NIMBHORKAR, SANKALP ULHAS. Adaptive Channel Width Allocation for Multihop Wireless Networks with Joint Scheduling and Congestion Control. (Under the direction of Dr Rudra Dutta.)

Traditionally wireless networks have used static width channel allocation. In a common usage in 802.11 networks, channel 1, 6 and 11 are used with 20MHz width. This limits the spectrum utilization. Intuitively a better spectrum utilization is possible if we relax the limitation of fixed width channels. Recently, there have been approaches to study the advantage of variable width channels. We investigate the case for adaptive channel width allocation in conjunction with back-pressure based scheduling/routing in TDM environment. The choice of TDM is made to study the effect of variable width channel allocation scheme without interference of contention resolution protocols.

Capacity of multihop wireless networks is limited by interference in local neighborhood. Transport layer congestion control cannot take into account the effect of link access latencies and hence, is inadequate. This leads to capacity limitation and unfair bandwidth distribution across traffic flows. Network utility maximization framework shows that injection rate into flows can be chosen so as to maximize overall network utility. Research studies have shown that isolated congestion control and scheduling may prevent the injection rates from converging to optimal solution to the utility maximization framework. It has also been proved that in a perfectly time slotted MAC, back-pressure based scheduling/congestion control would result in optimal throughput across the flows while maintaining queue stability. The notion of optimal throughput corresponds to the average throughput of a flow that would maximize its utility. Various schemes have been developed to translate the perfect TDMA-type solution to be applied to real-world CSMA/CA systems. But all the previous approaches assume static channel-width allocation.

We first present the case with continuous channel width allocation scheme, relaxing any relevant hardware constraints and making use of interference minimization framework as described later. We propose Gwidcont, a greedy adaptive channel-width allocation algorithm for a multihop wireless network. Each node is assumed to use a NIC capable of using arbitrarily wide channel within the permissible spectrum. We also assume this NIC to be able to transmit on an arbitrary number of channels at the same time. A dedicated control channel takes care of maintaining time-slots, message passing for the actual channel allocation algorithm and neighbor discovery throughout the network. A dedicated radio per node always listens to control channel.

Our theoretical model and simulation results prove that adaptive channel width allocation
schemes yield better throughput, fairness as well as delay statistics. We then propose Gwiddis, adaptation of Gwid which adheres to 802.11 n channel widths and compare its performance with Gwid.

This work does not address issues like determining symbol transmission duration, guard interval and inter-frame spacings (and hence the net transmission rate) for a given channel width. Such extensions are appropriate possible directions for future work in this area.
Adaptive Channel Width Allocation for Multihop Wireless Networks
with Joint Scheduling and Congestion Control

by
Sankalp Ulhas Nimbhorkar

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Computer Science
Raleigh, North Carolina
2012

APPROVED BY:

Dr George Rouskas
Dr David Thuente
Dr Rudra Dutta
Chair of Advisory Committee
DEDICATION

To my parents, teachers and professors.
The author was born in 1988 in Indore, Madhya Pradesh, India. After finishing schooling from a small town called Khamgaon and Junior College (High-school) from Akola, he enrolled for Bachelor of Technology program in Computer Engineering at the autonomous Veermata Jijabai Technological Institute, affiliated to the University of Mumbai in Mumbai, Maharashtra, India in 2005 and graduated in 2009. Then, in Fall 2009, he enrolled for Master of Science program in the department of Computer Science in North Carolina State University, Raleigh, NC.
ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr Rudra Dutta for guiding me in this study. I am really grateful to him for introducing me to the domain of back-pressure based congestion control in wireless mesh networks. I also thank him for providing valuable insight about validating results and various other topics.

I also thank Dr George Rouskas and Dr David Thuente for agreeing to serve on my thesis advisory committee.

I also thank my senior Parth Pathak for his insight about various problems on theoretical modeling of the topic.

Finally, I would like to thank my family and friends for their continued support and patience. I cannot even begin to fathom the depth of my debt to for their continued encouragement and love which has enabled me to pull through my graduate studies.
TABLE OF CONTENTS

List of Tables .................................................................................. ix

List of Figures .................................................................................. x

Chapter 1 Introduction ....................................................................... 1
  1.1 Back-pressure and utility maximization framework ....................... 1
  1.2 Channel width adaptation ........................................................... 2
  1.3 Our Choice of TDM ................................................................. 2
  1.4 Organization ............................................................................ 2

Chapter 2 Related Work ................................................................... 4
  2.1 Back-pressure scheduling and congestion control ....................... 4
  2.2 Channel Width Adaptation ....................................................... 5

