Recent advances in HARQ communications

— Tutorial to be presented at ICT 2019, Hanoi —

Presented by Pierre Duhamel (CNRS/CentraleSupelec/L2S, France)
Co-authors : Leszek Szczecinski (INRS, Canada), Philippe Ciblat (Telecom ParisTech, France) and Francesca Bassi (CNRS/CentraleSupelec, France)

April, 2019

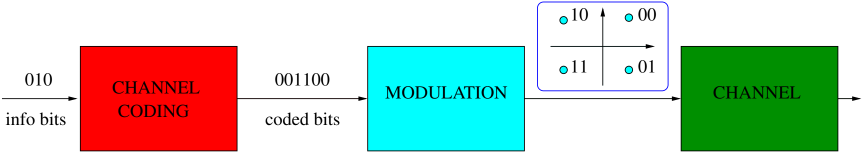
With many contributions from

Faton Maliqi, Alaa Khreis, Mohamed Jabi

This work was partly supported by the Labex Digicosme PhD scholarship from Université Paris-Saclay

Context : (Short) description of a simplified wireless commutation scenario

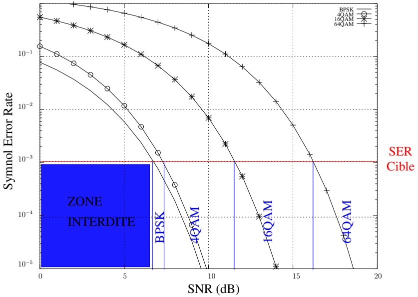
Transmitter (TX):



Traditional presentation :

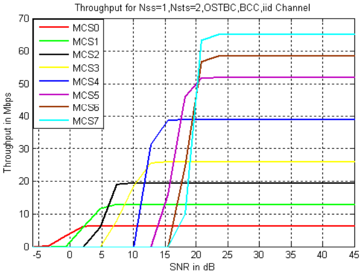
- Adaptive Modulation and Coding: adapts the amount of information transmitted to the "quality" of the channel
  - obviously requires the transmitter to know the channel parameters
  - and to have a performance model for the considered channel
- Transmitter does not know if the transmission failed

Example: AMC with QAM modulation



In actual situations : there exists a target error rate...

Another example: AMC with QAM modulation in 802.11n



MCS in 802.11n, by Meifang Zhu, MSc @ EIT

Drawbacks

- Not many degrees of freedom in the design of AMC
- Would require full knowledge of the instantaneous channel parameters
- When used with average channel conditions, lack of adaptivity (true propagation conditions, noise level ....)

Note also that practical implementations require anyway a feedback channel :
The receivers estimates the "quality" of the channel (usually the SNR) , and sends it back to the transmitter, which is then transmitting with the most appropriate Modulation and Coding Scheme (MCS)

Part 1 : The general picture

However, this is a pure "Physical Layer" point of view, and there could be many problems in the interactions between the various ingredients of a wireless communication network...

Therefore, we spend some time in giving an overview of the aspects that are strongly interconnected... (in order to propose the smartest HARQ...)

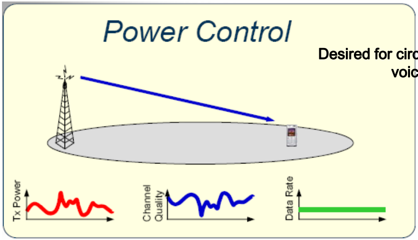
Motivation

- Rapid varying radio channel
  - Time-variant: coherence time (Doppler spread)
  - Frequency-selective: coherence bandwidth (delay spread)
  - Interference
- Exploit the channel variation *prior to* transmission
  - Link adaption : Set transmission parameters to handle radio channel variation
  - Channel-dependent scheduling: Efficient resource sharing among users
- Handle the channel variation *after* transmission
  - Hybrid ARQ : Retransmission request of erroneously received data packets

Link adaptation (1)

Power control:

- Dynamically adjust the transmit power to compensate for the varying radio channel condition
- Maintain a certain SNR at the receiver
- Constant data rate regardless of the channel variation



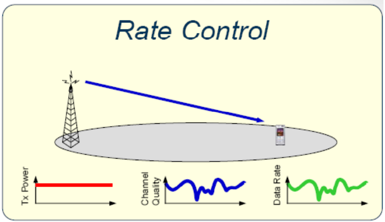
Desired for circuit-switched voice

By S. Parkvall

Link adaptation (2)

Rate control:

- Packet-data traffic: constant rate not a strong desire for constant rate (as high rate as possible)
- Dynamically adjust the data rate to compensate for the varying radio channel condition
- Full constant transmit power (desirable in multiuser systems)



By S. Parkvall

Link adaptation (3)

- Rate control
  - Adaptive Modulation and Coding(AMC) scheme
  - "Good" channel condition: Bandwidth limited (High-order modulation + high-rate coding)
  - "Poor" channel condition: Power limited (Low-order modulation + low-rate coding)
- In HSDPA link adaptation
  - QPSK for noisy channels and 16 QAM for clearer channels
  - 14Mbps, on clear channels using 16-QAM and close to1/1coding rate.
  - 2.4 Mbps, on noisy channels using QPSK and 1/3 coding rate (14 Mbps x 1/2 x 1/3)
  - This adaptation is performed up to 500 times per second

Link adaptation (4)

- Power control: constant rate
  - Desired for voice/video (Short-term rate variation not an issue with constant average data rate)
  - Inefficient use of transmit power
- Rate control: constant (max) transmit power
  - Adaptive data rate
  - Efficient use of transmit power
  - Desired in multiuser systems to reduce variations in interference power

[Chung & Goldsmith, 2001] Little spectral efficiency is lost when the power or rate is constrained to be constant, with optimal adaption.

