Exploiting Multiuser Diversity Using Multiple Feedback Thresholds

V. Hassel[1], H. Koubaa[1], M.-S. Alouini[2], D. Gesbert[3] and G. E. Øien[1]

[1] Dept. of Electronics and Telecommunications
Norwegian University of Science and Technology, Norway
email: {vegard,oien}@iet.ntnu.no
[2] Dept. of Electrical and Computer Engineering
University of Minnesota, USA
email: alouini@ece.umn.edu
[3] Dept. of Mobile Communications
Eurecom Institute, France
email: gesbert@eurecom.fr

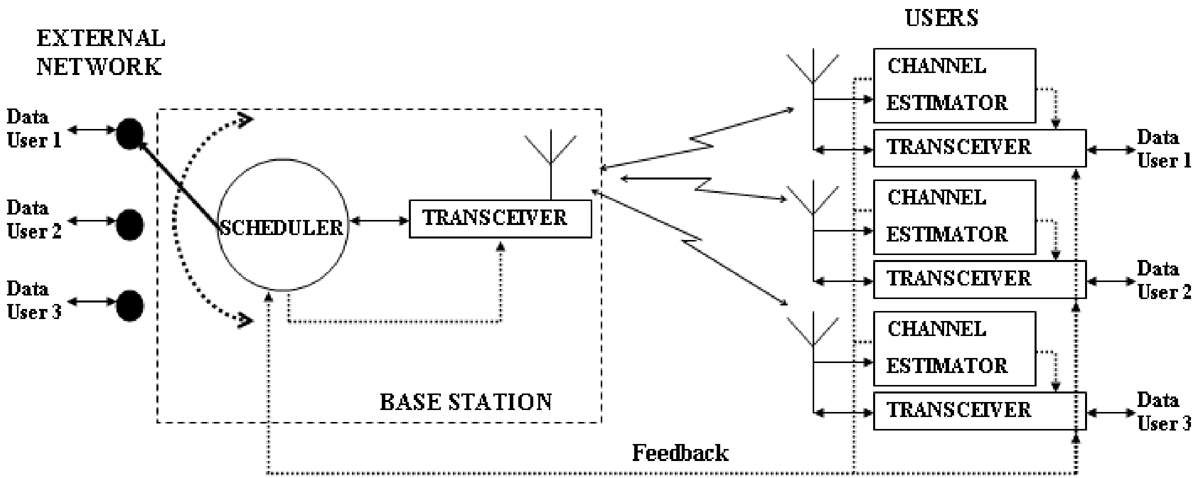
This presentation and the paper can also be found at http://www.tele.ntnu.no/projects/cuban/

Outline

- **Background:**
  - System Model
  - MUD: Multiuser Diversity
  - The Feedback Problem
  - SMUD: Selective Multiuser Diversity
- **Multiple Feedback Thresholds:**
  - Main Idea
  - Why Should This Algorithm be Implemented?
  - How Should This Algorithm be Optimized?
  - Three different implementations for IEEE802.11

System Model

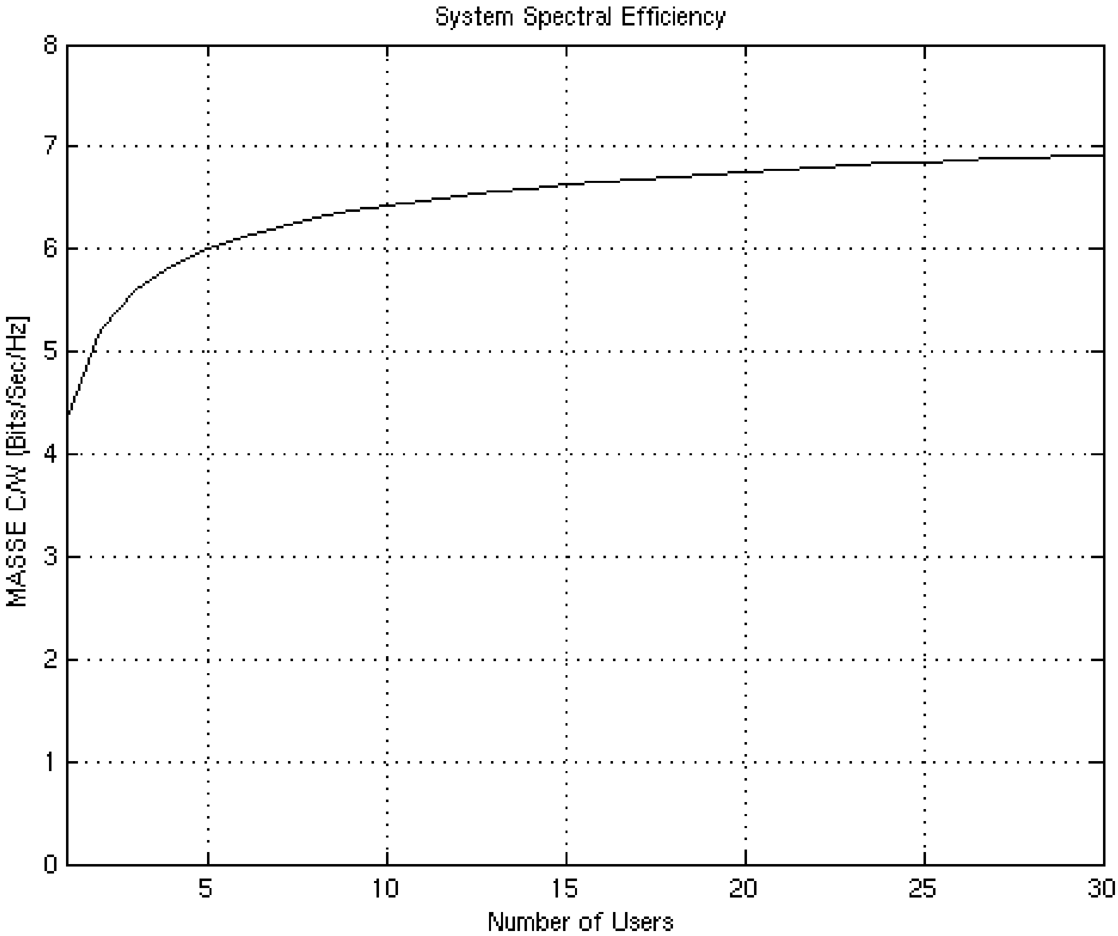
- Time Division Multiplexed (TDM) system with N mobile users
- Carrier-to-noise-ratios (CNRs) of the users' channels are i. i. d.
- The time slots are smaller than a coherence time
- The users have always data to send/receive



