An Efficient ARQ Mechanism i Multi-hop Relay Systems based on IEEE 802.16 OFDMA

VTC 2007 Fall
October, 2007

林咨銘
2008/06/19
tmlin@itri.org.tw

Outline

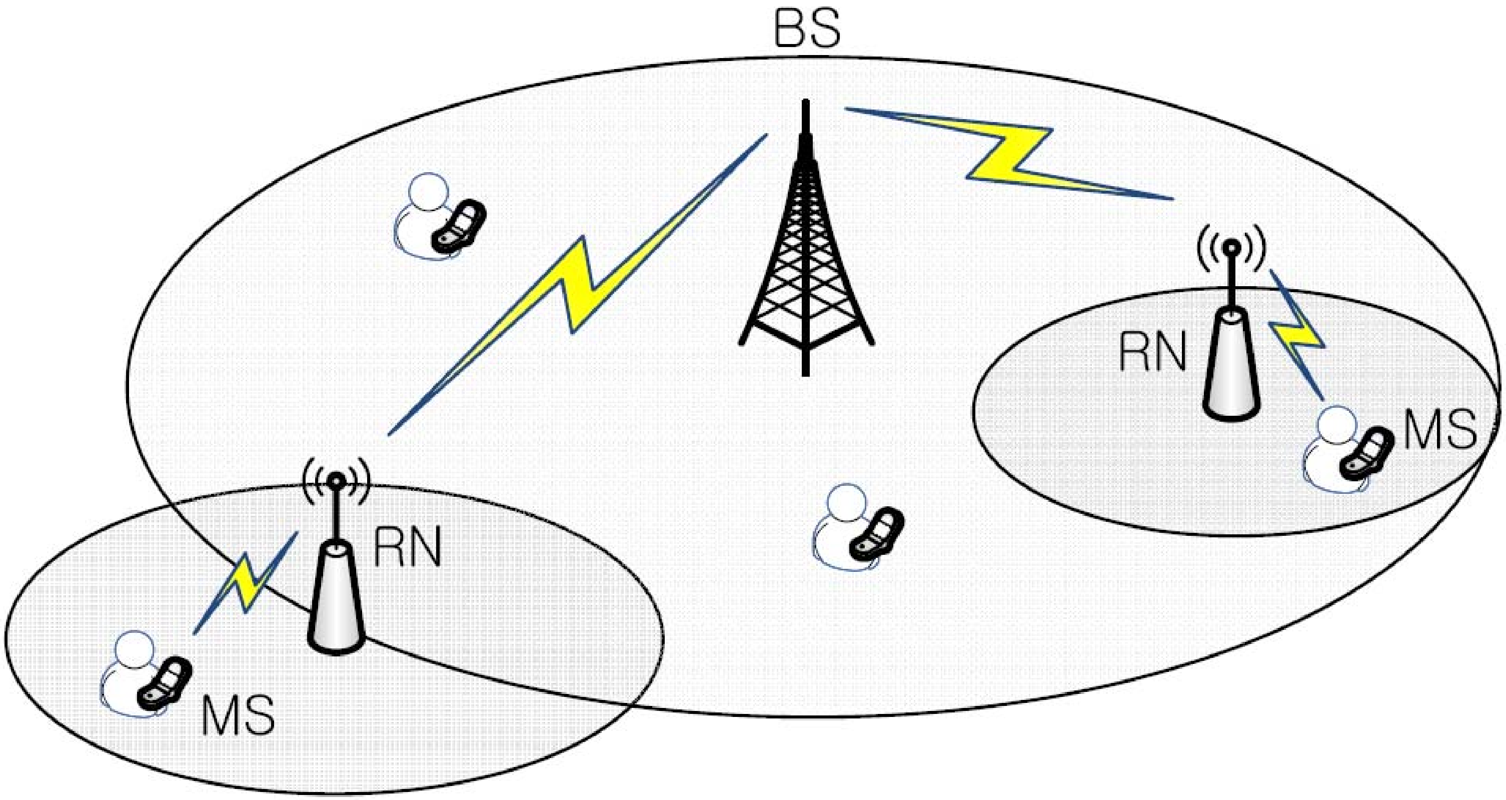
- Introduction
- Related Works
  - IEEE 802.16 ARQ
  - Relay ARQ
- Proposed ARQ Mechanism for MMR System
- Numerical Analysis
  - Delay
  - ARQ Transmission Efficiency
- Simulation Result
- Conclusions and Issues

Introduction

- Multi-hop relay has been a key technique for improving cell coverage and throughput
  - Reduce cost of deployment and operation for carrier also
- IEEE 802.16j was created to realize the first relay system

- Advantage of multi-hop relay
  - Enhanced cell throughput and coverage
    - Reduce loss between BS and MS
  - Conserve the power of MSs
    - Short communication distance