Scheduling

- The allocation of the shared resources among the users at each time instant
  - Whom?
  - How?
- Joint function with link adaption
- Channel dependent
- Downlink scheduling => Centralized resource
- Uplink scheduling => Distributed resource

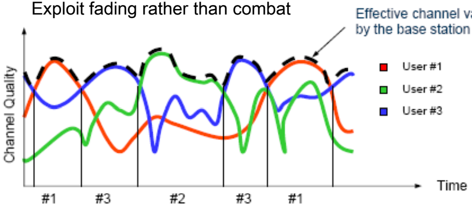
Two examples below of extreme choices for Downlink scheduling, and a more reasonable one (we do not consider uplink in this context description...)

Downlink Scheduling (1)

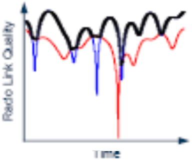
- Channel-dependent scheduling
  - Max-C/I (Max rate) scheduler
    - Schedule at the fading peaks
  - Independently varying radio links
    - Multiuser diversity gain
  - High system throughput but not fair

$$k = \arg\max_i R_i$$

Exploit fading rather than combat



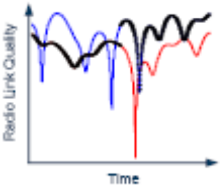
Starve the poor channel user



By S. Parkvall

Downlink Scheduling (2)

- Round-robin scheduling
  - Regardless of channel conditions
  - Fair? … same amount of the radio resources
  - Unfair! … service quality (more resources needed for poor channel)
  - Simple but poor performance



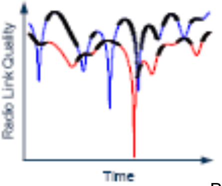
By S. Parkvall

Downlink Scheduling (3)

- Two-fold requirement
  - Take advantage of the fast channel variations
  - Ensure the same average user throughput
- Proportional-fair scheduler
  - Proportion between the instantaneous data rate and the average data rate during a certain period
  - High throughput and fairness

$$k = \arg\max_i \frac{R_i}{\overline{R}_i}$$

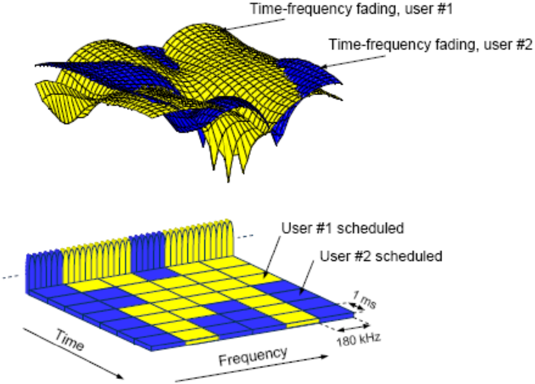
Schedule on fading peaks, regardless of the absolute quality



By S. Parkvall

Downlink Scheduling (3)

- LTE
  - channel-dependent scheduling in time and frequency domains



By S. Parkvall

Requirements on Channel state information

- CSI : Needed at TX for link adaption and channel-dependent scheduling
- Downlink
  - Pilot signal ? e.g.,Correlation channel estimator
  - Measured channel conditions reported to BS => Outdated if high mobility
  - Channel prediction : Additional complexity and constraint
  - Link adaption based on " long-term" average channel

How to adapt to channel's variation? : from AMC to ARQ

Summary : advanced packet radio wireless networks such as HSDPA, channel-dependent scheduling may be used to take advantage of favourable channel conditions to increase the throughput and system spectral efficiency ... (wireless communications are a very "liberal" situation: efficient channels / users should be used as much as possible)

- Since AMC is working with average (non instantaneous) performance,
- Idea: trial and error
  - First send a packet of symbols
  - if correctly received (ACK), $\nearrow$
  - if residual errors (NACK), $\searrow$ and send again a packet containing "same" information...
- This requires feedback channel : information on the instantaneous channel, and the success of the transmission.

.... and do not forget that there is delay in the feedback : processing time, transmission time, etc...

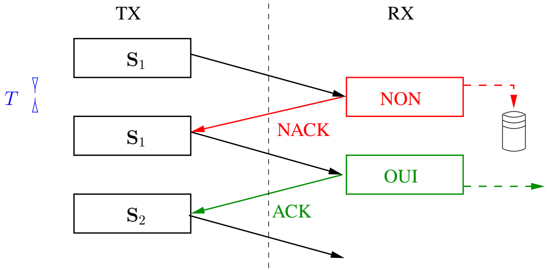
Part 2 : Classical ARQ/HARQ situations

ARQ (*Automatic ReQuest*) overview : the ingredients

- Forward Error Correction (FEC)
  - Add redundancy for error correction
- Automatic Repeat Request (ARQ)
  - Compatible with TCP behavior for packet data
  - Error-detecting code by Cyclic Redundancy Check (CRC)
  - CRC used as a check sum to detect errors (Division of polynomials in Galois field GF(2)...remainder...)
  - No error? Positive acknowledgement (ACK)
  - Error? Negative acknowledgement (NAK)
- Hybrid ARQ
  - Combination of FEC and ARQ
  - FEC: correct a subset of errors
  - ARQ: if still error detected

From ARQ (*Automatic ReQuest*) ...