NTNU
Department of
Electronics and Telecommunications

Multiuser Diversity (MUD)

- *Diversity* in wireless systems arises because of independently fading channels
- Traditional forms of diversity: space, time, and frequency
- *Multiuser diversity*: With many users in a cell, there is high probability of finding a user with a good channel at any time
- Max CNR Scheduling: To obtain the highest *system* spectral efficiency, the user with the best channel has to be chosen at all times
- Observe: While traditional forms of diversity give better link spectral efficiency, multiuser diversity increases the system spectral efficiency


System Spectral Efficiency
MASSE C/W [Bits/Sec/Hz]
Number of Users
0
1
2
3
4
5
6
7
8
5
10
15
20
25
30

The Feedback Problem

- IDEALLY: We only need feedback from the best user!

- BUT: The users do not know if they are the best

- HENCE: In conventional MUD systems all users need to feed back their CNR values

SMUD: Selective Multiuser Diversity

- Algorithm to reduce the feedback load
- The scheduler asks for the instantaneous CNR level only from the users that have CNR above a threshold
- If none of the users feed back their CNR, a random user is chosen
- The algorithm is not rate-optimal, but gives a significant reduction in the feedback load

Main Idea

- Generalization of the SMUD algorithm:
  We use multiple feedback thresholds
- First, the users are asked if they are above the *highest* threshold value
- If none of the users feed back their CNR value, the base station asks if the users are above the second highest threshold value
- This process continues until one or more users feed back their CNR value
- If the lowest threshold value equals zero, the base station is guaranteed that one or more users feed back their CNR

Why Should This Algorithm be Implemented?

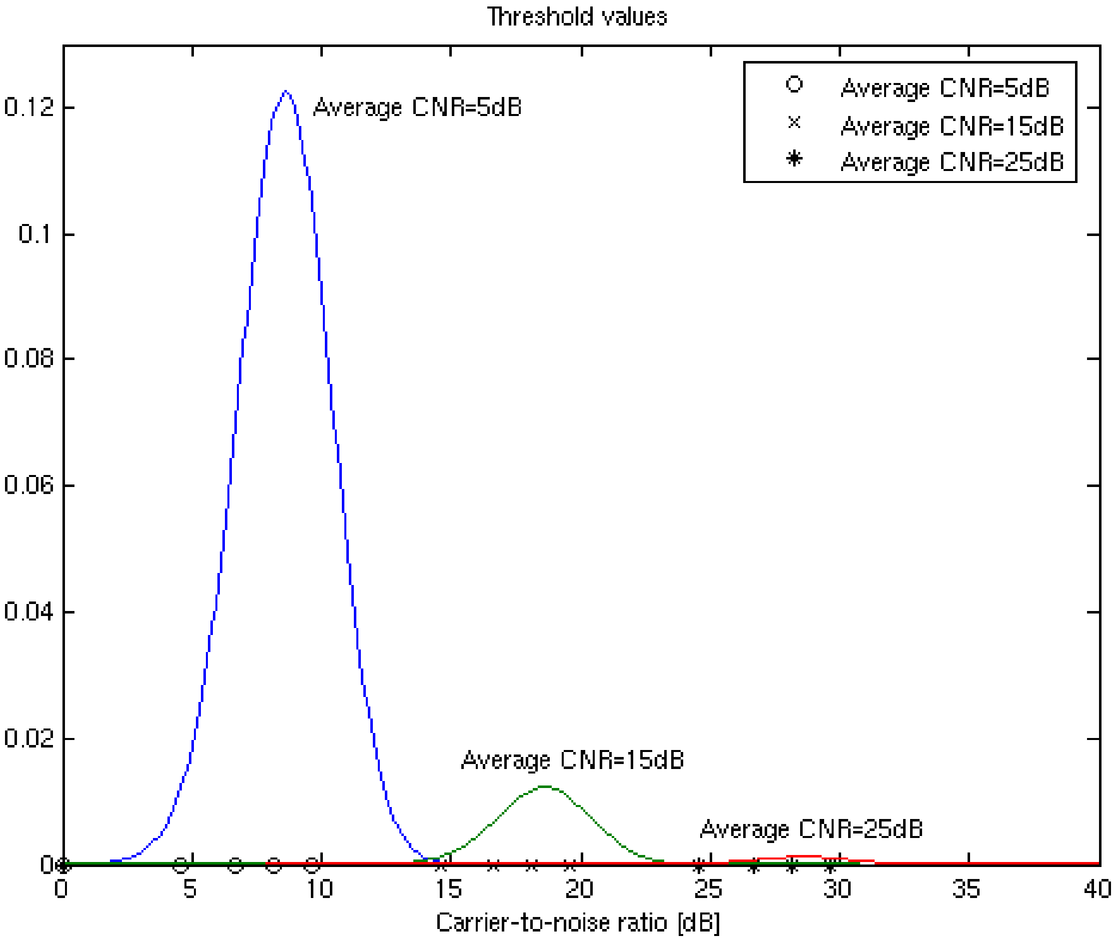
- PROS:
  - The time used to collect feedback is lower than for the full feedback algorithm. This will increase the total system capacity!
  - Reduced feedback from the mobile users will reduce their power consumption
- CONS:
  - Higher system complexity

