HARQ
ARQ versus FEC

• ARQ with CRC, but without FEC
  – Re-Tx only after failure of previous attempts
  – No error correction possible w/o Re-Tx

• FEC
  – Like implicit Re-Tx at the same time
  – Diversity in terms of ‘repetition’ & ‘redundancy’
  – Because of diversity, some errors may be corrected
Hybrid ARQ Type 1

• Hybrid of FEC and ARQ
• Single TX has FEC encoded packet that can correct some errors, besides detecting them
• If FEC cannot correct errors, then ARQ Re-Tx attempts may help with RX
• E.g. Rate 1/3 FEC coded packet transmission that uses ARQ as well
ARQ versus HARQ Type 1

- **Good channel conditions**
  - ARQ w/o FEC is efficient
  - Small number of erroneous pkts corrected with ARQ efficiently
  - HARQ type 1 => unnecessary FEC overhead

- **Bad channel conditions**
  - HARQ 1 is efficient as FEC diversity helps with error correction
  - Just ARQ => Numerous latency laden ReTx with extra ARQ status report (ACK NACK) overhead
HARQ Type 2: Efficient for Good & Bad Channels (1/2)

• TX attempts for a packet
  – Alternate between message bits with CRC and only FEC parity bits with CRC
  – If first transmission = error free, the FEC parity bits are never sent
  – Consecutive transmissions can be combined for error correction if neither is error free

• First TX does not have enough FEC info
  – If channel good, it works w/o overhead

• HARQ ACK/NACK typically STOP-n-WAIT
  – Immediate quick ACK/NACK feedback done
  – No new TX until current packet is successful (or maximum HARQ retries done)
HARQ Type 2: Efficient for Good & Bad Channels (2/2)

- RE-TX typically sends different set of bits than the previous TX => improving FEC ability when combined with the previous TX info
  - In bad channel conditions, ReTx helps with reception, with overhead only as ReTx happen
  - Combined data from multiple attempts better than each independent attempt
- Also, typically implemented in hardware instead of software
HARQ With Soft Combining

• HARQ with Soft Combining (most common)
  – Do not discard data from a previous attempt received in error (detected but not corrected)
  – Combine the packet received in previous transmission with that of later transmissions to decode the overall packet

• 2 Types
  – Chase Combining
  – Incremental Redundancy

Read more: http://en.wikipedia.org/wiki/Hybrid_automatic_repeat_request
Chase Combining

- Every retransmission = the same information (data and parity bits)
- Receiver uses **maximum-ratio combining** to combine the received bits with the same bits from previous transmissions
- All transmissions are identical => Chase combining seen as additional repetition coding
- Every retransmission adds extra energy to the received transmission for an increased $E_b/N_0$
Incremental Redundancy (IR)

- Multiple sets of coded bits each representing the same set of info bits.
- ReTx uses a different set of coded bits than the previous transmission, with different redundancy versions generated by puncturing the decoder output.
- Thus, every retransmission contains different information than the previous one.
- At every ReTx, the receiver gets additional info.
HARQ type 3, or Partial IR

- Add additional redundancy in each ReTx, but each ReTx attempt is independently decodable to the original packet
  - i.e. even without combining with other TXs
- Thus, ReTx packet can be chase combined with previous attempts as well
- LTE uses this type of HARQ
  - Even if the original or another attempt is missed completely, a single ReTx attempt can be independently decoded
Redundancy Versions (RVs) in LTE

- LTE HARQ has 4 RVs typically of a packet
  - 4 (re)transmissions packet bits
  - RV = Different combinations of systematic data bits + FEC bits
  - These transmitted in retransmissions and then cycled through again until
    - Correct reception; or
    - Max retries allowed are over => give up, and let ARQ in RLC to correct the loss

- LTE max retries = typically 4
LTE HARQ Processes

• A process is an independent invocation of the HARQ flow / channel
• Each transmission happens in a TTI or Transmit time interval = 1 subframe
• Stop-n-Wait is employed per process, but
  – It takes 8ms for a round trip
    • 4ms to get ACK/NACK, and another 4 to ReTx
  – So, use 8 HARQ processes to keep using all the sub-frames
DL TX and HARQ

• Asynchronous adaptive HARQ
  – processes can be transmitted in any order without fixed timing
• Uplink ACK/NAKs in response to downlink (re)transmissions are sent on PUCCH or PUSCH
• PDCCH signals the HARQ process number and if it is a transmission or retransmission
• Retransmissions are always scheduled through PDCCH
DL Asynchronous HARQ

**Figure 2**

From: [http://mobiledevdesign.com/standards_regulations/0918DSagil-Figure02.jpg](http://mobiledevdesign.com/standards_regulations/0918DSagil-Figure02.jpg)
UL TX and HARQ (1/2)

- Synchronous HARQ
  - Each uplink HARQ process is assigned to a specific subframe
- Maximum number of retransmissions configured per UE (as opposed to per radio bearer)
- Downlink ACK/NAKs in response to uplink (re)transmissions are sent on PHICH
- HARQ operation principles
  - Regardless of the content of the HARQ feedback (ACK or NACK), when a PDCCH for the UE is correctly received, the UE follows what the PDCCH asks the UE to do i.e. perform a transmission or a retransmission (referred to as adaptive retransmission)
UL TX and HARQ (2/2)