Let $\mathbf{S} = [s_0, \cdots, s_{N-1}]$ be a packet composed by N uncoded symbols

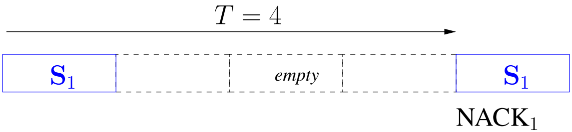


Management for T:

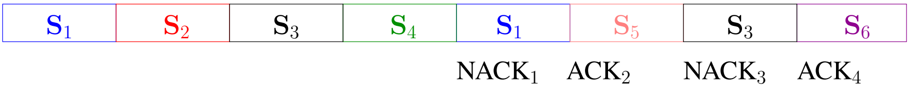
- Stop-and-Wait
- Parallel Stop-and-Wait/Selective Repeat

Management for T

STOP-AND-WAIT



PARALLEL/SELECTIVE-AND-REPEAT



Why $T \neq 1$?

- Decoding processing time at RX
- Framing : traffic for return channel
- Propagation time

Example: $T = 8$ in LTE

... Towards Hybrid ARQ (HARQ): Type-I HARQ

Remark

Retransmission does not contradict forward error coding (FEC)

Type-I HARQ: packet **S** is composed by coded symbols s_n

- first packet is more protected
- there is less retransmission
- transmission delay is reduced

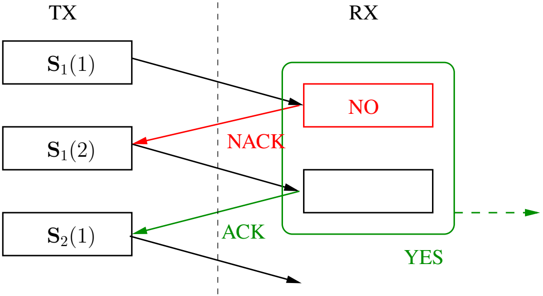
- Efficiency is upper-bounded by the code rate

Drawbacks

- Each received packet is treated independently
- Mis-decoded packet is thrown in the trash

Type-II HARQ

Memory at RX side is considered $\Rightarrow$ Type-II HARQ



Main examples:

- *Chase Combining* (CC)
- *Incremental Redundancy* (IR)

Examples: CC-HARQ and IR-HARQ

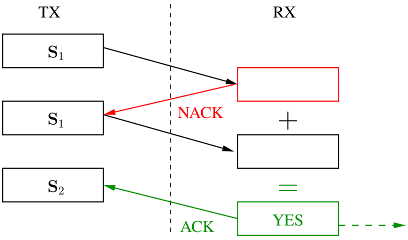
CC

$$\begin{aligned} Y_1 &= S_1 + N_1 \\ Y_2 &= S_1 + N_2 \end{aligned}$$

then detection on

$$Y = (Y_1 + Y_2)/2$$

SNR-Gain equal to 3dB



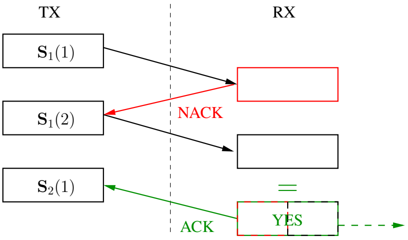
IR

$$\begin{aligned} Y_1 &= S_1(1) + N_1 \\ Y_2 &= S_1(2) + N_2 \end{aligned}$$

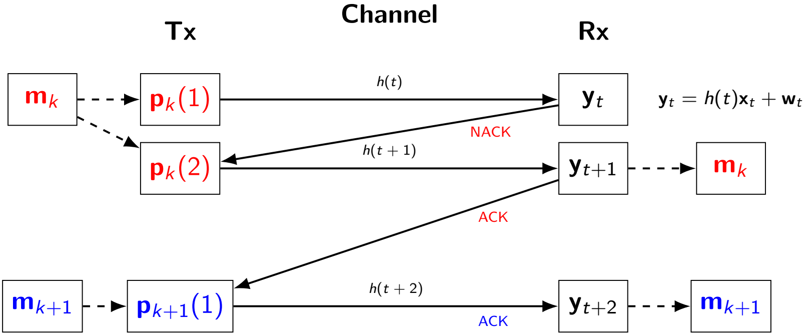
then detection on

$$Y = [Y_1, Y_2]$$

Coding gain



Hybrid ARQ (Automatic Repeat reQuest)



$h(t)$: Rayleigh flat fading channel

$\mathbf{p}_k(\ell)$: ℓ-th packet of message $\mathbf{m}_k$, $\ell \in \{1, \cdots, C\}$

$\mathbf{p}_k(1) = \mathbf{p}_k(2)$	for CC-HARQ (Chase Combining)	$\longrightarrow$ diversity gain
$\mathbf{p}_k(1) \neq \mathbf{p}_k(2)$	for IR-HARQ (Incremental Redundancy)	$\longrightarrow$ diversity + coding gain

Part 3 : Performance metrics

Performance metrics

- **Packet Error Rate** (PER):

$$\text{PER} = \text{Prob(information packet is not decoded)}$$

- **Efficiency** (*Throughput*/*Goodput*/etc):

$$\eta = \frac{\text{information bits received without error}}{\text{transmitted bits}}$$