How Should This Algorithm be Optimized?

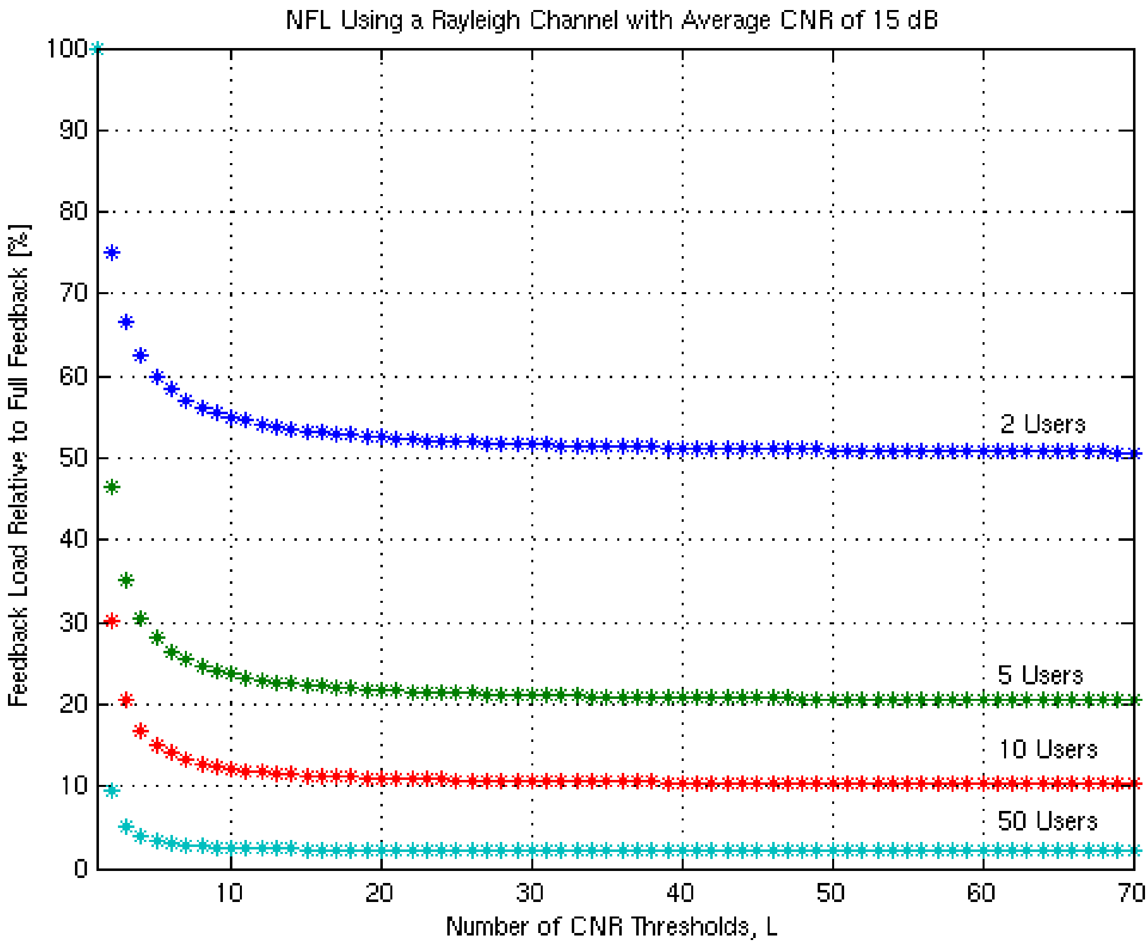
- The feedback collection process should have the following properties:
  - The duration of the process should be as short as possible
  - In order to maximize the MUD gain we should collect the CNR value of the best user
- The time used to collect feedback is a function of:
  - The number of jumps before the successful interval is reached
  - The number of users giving feedback from the successful interval
- Tradeoff between time it takes to look for the successful interval and the time the users use to feed back their CNR values

How Should This Algorithm be Optimized?

