Scheduling in WiMAX: Baseline Multi-class Simulations

Chakchai So-In, Raj Jain, and Abdel-Karim Al Tamimi
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@wustl.edu, cs5@cec.wustl.edu, and aa7@cec.wustl.edu

Presented at WiMAX Forum Meeting
Washington DC, 19th November 2007

These slides are available on-line at:
http://www.cse.wustl.edu/~jain/schd704.htm

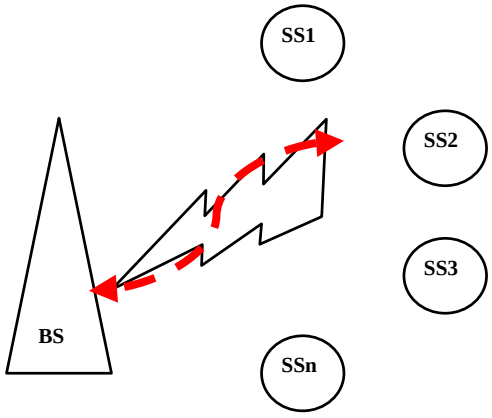
Overview

- Admission Control
- Fair scheduling algorithm
- Simulation results

Configuration

- Frame Duration: 5 ms
- Downlink ratio: 0.6 (DL 60%, UL 40%) ~ 1.5:1 (0.66 crashes)
- Initial Modulation Scheme: DIUC = 2 (QPSK1/2)
  - 2 bit/symbol × 1/2 repetition ⇒ 1 bit per symbol
- Bandwidth: 10 MHz (FFT: 1024)
- PUSC #DL Subchannels: 30, #UL Subchannels: 35
- ARQ Disabled (Bug: Higher loss with ARQ)

Configuration (Cont)



- Single BS with multiple SSs

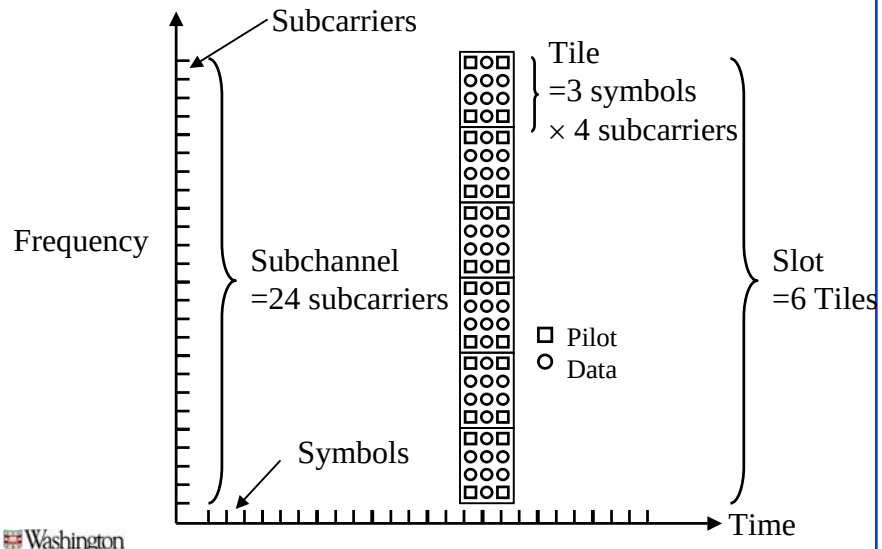
Workload

- UL only:
  - UGS: UDP (CBR) at rate 200 kbps (500 Bytes APDU)
  - BE: UDP at rate 150 kbps or 700 kbps (500 Bytes APDU)
- UGS: MAC request: 275 bytes every 10 ms ~220 kbps
- 1 BS, 1 SS or 3 SSs
- Metrics: Application Throughput (kbps)
- Simulation starts from 20 sec to 50 sec

Error Model

- Channel model: Cost231
- Fading Model: ITU_PDP PED_A

Symbols, Tiles, and Slots (Uplink)



Max BW and SS Calculation (Uplink)