- **(Mean) delay**:

$$d = \#\ \text{transmitted packets when information packet is received}$$

- **Jitter**:

$$\sigma_d = \text{delay standard deviation}$$

Quality of Service (QoS)

- Data: PER and efficiency
- Voice on IP: delay
- Video Streaming: efficiency and jitter

Part 4 : Degrees of freedom in the design of HARQ

4.1 rate allocation and adaptation (Leszek)

4.2 power allocation and adaptation (Leszek)

4.3 non orthogonal HARQ; reducing the delay and improving the throughput (Pierre)

Non orthogonal HARQ; reducing the delay and improving the throughput

- State of the Art ($T = 1$)
- Application to $T \neq 1$

State of the Art ($T = 1$)

Sending the superposition of two streams instead of one !

$$\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{w}$$

<u>But</u> superposition does not increase the capacity

$$R = R_1 + R_2 < \log_2(1 + P_1 + P_2) = \log_2(1 + P)$$

with P the transmit power.

<u>However</u> a way to be closer to the capacity, especially with retransmission (since ACK/NACK provides information)

Main Idea [Steiner06]:

- Frame 1: send two messages under superposition coding (SC), i.e., two layers with short power constraints P
- Frame 2: if one layer not decoded, send it again with full power P
- Frame 3: start with two new messages

Two contexts:

- Channel constant over each retransmission
- Channel time-varying at each retransmission

Additional works:

- Practical implementation of [Steiner06] with $P_1 = 0.8P$ [Assimi2009]
- CSI at the TX for relevant actions (SC or not with Markov Decision Process) [Jabi2015]
- At TCP level: flooding the TCP packet with hierarchical superposition coding [Zhang2009]

Application to $T \neq 1$

Idea To reduce the delay, send in advance (before receiving any ACK/NACK) redundant packets in superposition to standard parallel HARQ with low power (for minimizing the disturbance):

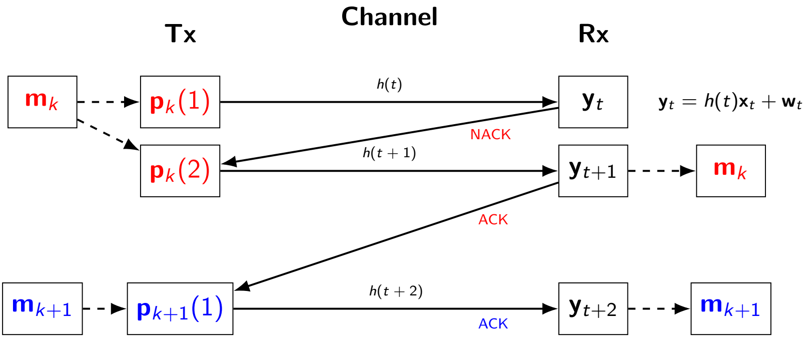
$$\begin{cases} \mathbf{S}_k(\ell), & \text{if no superposition,} \\ \sqrt{\alpha}\mathbf{S}_k(\ell) + \sqrt{1-\alpha}\mathbf{S}_{k'}(\ell'), & \text{if superposition.} \end{cases}$$

with k, k' the messages.

We have two layers :

- The first one is standard parallel HARQ
- The second one corresponds to superposed packets chosen as:
  1. $\mathbf{S}_{k'}(\ell')$ is not superposed if $\mathbf{m}_{k'}$ is in timeout or previously ACKed
  2. Superposed packet is the unsent packet of the lowest index ℓ' of the most recent message $\mathbf{m}_{k'}$, with $k' \neq k$
  3. If the transmitter already sent all the packets, superposed packet is with the lowest index ℓ' not previously sent in the second layer.
  4. No packet is superposed to a packet of the first layer that has $\ell = L$.

Hybrid ARQ (Automatic Repeat reQuest)



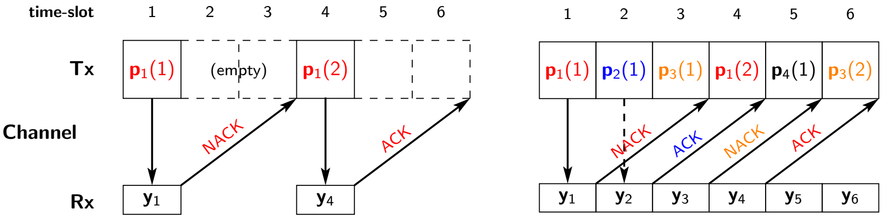
$h(t)$: Rayleigh flat fading channel

$\mathbf{p}_k(\ell)$: ℓ-th packet of message $\mathbf{m}_k$, $\ell \in \{1, \cdots, C\}$

$\mathbf{p}_k(1) = \mathbf{p}_k(2)$ for CC-HARQ (Chase Combining) $\rightarrow$ diversity gain

$\mathbf{p}_k(1) \neq \mathbf{p}_k(2)$ for IR-HARQ (Incremental Redundancy) $\rightarrow$ diversity + coding gain

HARQ with feedback delay ($T = 3$)



Stop-and-Wait

Parallel HARQ (Selective Repeat)

Why $T \neq 1$**?** ($T = 8$ in LTE)

- Decoding (processing) time at the receiver
- Framing: traffic for return channel
- Propagation time

Non-orthogonal transmission

Idea

- Superpose (re)transmitted packets to increase the throughput [Shamai08, Assimi09, Szczecinski14]

Objectives

- Low latency
- High reliability
- Large throughput

Why non-orthogonal transmission?