- We cannot find sensible threshold values which minimizes the number of jumps
- We therefore choose find the threshold values which minimizes the feed-back load
- The optimization process can now be done in two stages:
  - Find optimal threshold values for a given number of thresholds, L
  - Find optimal number of thresholds, L
- The optimal L can be found for a set of system parameters. The optimum will be where the advantage of introducing a new threshold is less than the disadvantage



NTNU
Department of
Electronics and Telecommunications



NTNU
Department of Electronics and Telecommunications

Implementations for IEEE802.11 Ad Hoc

Assumptions:

- To maximize the MUD gain, a user wants to transmit to the user which has the highest CNR.
- Both feedback and user data will be transmitted on a *contention channel*.
- The collection of feedback will be done within a *guard time*.
- The first part of the guard time consists of a *broadcast message*.
- All users can hear each other.

Implementations for IEEE802.11 Ad Hoc

- The threshold is successively lowered until the *successful interval* is found.
- The time used to switch from receive to send, T_{MS}, is used for each unsuccessful trial.
- If there is only one user within the successful interval, feedback will be transmitted and data transmission can start directly afterwards.
- If two or more users are trying to submit feedback at the same time, we will have a *contention problem*.

Our proposed solution to the contention problem is based on two principles:

- Ranking (indexing) of the users
- Exponential backoff

Ranking of Users

- After the successful interval is found, every user within this interval will have a possibility to transmit feedback.
- The user with the highest rank (index) is allowed to transmit first, then the user with the second lowest rank, etc.
- The time used by a user within the successful interval: T_{FB}=160 μs.
- The time used by a user NOT within the successful interval: T_{MS}=50 μs.
- Can choose to collect feedback from only the first user or from all users within the successful interval.