Introduction



Introduction

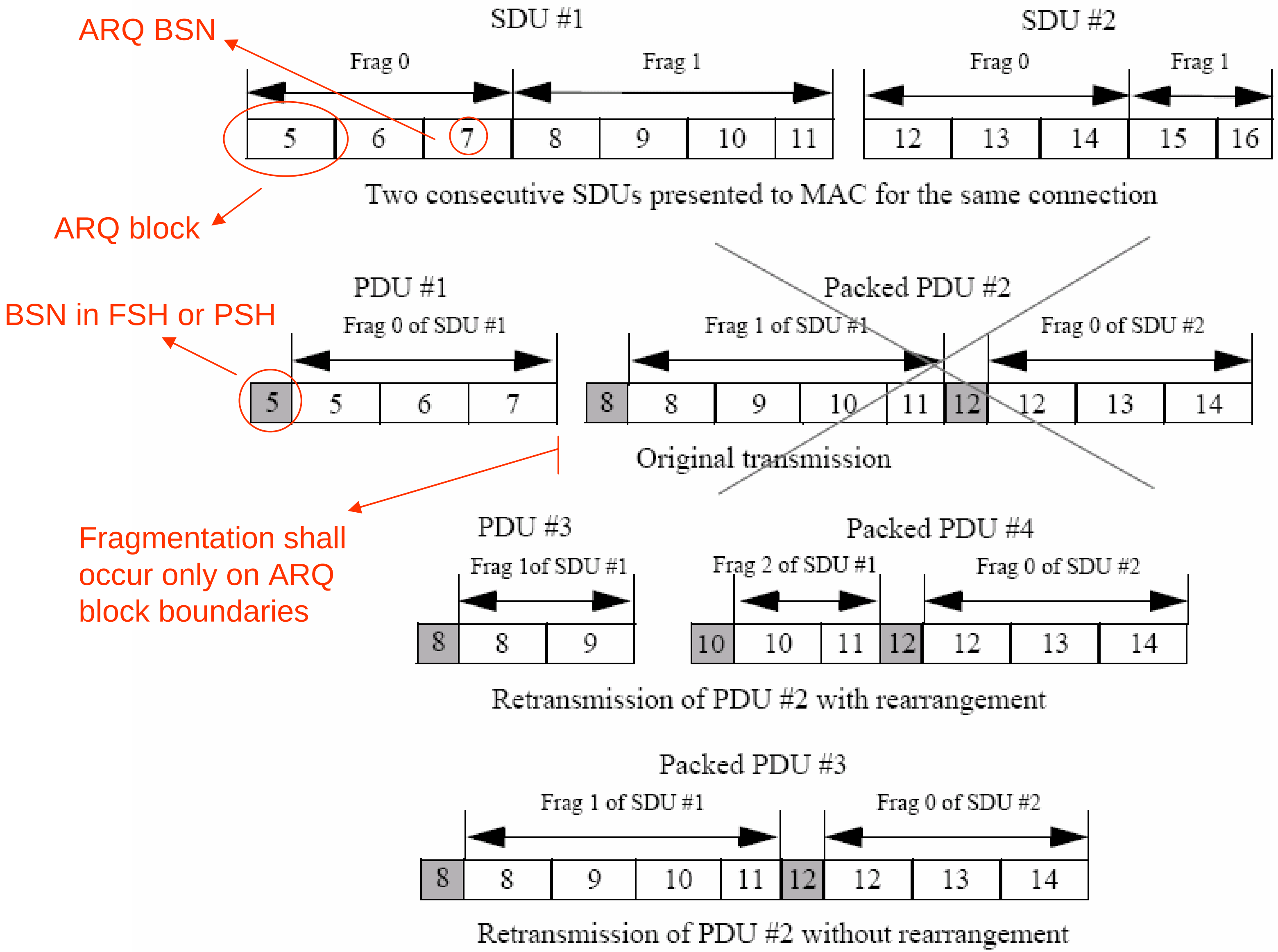
- Background
  - Conventional ARQ mechanisms are optimized for single hop transmission

- Problems
  - Additional radio resources are used
  - Complexity of system is increased
  - Reliability is reduced
    - Data is transmitted through many wireless links