- Non-orthogonal transmission exploits the potential of MAC
- Other strategies usually require CSI at the transmitter [Kasper17]
  - time-sharing
  - rate adaption

MAC: Multiple Access Channel - CSI: Channel State Information

General idea

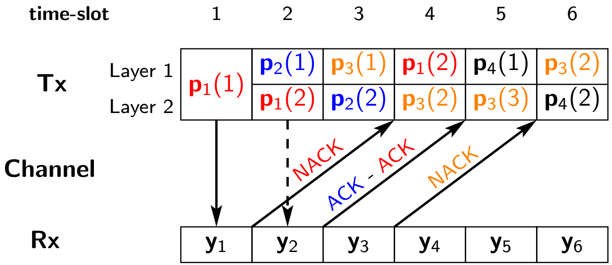
Send additional redundant packets using two layers

- Before receiving the ACK/NACK feedback
  - Superposed to parallel HARQ
    - With low power

Layer 1: parallel HARQ VERY important

Layer 2: superposed packets

$\mathbf{p}_k(\ell)$	without superposition
$\sqrt{\alpha}\mathbf{p}_k(\ell) + \sqrt{1-\alpha}\mathbf{p}_{k'}(\ell')$	with superposition



Proposed protocol, $T = 3$

Transmitter

How do we choose the superposed redundant packets?

- Superpose packets of the most recent messages
  $\rightarrow$Low latency
- Superpose unsent redundant packets
  $\rightarrow$Transmit diversity
  $\rightarrow$High reliability

	time-slot	1	2	3	4	5	6
Tx	Layer 1	$\mathbf{p}_1(1)$	$\mathbf{p}_2(1)$	$\mathbf{p}_3(1)$	$\mathbf{p}_1(2)$	$\mathbf{p}_4(1)$	$\mathbf{p}_3(2)$
	Layer 2		$\mathbf{p}_1(2)$	$\mathbf{p}_2(2)$	$\mathbf{p}_3(2)$	$\mathbf{p}_3(3)$	$\mathbf{p}_4(2)$
		NACK	ACK ACK	NACK			

Proposed protocol, $T = 3$

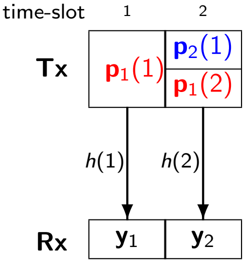
Low latency + High reliability $\rightarrow$Large throughput

Decoding

Let $\mathcal{M}$ be the set of messages that the receiver is attempting to decode at time-slot t.

- If the receiver successfully decodes the subset $\mathcal{D} \subseteq \mathcal{M}$ and none of the messages in $\mathcal{M} \setminus \mathcal{D}$, we say that the decoder operates in the rate region $\mathcal{R}_{\mathcal{D}}$.
- The set $\mathcal{D}$, along with the rules of the transmit protocol, allows to obtain $\mathcal{F}_t$ the set of ACK/NACK.
- In order to characterize the decoding outcome, we
  1. evaluate the rate region $\mathcal{R}_{\mathcal{D}}$ for every possible $\mathcal{D} \subseteq \mathcal{M}$, by checking the corresponding rate inequalities
  2. determine, on the basis of the available observations, the operating rate region $\mathcal{R}_{\mathcal{D}}$ of the receiver.

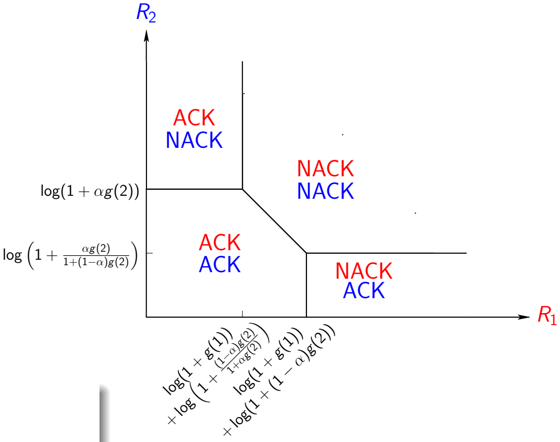
Performance with capacity-achieving codes



Received signals

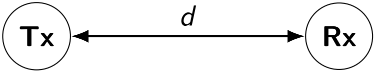
$$\mathbf{y}_1 = h(1)\mathbf{p}_1(1) + \mathbf{w}_1$$

$$\mathbf{y}_2 = h(2)\sqrt{\alpha}\mathbf{p}_2(1) + h(2)\sqrt{1-\alpha}\mathbf{p}_1(2) + \mathbf{w}_2$$



Rate regions at $t=2$ [ElGamal12]
$g(t) = |h(t)|^2$

Setup for numerical evaluation

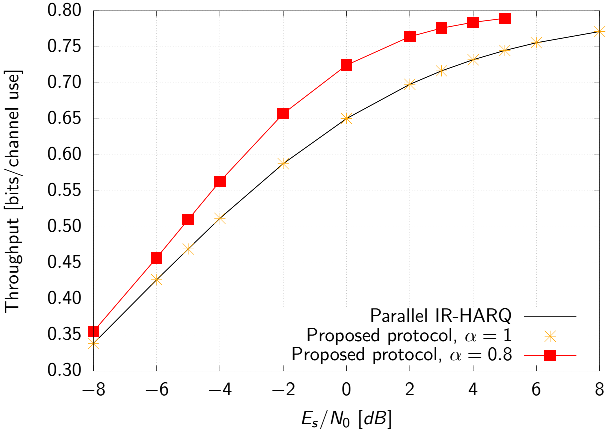


Distance between the transmitter and the receiver

$d = 15u$ where u is a unit of distance

- **Variance** : $\sigma^2 = \left(\frac{c}{d^2}\right)^2$ where c is a constant, fixed as $c = 400u^2$
- **HARQ protocol** : IR-HARQ with $C = 4$ and $R = 0.8$
- **Feedback delay** : $T = 3$ time-slots
- **Transmit energy** : E_s per symbol