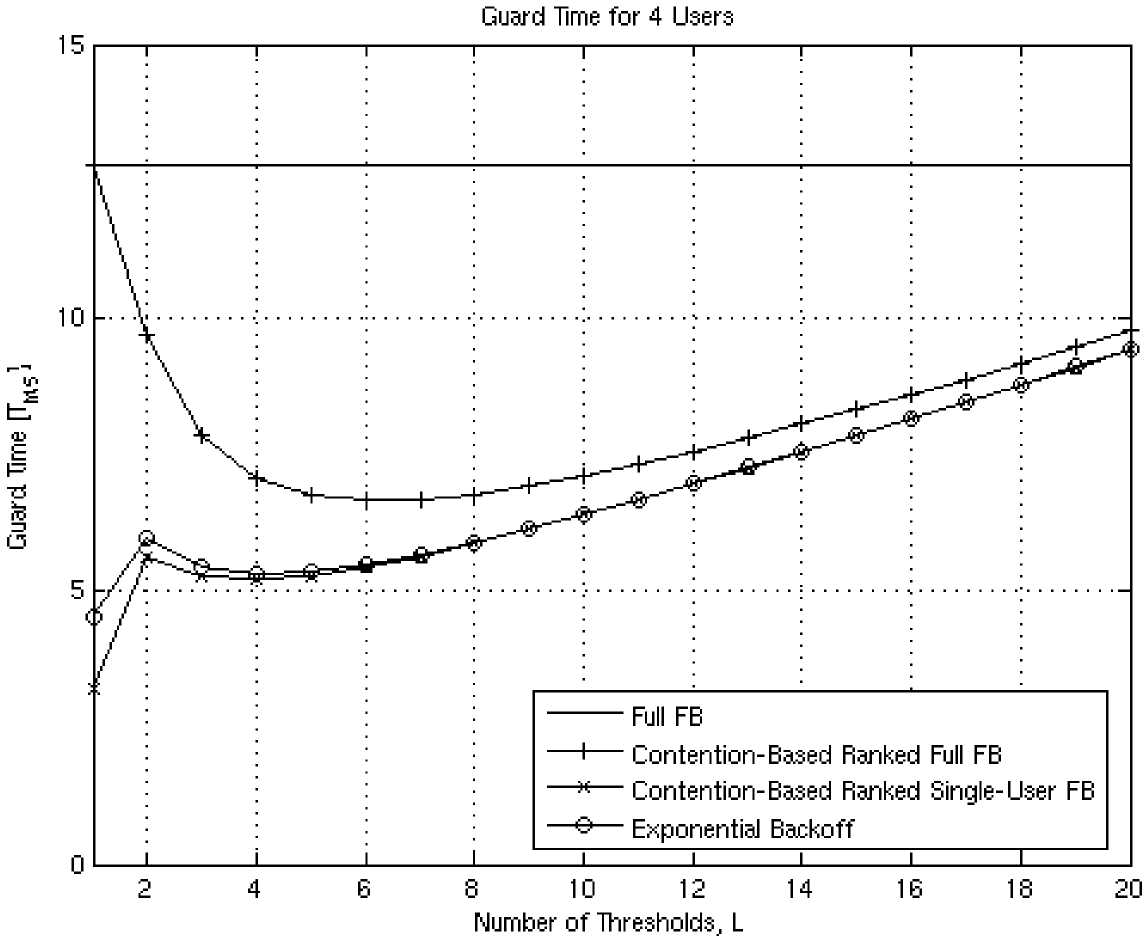
Exponential Backoff

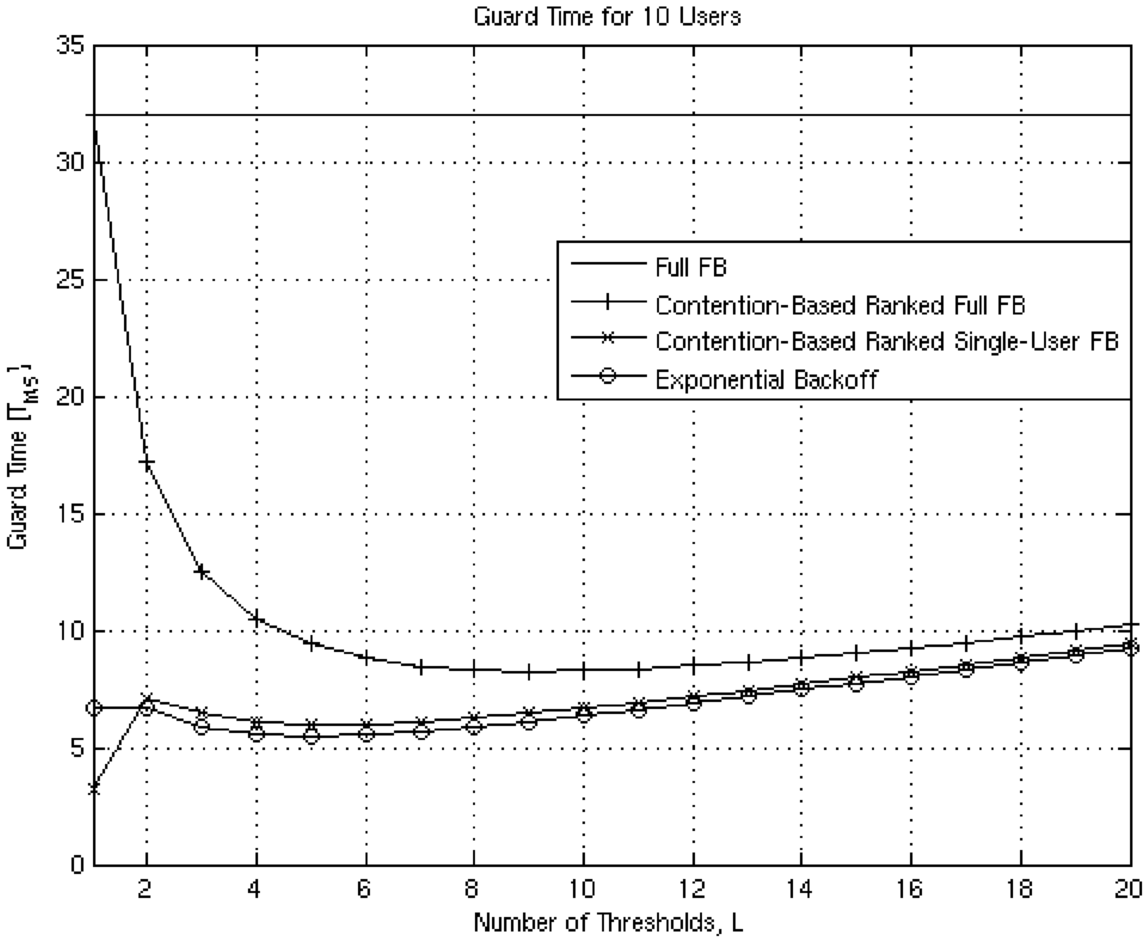
- After the successful interval is found, we half the transmission probability if collisions occur.
- If no collision occur, we will have an unchanged transmission probability.
- Data transmission will start after one user has transmitted feedback successfully.