- 1 frame (5 ms): 44 symbols after RTG and TTG
- With DL ratio=0.6, DL=26 and UL=18 symbols
- Ranging = *2+1*, Contention=1+1, 12+1 left for data
- Each tile is 3 symbols wide ⇒ 12/3 tile-columns
- Each slot is 6 tiles high ⇒ 1 subchannel
- PUSC ⇒ 35 subchannels
- Per UL Frame = 35 × 12/3 = 140 free slots
- Each slot has 72 symbol-subcarriers but only 48 for data + 24 pilots
- QPSK1/2 = 1 bit/symbol, 48 × 1/8 = 6 bytes per slot
- UL Capacity = 140 × 6 × 8 / (5 ms) = 1.344 Mbps (Max BW)

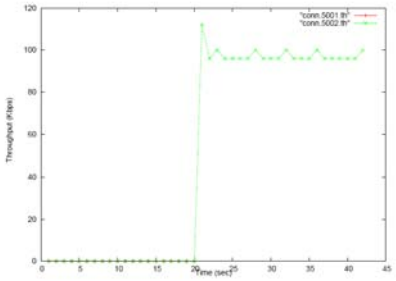
Original Scheduling Methodology

- For UGS: grant = reserved slots;
- For others: grant = request
- Problems:
  - Can't support overload traffic
  - Does not allocate BW fairly

New Scheduling Methodology

- For UGS: grant = reserved slots;
  - Left slots = free slots after UGS
  - Fair Share = left slots/#non-UGS connections;
- For others: if (request < Fair share) grant = request,
  else grant = Fair share
  - Loop till no more free slots

UL Throughput for BE (2SS), 700Kbps

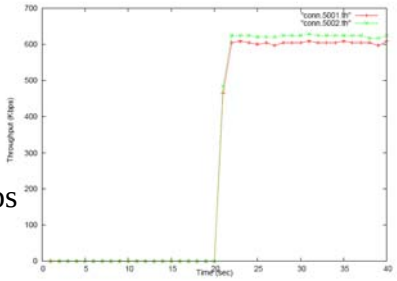


Old Scheduler:

← Th_{avg} = :97.97 Kbps

Drop = 6459 packets

2nd User does not get fair share



New Scheduler:

Th_{avg} = 1208.03Kbps

Drop = 956packets

Fair Allocation

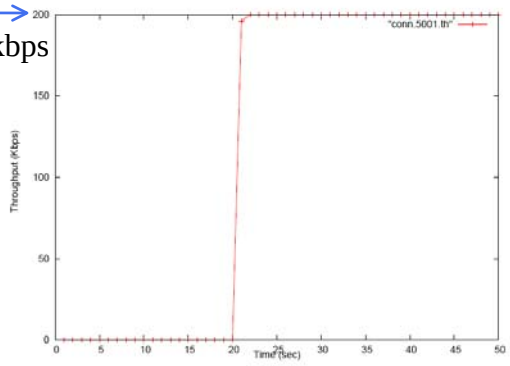
Simple Admission Control

- UGS connections are rejected if total load is more than capacity
- For UGS (220 kbps)
  Assuming 20 kbps for packing/segmentation overhead
  ⇒ Can support max 6 SS connections

Simulation Results for Simple Admission Control

Throughput: 1 SS (200 kbps)