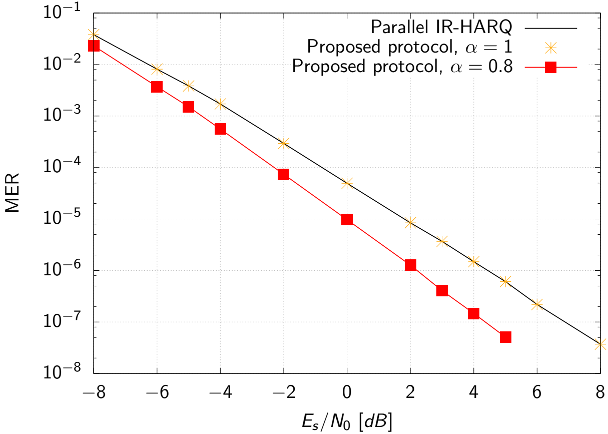
Throughput using capacity-achieving codes



1dB to 2.5dB gain at moderate SNR

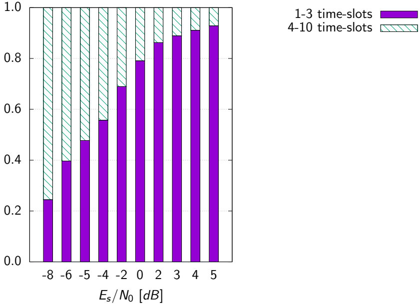
10% throughput gain at 0dB

Message Error Rate using capacity-achieving codes

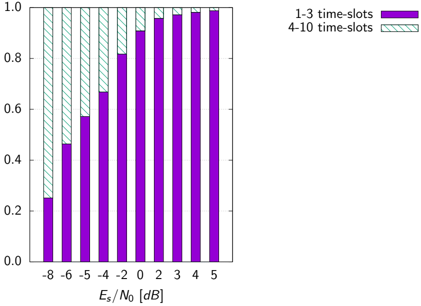


Additional diversity gain due to multi-layer transmission

Latency using capacity-achieving codes



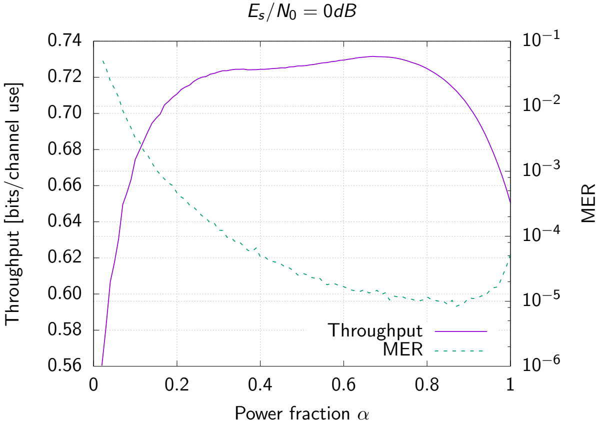
Parallel IR-HARQ



Proposed protocol

More packets are served with small delays ($<$ 4 time-slots)

Numerical optimization of α

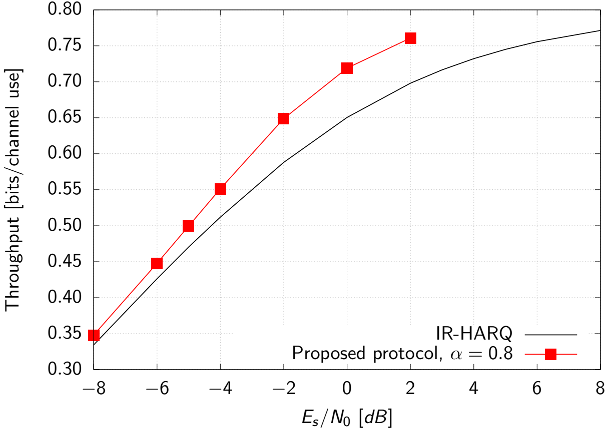


$\alpha = 0.7$ provides the best performance at $0dB$

α can be numerically optimized for each SNR

Proposed protocol in comparison to 3GPP LTE

Throughput using $C = 4$, $T = 8$ and capacity-achieving codes



LTE: Long-Term Evolution - 3GPP: 3rd Generation Partnership Project [TS 36.213] [TS 36.321]

Part 5 : HARQ and AMC; joint or separate design (Leszek)

Cintent :

- separate design: conflicting objectives and counterproductive actions
- joint design: complexity issues
- semi-joint design via layered coding

Part 6 : Extensions and wrap up

Content :

- security : rate adaptation for secure HARQ (Leszek) ???? on maintient ?
- cooperative communications (Pierre)
- conclusions on theoretical and practical issues

Introduction

Interaction between Relaying and HARQ:

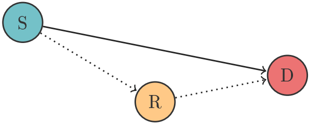
- Both techniques applied solely will bring improvement;
- What improvement will bring if these two techniques are applied together?
- What is the best way of combining them?