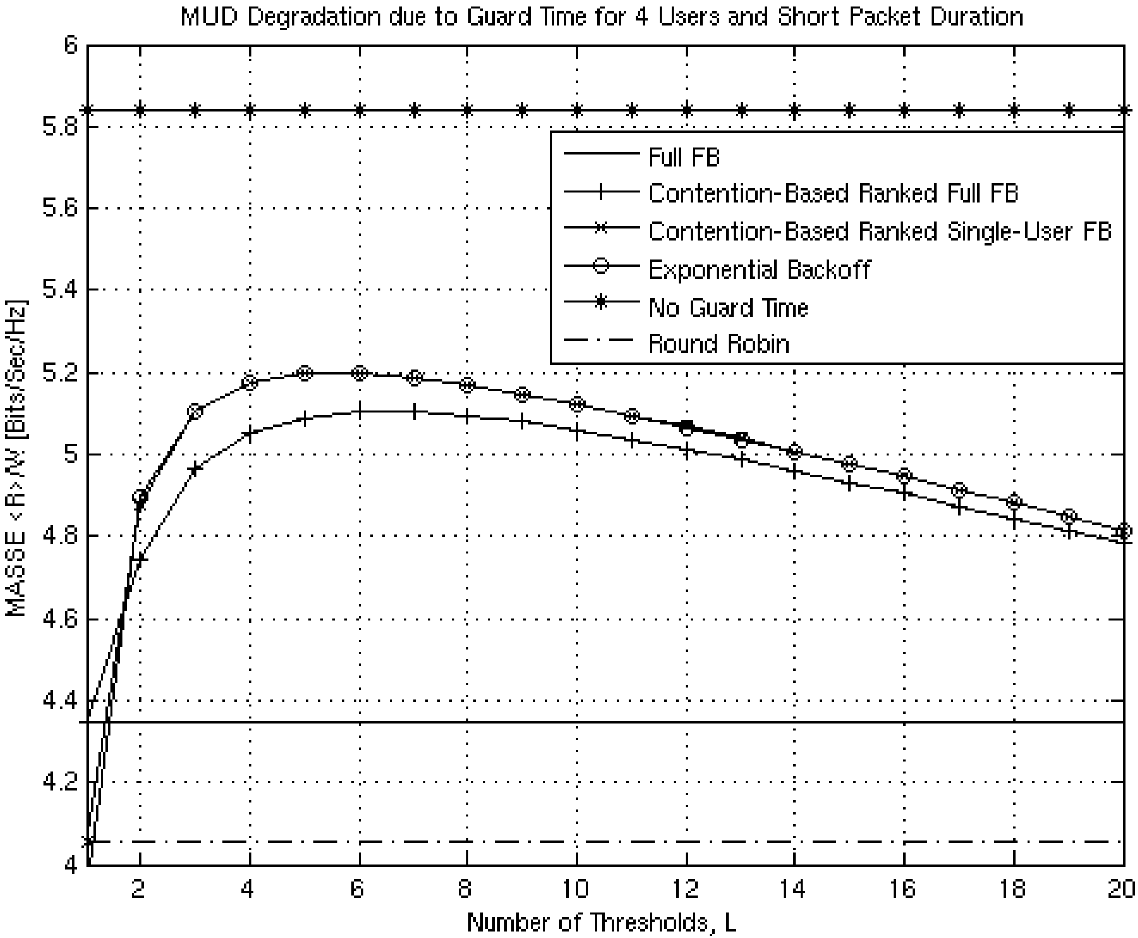
Implementations for IEEE802.11 Ad Hoc

We compare our 3 proposed algorithms with the following schemes:

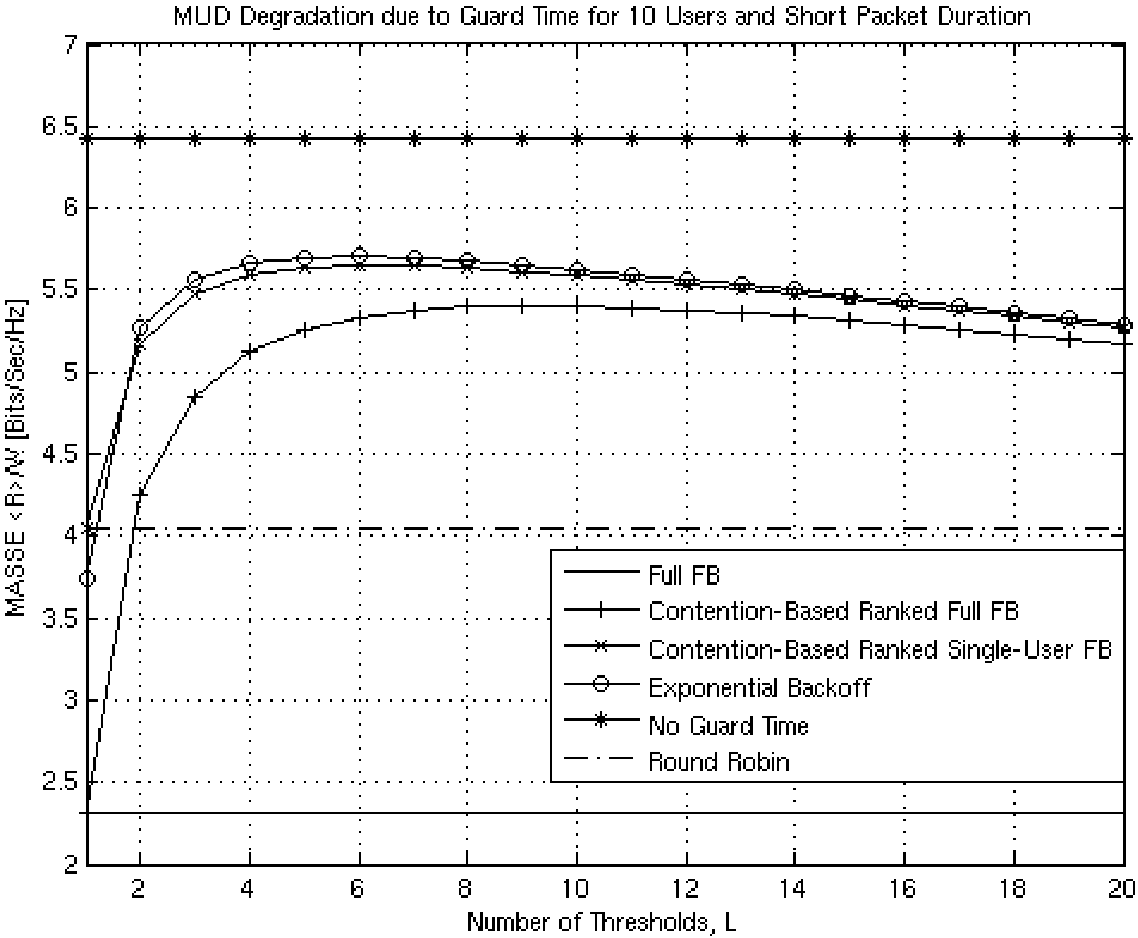
- Full feedback
- No guard time + Full MUD
- Round Robin