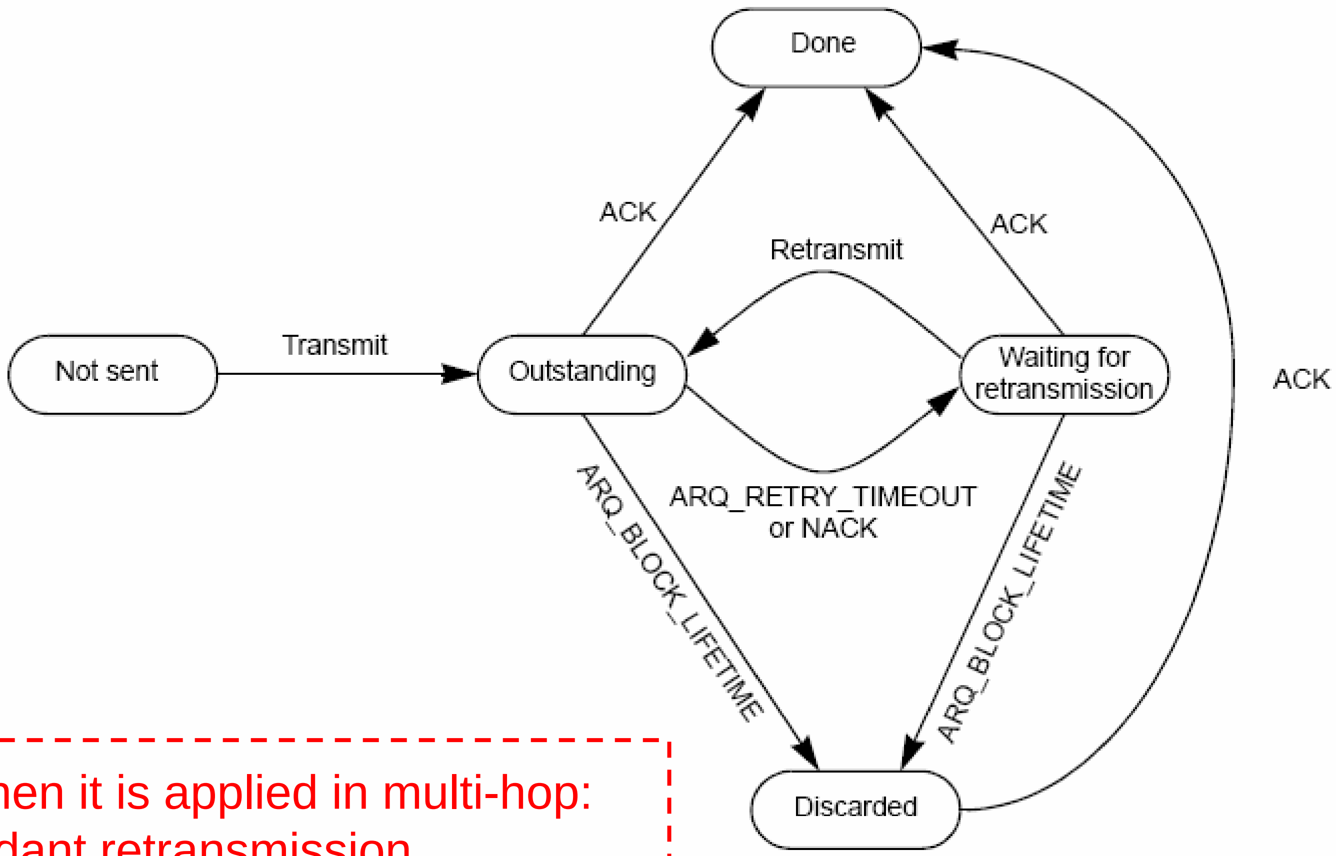
IEEE 802.16 ARQ

- ARQ mechanism
  - May be enabled on a per-connection basis
    - ARQ parameters are negotiated during connection creation
    - Enabling fragmentation is optional
  - Fragmentation shall occur only on ARQ block boundaries
  - If ARQ is enabled at the connection, Fragmentation and Packing subheaders contain a ***BSN*** (Block Sequence Number)
    - The number of the first ARQ block in the sequence of blocks

IEEE 802.16 ARQ

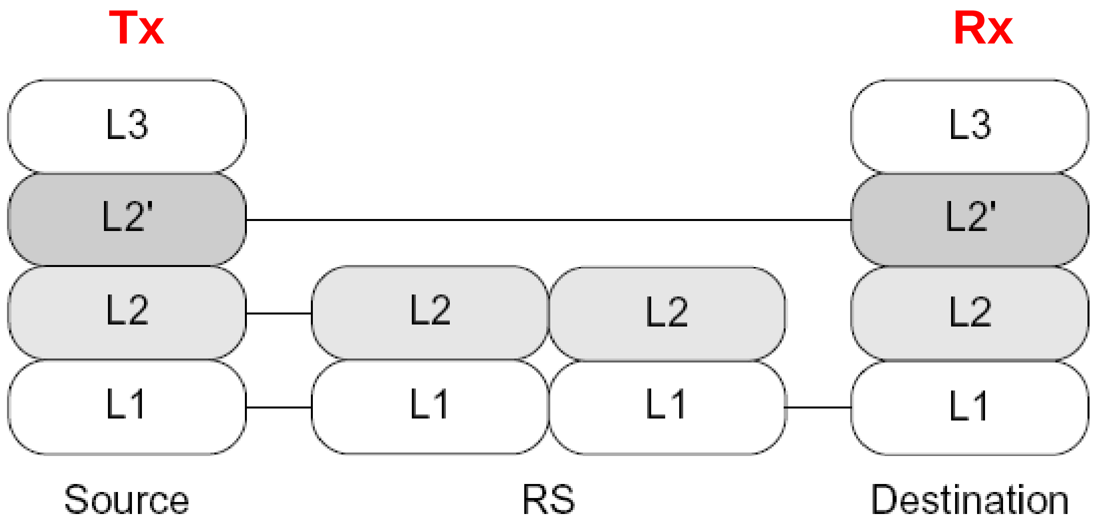


IEEE 802.16 ARQ



Issues when it is applied in multi-hop:
1. Redundant retransmission
2. Long delay
3. ARQ state management is complex
4. ARQ state exchange during handover