Reference literature

- Combination of these two techniques in literature:
  - Energy efficiency is studied in [Stanojev, 2009], and from the perspective of information theory is studied in [Falavarjani, 2010];
  - The interaction is mostly studied via deterministic protocols [Krikidis, 2007]; We focus on both: deterministic and probabilistic protocols;
  - The Relay is mostly considered in Decode-and-Forward (DCF) mode; We focus more on the Demodulate-and-Forward (DMF) mode.
- For theoretical analysis we focus on Finite State Markov Chain (FSMC).

System model

- Example scenario:
  - Source-Relay-Destination network;
  - ARQ mechanism (stop-and-wait policy);
  - All the nodes listen to control messages (ACK/NACK) issued by D.

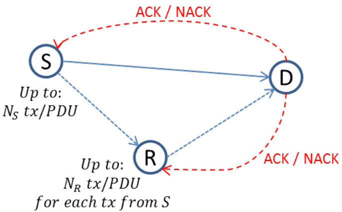


  - Relay mode:
    - Decode-and-Forward (DCF) - Relay always forward the correct copy.
    - Demodulate-and-Forward (DMF) - demodulation errors of R are taken into account when evaluating likelihood function at the decoder:

$$p\left(y_{RD,n}|c_{n,i}\right)=p\left(y_{RD,n}|D_R=0,c_{n,i}\right)p\left(D_R=0|c_{n,i}\right)+p\left(y_{RD,n}|D_R=1,c_{n,i}\right)p\left(D_R=1|c_{n,i}\right)$$

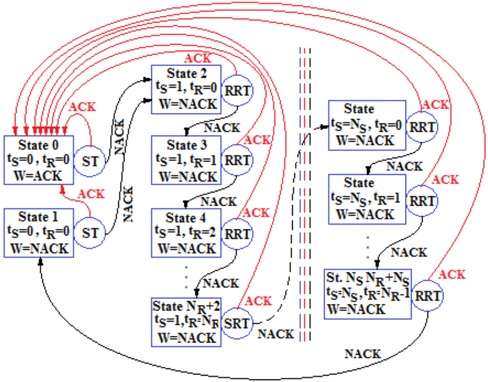
The deterministic protocol, DMF mode

- The example protocol:



- Finite State Machine (FSM):
  - Systematic way for analyzing protocols;
  - FSM enters a state in each time-slot;
  - The state determines the action that is going to be taken during the time-slot;
  - The outcome of the action determines the transition to the next state.

From FSM to FSMC, DMF



- Monte Carlo simulation for evaluation of:
  - $\pi_{[1,0]}$ - probability of NACK on the channel S-D;
  - $\pi_{[0,1]}$ - probability of NACK on the channel R-D;
  - $\pi_{[A,B]}$ - prob. of NACK combining A cop. from S and B cop. from R.

Probability transition matrices, DMF

$$P_{\mathrm{I}} = \begin{pmatrix} 1-\pi_{[1,0]} & 0 & \pi_{[1,0]} & 0 & \cdots & 0 & \cdots \\ 1-\pi_{[1,0]} & 0 & \pi_{[1,0]} & 0 & \cdots & 0 & \cdots \\ 1-\pi_{[0,1]} & 0 & 0 & \pi_{[0,1]} & \cdots & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \\ 1-\pi_{[1,0]} & 0 & 0 & 0 & \cdots & \pi_{[1,0]} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \\ 1-\pi_{[0,1]} & \pi_{[0,1]} & 0 & 0 & \cdots & 0 & \cdots \end{pmatrix}$$

$$P_{\mathrm{II}} = \begin{pmatrix} 1-\pi_{[1,0]} & 0 & \pi_{[1,0]} & 0 & \cdots & 0 & \cdots \\ 1-\pi_{[1,0]} & 0 & \pi_{[1,0]} & 0 & \cdots & 0 & \cdots \\ 1-\pi_{[1,1]} & 0 & 0 & \pi_{[1,1]} & \cdots & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \\ 1-\pi_{[1,N_R]} & 0 & 0 & 0 & \cdots & \pi_{[1,N_R]} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \\ 1-\pi_{[N_S,N_S N_R]} & \pi_{[N_S,N_S N_R]} & 0 & 0 & \cdots & 0 & \cdots \end{pmatrix}$$

Performance evaluation using FSMC

- Performance metrics:
  - PDU error rate (PER) - the proportion of PDUs that were transmitted but never ACK-ed by D;
  - $\overline{T}$ - average number of transmissions per PDU;
  - Goodput (G) - the number of successfully delivered information PDU's per unit of time.
- Performance analysis using FSMC representation:
  - We evaluate the steady state vector $\mathbf{p}$ from matrix P_{I} or P_{II};
  - We obtain the steady state probabilities of the initial states p_0 and p_1;
  - The performance metrics can be obtained as:

$$PER = \frac{p_1}{p_0 + p_1}, \qquad \overline{T} = \frac{1}{p_0 + p_1}$$

$$G = R_c \cdot \frac{1 - PER}{\overline{T}} \left[\frac{PDUs}{tu} \right] = R_c \cdot p_0 \left[\frac{PDUs}{tu} \right]$$

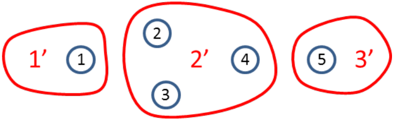
Accurate performance evaluation.... but can become computationally expensive

- As the protocol gets more sophisticated, the FSMC analysis becomes more complex:
  - Increasing the number of nodes or the number of transmissions, the number of states increases very quickly;
  - Switching the Relay from DMF mode to DCF mode, the number of states increases also quickly.