Guard Time for 4 Users
15
10
5
0
Guard Time [T_MS]
2
4
6
8
10
12
14
16
18
20
Number of Thresholds, L
Full FB
Contention-Based Ranked Full FB
Contention-Based Ranked Single-User FB
Exponential Backoff

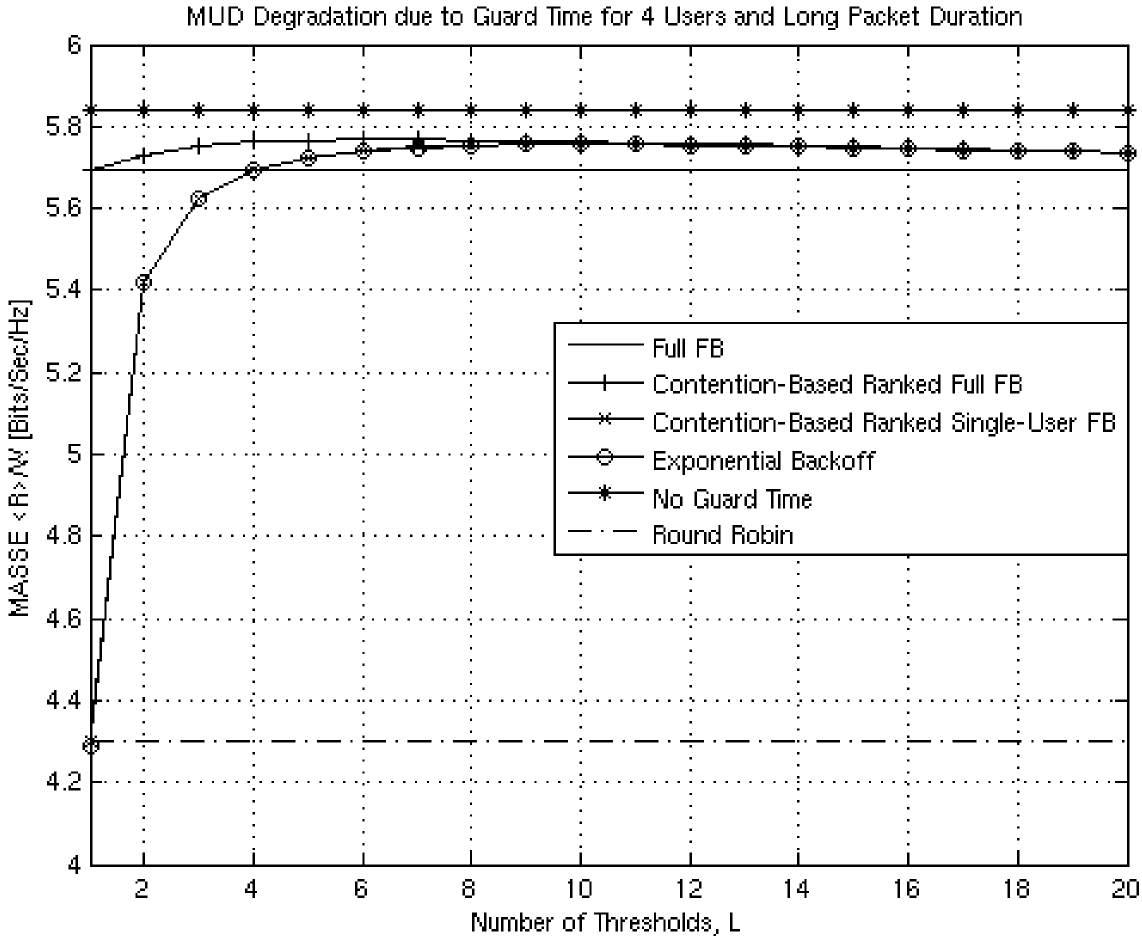

Guard Time for 10 Users
Guard Time [T_{MS}]
Number of Thresholds, L
Full FB
Contention-Based Ranked Full FB
Contention-Based Ranked Single-User FB
Exponential Backoff
0
5
10
15
20
25
30
35
2
4
6
8
10
12
14
16
18
20