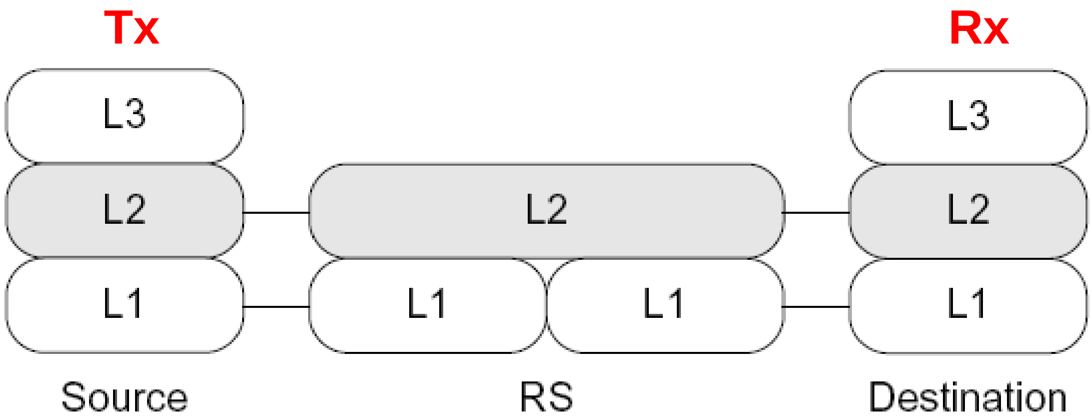
Relay ARQ - 1



- Lower layer (L2) : each hop
- Upper layer (L'2): end-to-end multi-hop

- Simple
- Additional overhead
  - Header and control information of both layers
- Backward compatibility

Relay ARQ - 2

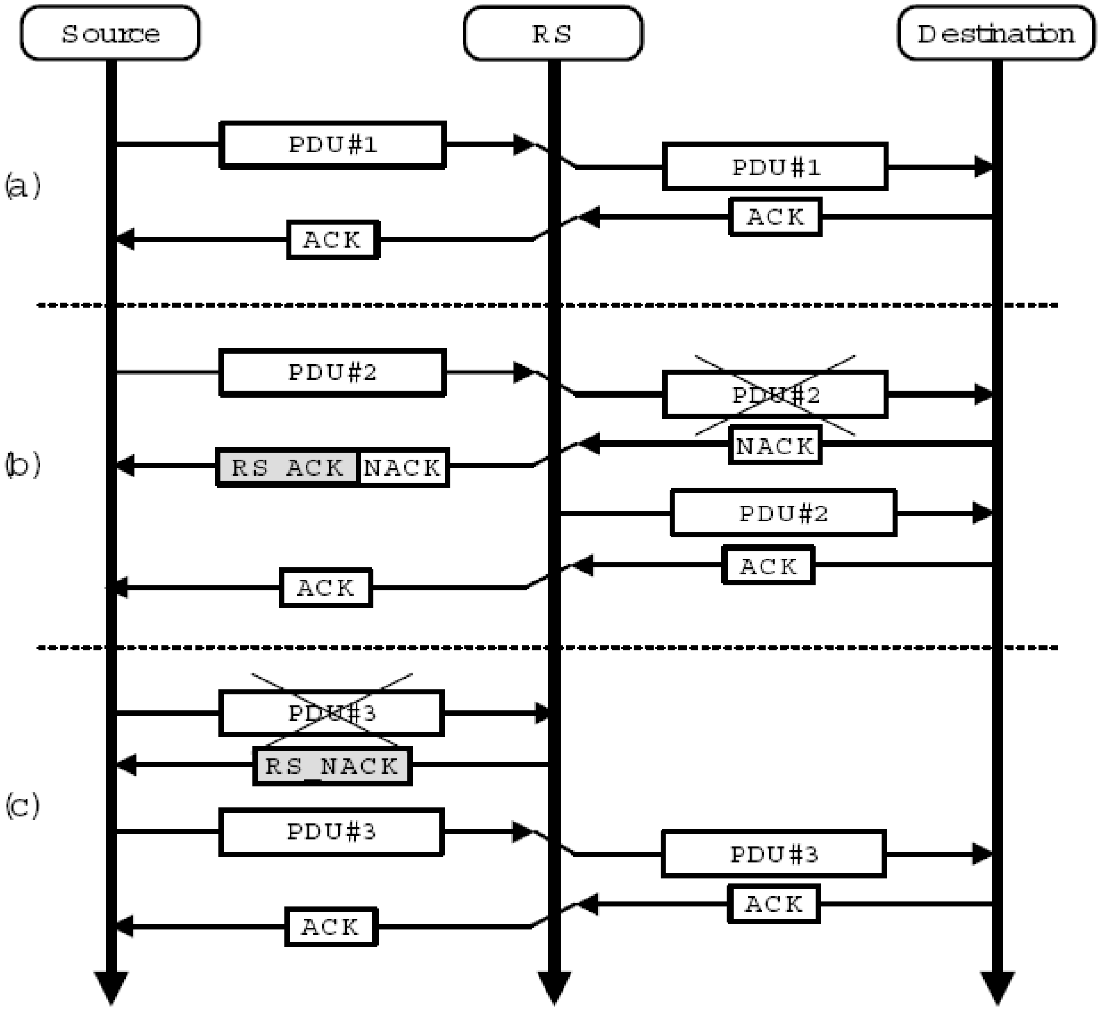


- Link layer version of snoop TCP
  - A snoop agent located between Tx and Rx
- New feedback message – RS_ACK
  - Tx delegates the data transmission responsibility to RS if RACK is received

