices:
  - Each puncturing matrix generates a sequence which is decodable on its own
  - Each code bit is transmitted with at least 1 puncturing matrix
  - Multiple transmission of code bits by several matrices is allowed

$$\mathbf{P}_1 = \left(\begin{array}{cccc|cccc} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{array}\right) \qquad \mathbf{P}_2 = \left(\begin{array}{cccc|cccc} 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right)$$

$$\mathbf{P}_1 + \mathbf{P}_2 = \left(\begin{array}{cccc|cccc} 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \\ 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right)$$

- Each constituent code has rate of 1/3
- Total rate of 1/6 for 2 transmissions