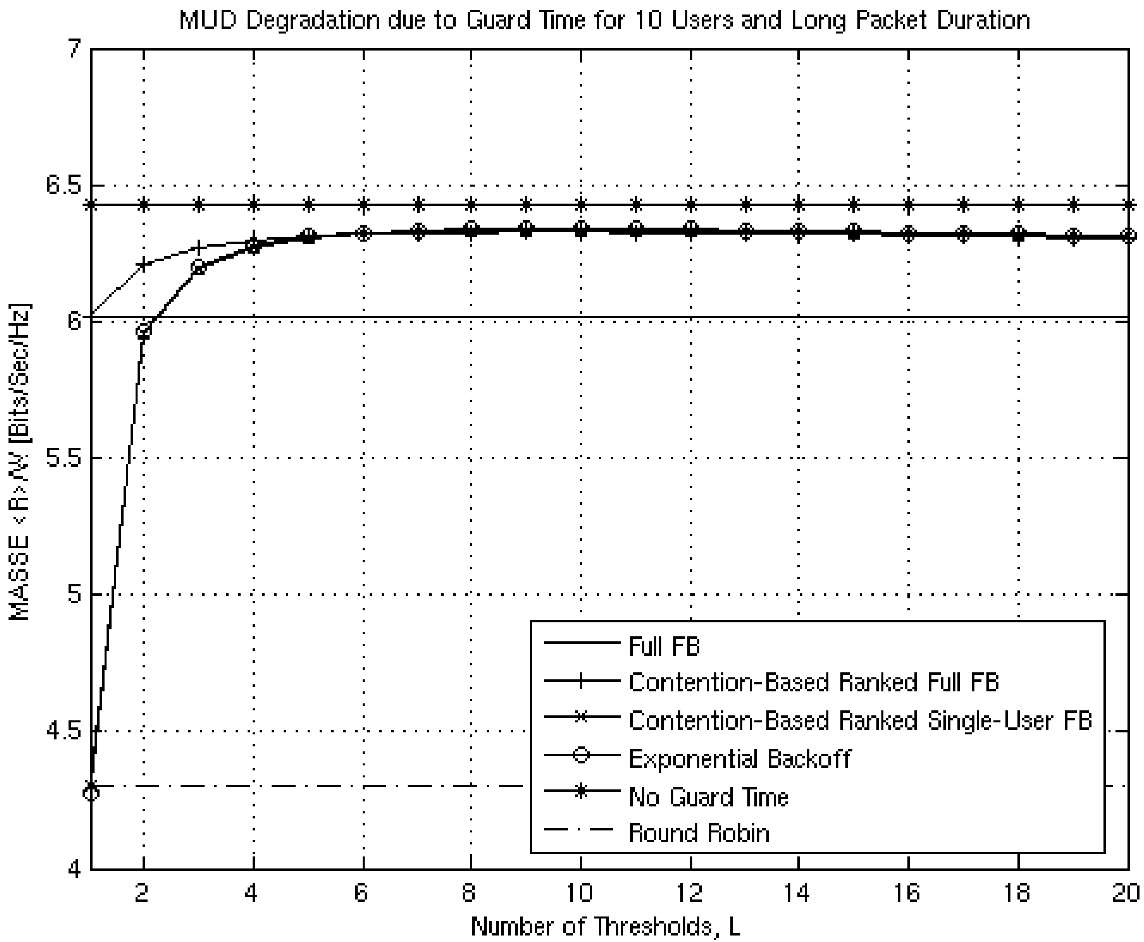
NTNU
Department of
Electronics and Telecommunications



MUD Degradation due to Guard Time for 10 Users and Short Packet Duration
MASSE <R> /W [Bits/Sec/Hz]
Number of Thresholds, L
Full FB
Contention-Based Ranked Full FB
Contention-Based Ranked Single-User FB
Exponential Backoff
No Guard Time
Round Robin



MUD Degradation due to Guard Time for 4 Users and Long Packet Duration
MASSE <R> /W [Bits/Sec/Hz]
Number of Thresholds, L
6
5.8
5.6
5.4
5.2
5
4.8
4.6
4.4
4.2
4
2
4
6
8
10
12
14
16
18
20
Full FB
Contention-Based Ranked Full FB
Contention-Based Ranked Single-User FB
Exponential Backoff
No Guard Time
Round Robin



NTNU
Department of
Electronics and Telecommunications

Conclusion

- By using multiple feedback thresholds, we obtain an increase in the system spectral efficiency compared to full feedback.
- This algorithm will be most efficient for a fast fading channel, i. e. for short coherence times.