- Prevent redundant retransmissions
- Additional overhead due to RS_ACK transmission
- Long retransmission delay
  - Retransmission is triggered by the feedback message from Rx or packet timeout

Proposed ARQ Mechanism for MMR System

- RS sends RS_ACK to Tx
  - To prevent redundant retransmission
  - Only when RS receives data successfully
- Introducing RS_NACK
  - Request for fast retransmission
- Tx classifies the ARQ feedback messages by the CID of Rx or RS in the packet header



Proposed ARQ Mechanism for MMR System

- RS feedback message is not always sent

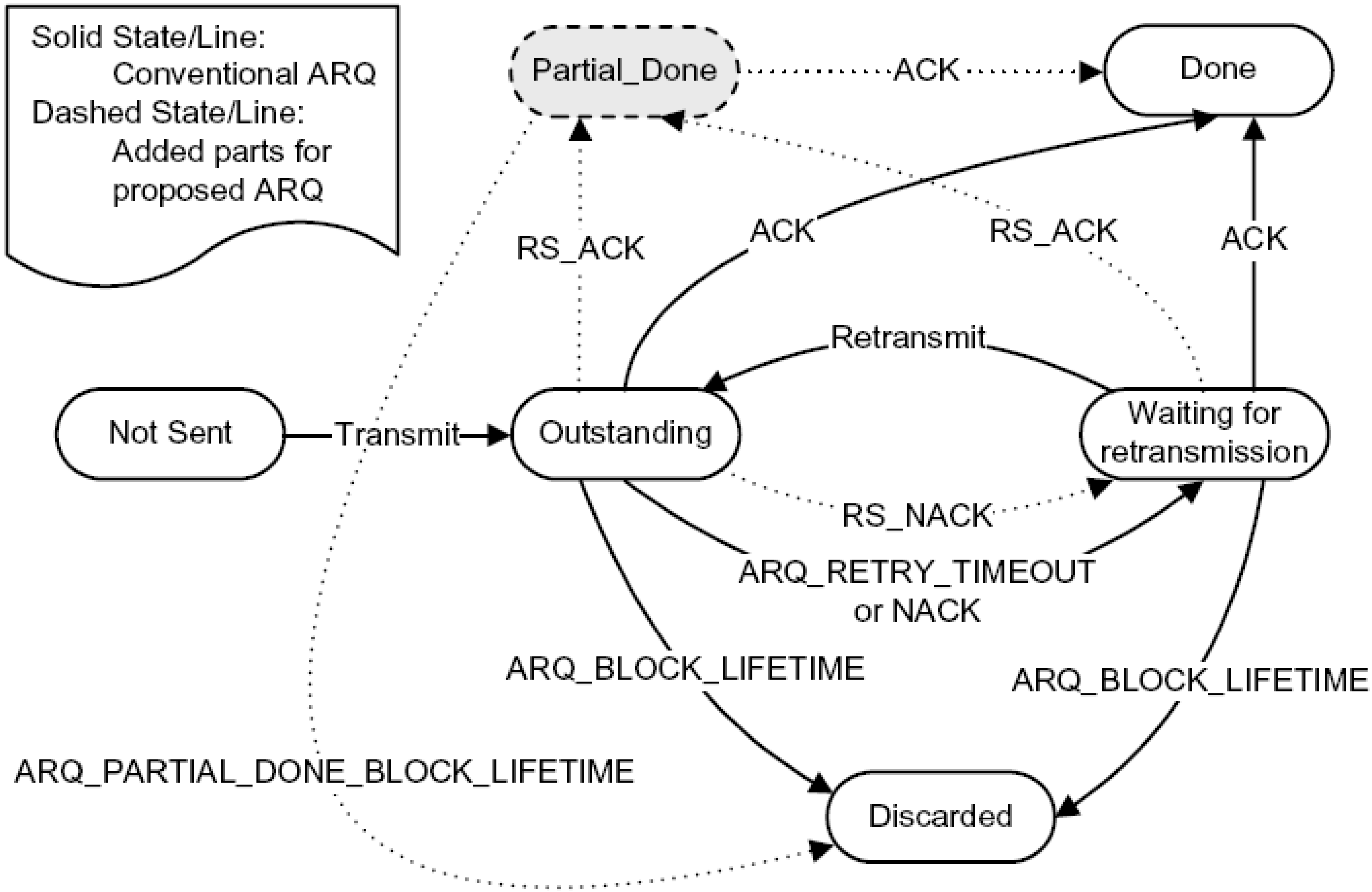
  =>Reduce overhead

- The ARQ feedback message made by RS is sent to Tx only at two cases
  - RS fails to receive the data from Tx
    - RS_NACK can trigger the fast retransmission of the data which RS fails to receive