- HARQ principles continued
  - When no PDCCH addressed to the C-RNTI of the UE is detected, the HARQ feedback dictates how the UE performs retransmissions:
    - NACK: the UE performs a non-adaptive retransmission i.e. a retransmission on the same uplink resource as previously used by the same process
    - ACK: the UE does not perform any UL (re)transmission and keeps the data in the HARQ buffer. A PDCCH is then required to perform a retransmission i.e. a non-adaptive retransmission cannot follow.
UL Synchronous HARQ

Figure 1

Src: http://mobiledevdesign.com/standards_regulations/0918DSagil-Figure01.jpg
UL HARQ Operation with PDCCH

<table>
<thead>
<tr>
<th>HARQ feedback seen by the UE</th>
<th>PDCCH seen by the UE</th>
<th>UE behaviour</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK or NACK</td>
<td>New Transmission</td>
<td>New transmission according to PDCCH</td>
</tr>
<tr>
<td>ACK or NACK</td>
<td>Retransmission</td>
<td>Retransmission according to PDCCH (adaptive retransmission)</td>
</tr>
<tr>
<td>ACK</td>
<td>None</td>
<td>No (re)transmission, keep data in HARQ buffer and a PDCCH is required to resume retransmissions</td>
</tr>
<tr>
<td>NACK</td>
<td>None</td>
<td>Non-adaptive retransmission</td>
</tr>
</tbody>
</table>

PDCCH always overrides HARQ ACK-NACKs
Why UL Synchronous & DL Asynchronous

- PDCCH transmission of allocation info versus the actual allocated resource blocks
  - In DL: same sub-frame
  - In UL: 4 sub-frames into future
- eNB is the scheduler doing PDCCH allocations => DL anytime scheduling
- UL Synchronous HARQ is simpler for UE that are not as computation intensive as eNBs
Round Trip Time in HARQ

• ACK/NACK sent in 4\textsuperscript{th} subframe after the TX
• ReTx or New TX in the 4\textsuperscript{th} subframe after the NACK
• Total Round Trip Time = 8ms
• However, 8 HARQ processes can be going on simultaneously to utilize all subframes
ARQ Principles in RLC

- ARQ retransmits RLC PDUs or RLC PDU segments based on RLC status reports
  - Reports from Receiver to Sender
- Polling for RLC status report is used when needed by RLC
  - Sender polls
- RLC receiver can also trigger RLC status report after detecting a missing RLC PDU or RLC PDU segment
  - Receiver sends status reports w/o a poll
ARQ HARQ ACK Differences

• HARQ ACK/NACKs
  – Typically not packets but “Signals”
  – These signals have to be designed into lower MAC-PHY; e.g. channels
  – Less overhead than MAC packets
  – Sent for every TX and ReTx

• ARQ ACK/NACK/Status Reports
  – Typically packets with preamble, header, CRC etc.
  – Thus, higher overhead than HARQ ACK-NACK “signals”
  – And sent less often, as typically sent for a “block” of packets
HARQ-ARQ Error Rates Example

• HARQ operates at 10% packet error rate
  – Assume every HARQ TX/ReTX causes successful RX with probability 0.9
• If 4 total transmissions (3 Retransmissions) allowed max
• What is packet error rate seen above the HARQ (by ARQ)
• Simple Solution
  – Error Prob. => all 4 HARQ TX failed
  – $0.1^4 = 1e-4$
Should HARQ talk to ARQ Locally?

- **At Receiver**
  - Packet RX successful
    - ARQ gets the packet anyway, so no need
  - Packet RX failure after max retries
    - No knowledge for ARQ about what is contained in failed packet

- **At Transmitter**
  - Packet TX successful
    - ARQ can be pre-notified of success, but
    - ARQ status report should be relied upon instead
  - Packet TX failure after max retries
    - **ARQ can be told exactly what packet/s failed, so ARQ can initiate ReTX even without status reports or ACK timeouts**
MAC Scheduling & Muxing

Besides HARQ, the MAC sub-layer provides Scheduling, Priority-handling, and Multiplexing Services
UE MAC Scheduling

• Allocations from eNB are to UEs, and not to specific logical channels
• MAC at UE decides how logical channels shall share the allocation
• MAC to RLC
  – Send me Bi bits from logical channel i : for i =1 to k
  – Thus MAC decides Muxing and Sharing of resource
  – MAC decision based on
    • QoS priorities
    • Knowledge of backlog in each logical channel
• RLC to MAC
  – Backlog info about each logical channel
  – Bi bits from each ith logical channel as requested by MAC
RLC-MAC Interactions at UE

RLC

Buffer Status in different channels

Data request: Bi bits from ith channel

MAC

Different channel data as requested, to be muxed by the MAC
MAC Scheduling at eNB (1/2)

• Significantly more involved than MAC scheduling at UE
  – Both UL and DL scheduling across all UEs
  – taking into account all flows’ QoS requirements & Backlogs

• 2 Criteria for scheduling
  – QoS needs of various logical channels *across all UEs*
    • High priority traffic gets prioritized access
    • Share resources equally among identical priority traffic
  – Reception Quality at each UE
    • Schedule UEs with better channel conditions, as more chance of success
    • Schedule UEs in sub-carriers with good conditions
    • Wait for channel conditions to get better at certain UEs at certain times
MAC Scheduling at eNB (2/2)

• eNB knowledge of QoS requirements & Backlog
  – Through UE buffer status reports

• eNB knowledge of UL and DL channel conditions
  – CQI reports by UEs about DL channel conditions
  – UL sounding reference signals by UE let eNB estimate channel conditions in various carriers for that UE
Summary

• LTE MAC performs
  – HARQ
  – Scheduling, priority handling, and muxing