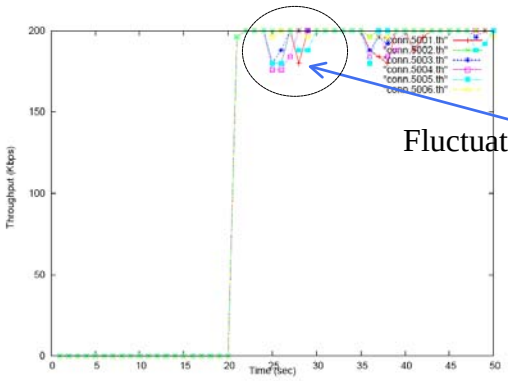
Perfect 200 kbps



- Throughput: 200 kbps, Drop: 0 packet

Throughput: 6 SS, 200 kbps

Throughput



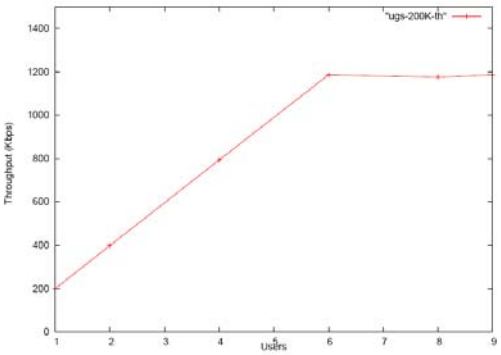
Fluctuation due to drop

Time

- Throughput: 1185.97 kbps, Drop: 377 packets

Throughput vs. #SSs

Total Throughput

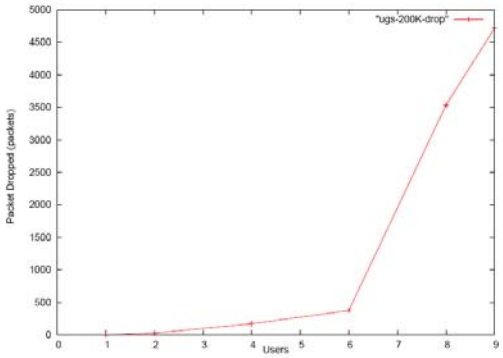


Users

- Supports max of 6 connections
- Throughput increases linearly
- Original simulation w/o admission control crashes on overload

Loss vs. #SSs

Packets Dropped



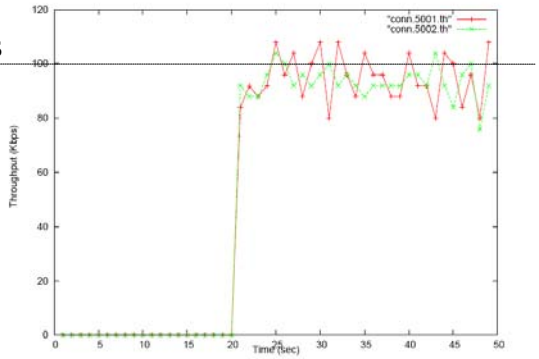
Users

- Packets dropped increases linearly
- Conclusion: Admission control is effective

Simulation Results for BE Traffic

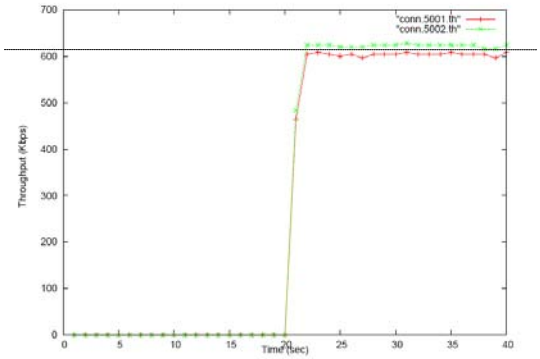
Throughput for BE (2 SSs, 100 kbps)

100 kbps



- Capacity = 1.33 Mbps, Load = 0.2 Mbps ⇒ Under-load Scenario
- Throughput: 189.13 kbps, Drop: 189 packets
- Statistically fair

Throughput for BE (2 SSs, 700 kbps)



- Capacity = 1.33 Mbps, Load = 1.4 Mbps ⇒ Overload Scenario
- Throughput: 1218.86 kbps, Drop: 1432 packets
- Statistically fair

Summary

- Need admission control for classes with guaranteed service
  - Implemented a simple admission control
  - Allows us to do simulations with larger number of users without crashing
- Need fairness for overload situations
  - Implemented a simple fairness mechanism
  - Both UGS and BE simulations now show fair throughput
  - Old simulation used to starve some users.

Future Work

- Debug BE traffic (stop crashes)
- Study delay behavior
- Simulate other classes of traffic
- Fixed Modulation per connection
  $\Rightarrow$ Allows different modulations per SS
- Fixed 1 connection per node
  $\Rightarrow$ Allow multiple connections and classes per SS