    =>Reduce transmission delay
  - RS receives the NACK made by Rx for the data which the RS already received successfully
    - Prevent Tx from retransmitting the data that RS had received successfully

    =>Save radio resource

Proposed ARQ Mechanism for MMR System



Numerical Analysis - Delay

D_1 : Average delay in the RS receiving the data from Tx successfully
D_2 : Average delay in the Rx receiving the data from RS successfully
P_1 : Packet Error Rate between Tx and RS
P_2 : Packet Error Rate between RS and Rx
P_f : Probability that retransmission is triggered by RS (RS_NACK)
R_T : *ARQ_RETRY_TIMEOUT* period
T : Frame duration

Delay for RS receiving data from Tx in (i+1)th transmission

$$D_1 = \sum_{i=0}^{\infty} (T + 2i \times T \times P_f + R_T \times (1 - P_f)) \times P_1^i(1 - P_1)$$

Probability that RS receiving data from Tx in (i+1)th transmission

Assume $P_f = 1$

$$D_1 = \sum_{i=0}^{\infty} (1 + 2i) \times T \times P_1^i(1 - P_1) = \frac{1 + P_1}{1 - P_1} \cdot T.$$

Delay for RS receiving ACK from Rx in (i+1)th transmission

$$D_t = \sum_{i=0}^{\infty} (2T + 2i \times T \times P_f + R_T \times (1 - P_f)) \times P_2^i(1 - P_2) + D_1 + T$$

Proposed ARQ

$$D_t = \sum_{i=0}^{\infty} (1 + i) \times 2T \times P_2^i(1 - P_2) + D_1 + T = \frac{4 - 2(P_1 + P_2)}{(1 - P_1)(1 - P_2)} \cdot T$$

$$D_t = \frac{4}{(1 - P_1)(1 - P_2)} \cdot T$$

Conventional ARQ

Numerical Analysis – Transmission Efficiency

ARQ transmission efficiency

$$\eta_r = \frac{n_p/t_p(1+N_r)}{n_p/t_p} = \frac{1}{1+N_r}$$

N_r : Average number of retransmission per packet
t_p : transmission of a packet
n_p : total number of bits in a packet
P : PER of a link

$$N_r = \sum_{i=1}^{\infty} i(1-P)P^i = \frac{P}{1-P} \Rightarrow \eta_r = 1-P.$$