Chapter 3 Backpressure Framework ............................................... 6
  3.1 Back-pressure ........................................................................ 6
  3.2 Utility Maximization Framework ............................................. 7
  3.3 The Relation between Throughput and Utility ........................... 7
  3.4 Elastic, semi-elastic and inelastic flows ..................................... 8

Chapter 4 System Model .................................................................. 10
  4.1 Terminology ........................................................................... 10
    4.1.1 PDQ, Queue and Flow ....................................................... 10
    4.1.2 Backlog .......................................................................... 10
    4.1.3 Urgency Weight .............................................................. 10
    4.1.4 Max weight or max weight first scheduling ....................... 10
    4.1.5 MAC Priority ................................................................. 11
    4.1.6 Activation Vector ............................................................ 11
    4.1.7 Source and Sink .............................................................. 11
  4.2 Network Model ....................................................................... 11
    4.2.1 Notation ......................................................................... 11
    4.2.2 Assumptions ................................................................... 12
    4.2.3 Channel Width And Transmission Rate ........................... 12
    4.2.4 The Interference Minimization Scheme ............................ 13
  4.3 The Card GRANWID ............................................................... 14
  4.4 Lower bound on Throughput .................................................... 16
    4.4.1 Corollary I ..................................................................... 16
    4.4.2 Corollary II ..................................................................... 16
    4.4.3 Inference ......................................................................... 17
# Chapter 5 Greedy Channel Width Allocation Algorithm

5.1 MAC Priorities .................................................. 18
5.2 Utility-based Rate Control or Source Rate Control .......................... 18
5.3 Testing for Interference ........................................ 20
5.4 Choice of Time Slot Duration .................................. 20
5.5 Calculating the Activation Vector ................................ 21
  5.5.1 Greedy Backlog Reversing ................................ 21
  5.5.2 Local Proportional Channel Width Assignment ................. 21
  5.5.3 MAC Priorities ............................................. 21
5.6 The Algorithm Gwid ............................................ 22
5.7 Discrete Channel Width Allocation ................................ 23
5.8 Example ......................................................... 24
  5.8.1 Gwid-cont .................................................. 24
  5.8.2 Gwid-dis .................................................. 24
5.9 Realization Feasibility .......................................... 26

# Chapter 6 Experimental Setup .................................. 28

6.1 Metrics of Interest ............................................. 28
6.2 Cases to Compare ............................................... 29
6.3 Simulation Methodology ........................................ 29
  6.3.1 Packet Generation ......................................... 30
  6.3.2 Rate Control .............................................. 30
  6.3.3 Arrange PDQs ............................................ 30
  6.3.4 Activation Vector ........................................ 30
  6.3.5 Transmission .............................................. 30
6.4 Network Setup .................................................. 33
  6.4.1 Topology Grid ............................................. 33
  6.4.2 Random Topologies ....................................... 33