- Resulting number of nodes can quickly become much larger than 100, hence:
  - can we reduce teh size of the FSMC while keeping PER, $\overline{T}$ and G, untouched ? (equivalent to keep State 0 and State 1 untouched);
  - Since each state it is associated with an action, we cannot aggregate states with different actions, and it is more easy to aggregate states with the same actions.

State aggregation on the FSMC

- Let us consider the following example:



- If I is a new state resulting from the aggregation of the set of states $\mathcal{I}$, then the steady state probability of being in state I is:

$$z_I = \sum_{i \in \mathcal{I}} p_i.$$

- The transition probabilities between the aggregated states can be evaluated as:

$$Z_{IJ} = \frac{\sum_{i \in \mathcal{I}} p_i \left(\sum_{j \in \mathcal{J}} P_{ij} \right)}{\sum_{i \in \mathcal{I}} p_i}.$$

State aggregation: simplified FSMC, DMF

- The simplified transition matrix contains only four states:

$$Z=\begin{bmatrix} 1-\pi_{[1,0]} & 0 & \pi_{[1,0]} & 0 \\ 1-\pi_{[1,0]} & 0 & \pi_{[1,0]} & 0 \\ 1-\pi_{[RF]} & \gamma\cdot\beta\pi_{[RF]} & (1-\gamma)\,\pi_{[RF]} & \gamma\,(1-\beta)\,\pi_{[RF]} \\ 1-\pi_{[SF]} & 0 & \pi_{[SF]} & 0 \end{bmatrix}.$$

  where, parameters $\pi_{[RF]}$, $\pi_{[SF]}$, γ and β link the original transition matrix with the simplified one, and can be obtained from the state aggregation procedure;

- The idea of state aggregation can be extended similarly for the case of DCF mode.

Protocol associated with the simplified FSMC

- Aggregation of states:
  - The actions remain the same;
  - Some transitions now will become probabilistic;
  - If we define:

    γ - the probability that R is not allowed to retransmit one more time after it failed previously;

    β - the probability that S is not allowed to retransmit one more after R failed and it is not allowed to retransmit one more time.
  - We can associate the simplified transition matrix Z with a FSM and a protocol.

The probabilistic protocol: FSM at the transmitter

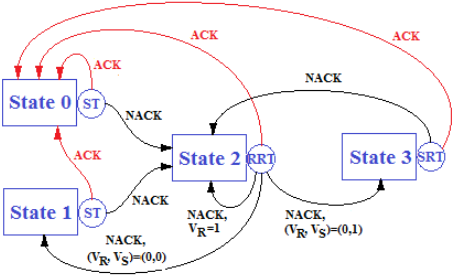
- Definition of the probabilistic protocol:
  - The protocol starts either from State 0 or from State 1;
  - If NACK from D: the first retransmission comes from R;

- If R is retransmitting, the next action is chosen by realization of two random parameters V_S and V_R:

  R retransmits with probability $(1-\gamma)$;

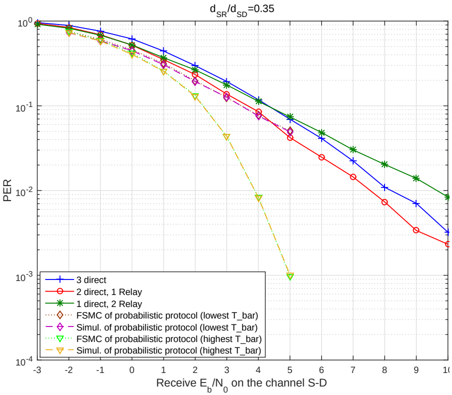
  S retransmits with probability $(\gamma(1-\beta))$;

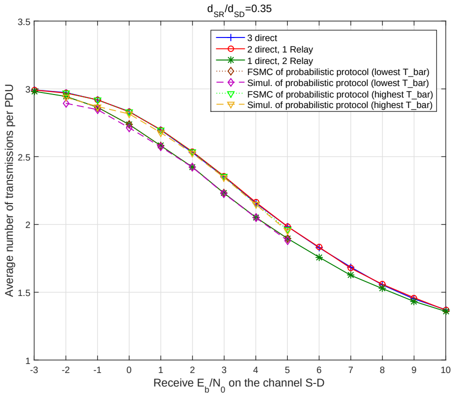
  Neither S or R are allowed to retransmit, with probability $\gamma \cdot \beta$. The PDU is lost.



Comparison with a referent protocol, type II decoder

- Comparison with a referent deterministic protocol:
  - Comparison in PER and $\overline{T}$;





In summary

- HARQ is "yet another" way of adapting the communication protocol to the actual channel values, therefore ...
  - the compatibility with other ingredients of the protocol has to be checked
  - and some adaptation has to be implemented
- but these adaptations also open new possibilities, with improved performance...
- Clearly, non orthogonal superposition instead of orthogonal retransmission brings a lot of improvements...