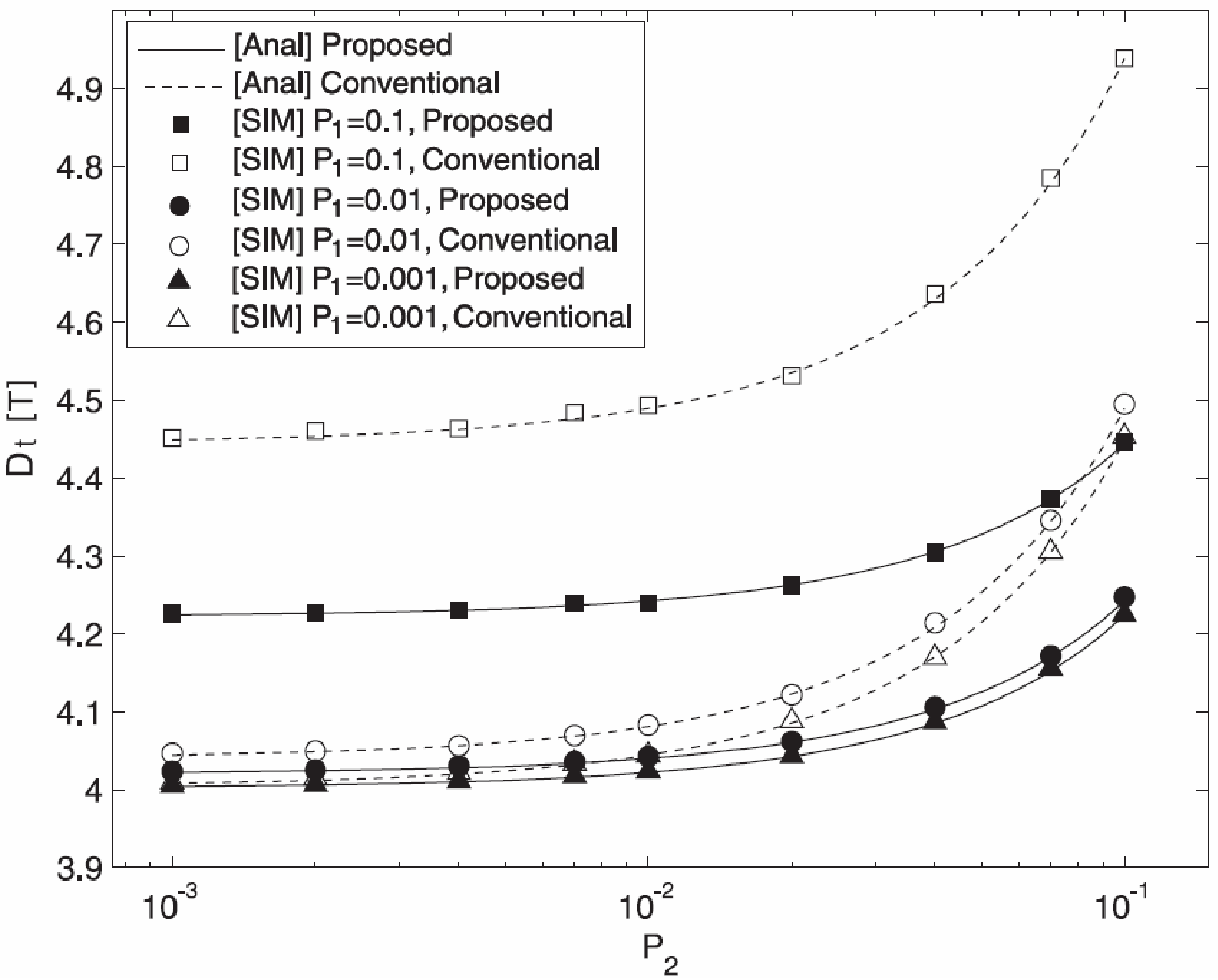
Proposed ARQ

$$\eta_r = min(1-P_1, 1-P_2)$$

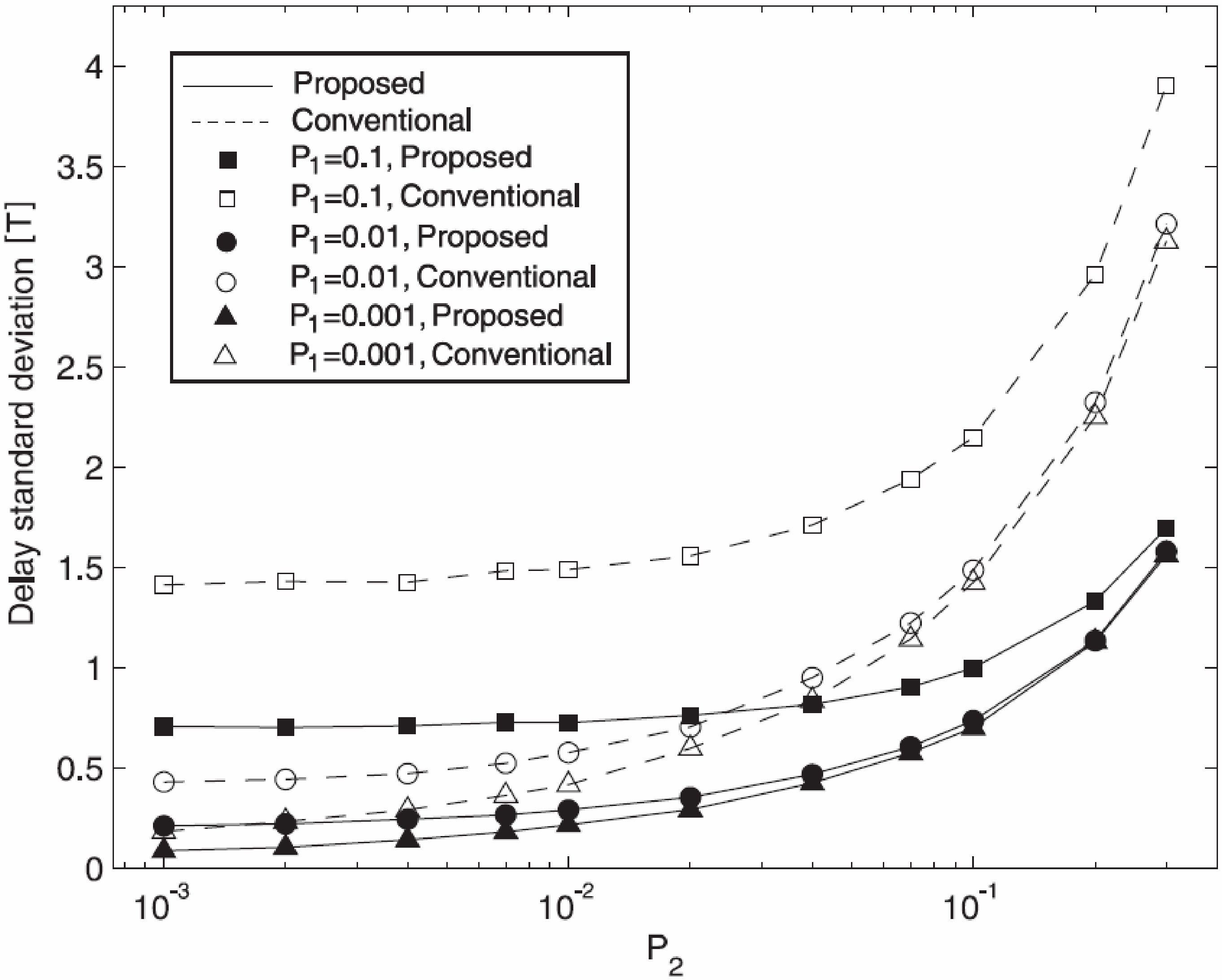
Conventional ARQ

$$\eta_r = (1-P_1)(1-P_2)$$

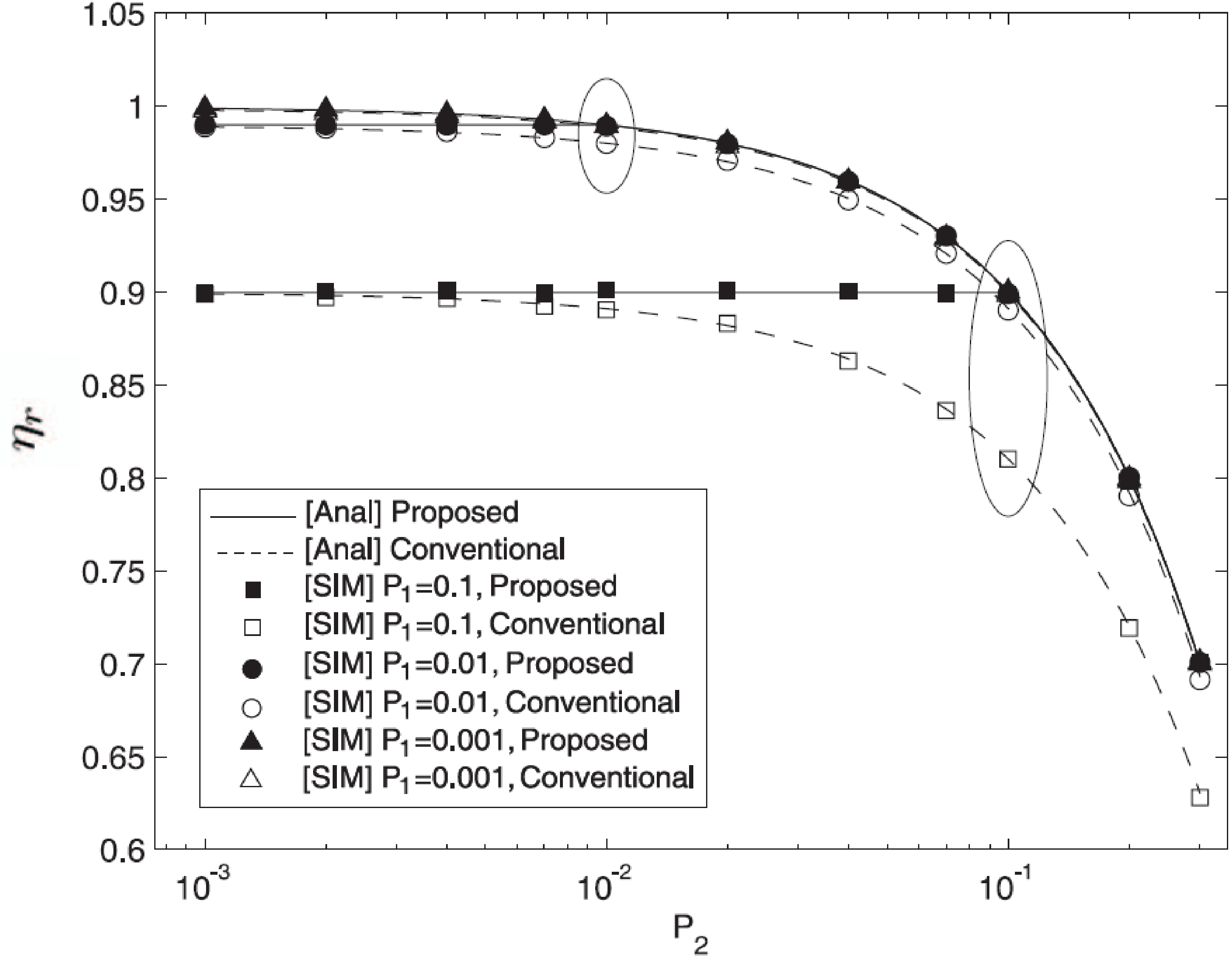
Simulation Result – Average Transmission Delay



Simulation Result – Average Transmission Delay



Simulation Result – Transmission Efficiency



Conclusions

- Efficient ARQ is proposed for MMR system
  - Introduce ARQ feedback message made from RS
  - Retransmit data quickly
  - Avoid meaningless retransmission
- Analysis and simulation shows better performance of proposed ARQ
  - Delay
  - Transmission Efficiency

Issues

- How does the proposed ARQ applied in multi-hop cases?
  - Only two hop cases are shown
- Overheads may not be saved
  - Header
    - RS_ACK/RS_NACK is sent out in different message
  - Message size
    - ARQ mechanism uses ACK MAPs to indicate which blocks are received without error
    - Additional MAPs for RS_ACK and RS_NACK are required
    - All RS_ACK and RS_NACK shall be report to Tx