# Chapter 7 Results .................................................. 35

7.1 Topology - Grid ................................................ 35
  7.1.1 Aggregate Injection Rate .................................. 35
  7.1.2 Per Flow Injection Rate ................................... 36
  7.1.3 Per Flow Delay ............................................ 37
  7.1.4 Spatial Reuse ............................................. 38
  7.1.5 Air time Utilization ...................................... 39
  7.1.6 Throughpt ................................................ 40
  7.1.7 Aggregate Network Utility ............................... 41
  7.1.8 Buffer Occupancy ........................................ 42
  7.1.9 Fairness ................................................... 43
7.2 Random Topologies ............................................. 44
  7.2.1 Aggregate Injection Rate .................................. 44
  7.2.2 Delay ...................................................... 45
  7.2.3 Spatial Reuse ............................................. 46
  7.2.4 Air time Utilization ...................................... 47
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2.5 Throughput</td>
<td>48</td>
</tr>
<tr>
<td>7.2.6 Network Utility</td>
<td>49</td>
</tr>
<tr>
<td>7.2.7 Buffer Occupancy</td>
<td>50</td>
</tr>
<tr>
<td>7.2.8 Fairness</td>
<td>52</td>
</tr>
<tr>
<td><strong>Chapter 8 Conclusion and Future Work</strong></td>
<td>53</td>
</tr>
<tr>
<td>8.1 Conclusion</td>
<td>53</td>
</tr>
<tr>
<td>8.1.1 Throughput and Utility</td>
<td>53</td>
</tr>
<tr>
<td>8.1.2 Fairness</td>
<td>53</td>
</tr>
<tr>
<td>8.1.3 Delay</td>
<td>54</td>
</tr>
<tr>
<td>8.2 Future Work</td>
<td>54</td>
</tr>
<tr>
<td><strong>References</strong></td>
<td>55</td>
</tr>
<tr>
<td><strong>Appendix</strong></td>
<td>57</td>
</tr>
<tr>
<td>Appendix A Random Topologies and Simulation Results</td>
<td>58</td>
</tr>
<tr>
<td>A.1 Topologies</td>
<td>58</td>
</tr>
<tr>
<td>A.2 Results</td>
<td>74</td>
</tr>
<tr>
<td>A.2.1 Random Topology 1</td>
<td>74</td>
</tr>
<tr>
<td>A.2.2 Random Topology 2</td>
<td>85</td>
</tr>
<tr>
<td>A.2.3 Random Topology 3</td>
<td>96</td>
</tr>
<tr>
<td>A.2.4 Random Topology 4</td>
<td>107</td>
</tr>
<tr>
<td>A.2.5 Random Topology 5</td>
<td>118</td>
</tr>
<tr>
<td>A.2.6 Random Topology 6</td>
<td>129</td>
</tr>
<tr>
<td>A.2.7 Random Topology 7</td>
<td>140</td>
</tr>
<tr>
<td>A.2.8 Random Topology 8</td>
<td>151</td>
</tr>
<tr>
<td>A.2.9 Random Topology 9</td>
<td>162</td>
</tr>
<tr>
<td>A.2.10 Random Topology 10</td>
<td>173</td>
</tr>
<tr>
<td>A.2.11 Random Topology 11</td>
<td>184</td>
</tr>
<tr>
<td>A.2.12 Random Topology 12</td>
<td>195</td>
</tr>
<tr>
<td>A.2.13 Random Topology 13</td>
<td>206</td>
</tr>
<tr>
<td>A.2.14 Random Topology 14</td>
<td>217</td>
</tr>
<tr>
<td>A.2.15 Random Topology 15</td>
<td>228</td>
</tr>
<tr>
<td>A.2.16 Random Topology 16</td>
<td>239</td>
</tr>
<tr>
<td>A.2.17 Random Topology 17</td>
<td>250</td>
</tr>
<tr>
<td>A.2.18 Random Topology 18</td>
<td>261</td>
</tr>
<tr>
<td>A.2.19 Random Topology 19</td>
<td>272</td>
</tr>
<tr>
<td>A.2.20 Random Topology 20</td>
<td>283</td>
</tr>
<tr>
<td>A.2.21 Random Topology 21</td>
<td>294</td>
</tr>
<tr>
<td>A.2.22 Random Topology 22</td>
<td>305</td>
</tr>
<tr>
<td>A.2.23 Random Topology 23</td>
<td>316</td>
</tr>
<tr>
<td>A.2.24 Random Topology 24</td>
<td>327</td>
</tr>
<tr>
<td>A.2.25 Random Topology 25</td>
<td>338</td>
</tr>
<tr>
<td>A.2.26 Random Topology 26</td>
<td>349</td>
</tr>
<tr>
<td>A.2.27 Random Topology 27</td>
<td>360</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

Table 4.1 Variations in Received Power, Transmission Range and Interference Range with Channel Width ........................................... 15

Table 6.1 Flow Setup - Topology Grid ............................................. 34
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Optimal throughput policy vs NUM. The figures in parentheses represent the queue backlogs</td>
<td>8</td>
</tr>
<tr>
<td>4.1</td>
<td>Explaining the Model</td>
<td>13</td>
</tr>
<tr>
<td>4.2</td>
<td>Interference minimization Scheme</td>
<td>14</td>
</tr>
<tr>
<td>4.3</td>
<td>Transmission power vs channel width for $\text{trange} = 100m$</td>
<td>16</td>
</tr>
<tr>
<td>4.4</td>
<td>Channel Allocation Schemes</td>
<td>17</td>
</tr>
<tr>
<td>5.1</td>
<td>Sample Run</td>
<td>26</td>
</tr>
<tr>
<td>6.1</td>
<td>Simulation Components</td>
<td>31</td>
</tr>
<tr>
<td>7.1</td>
<td>Grid - Pure injection rate</td>
<td>35</td>
</tr>
<tr>
<td>7.2</td>
<td>Grid - Per flow Injection Rate</td>
<td>36</td>
</tr>
<tr>
<td>7.3</td>
<td>Grid - Per Flow Delay</td>
<td>37</td>
</tr>
<tr>
<td>7.4</td>
<td>Grid - Average Delay</td>
<td>38</td>
</tr>
<tr>
<td>7.5</td>
<td>Grid - Spatial Reuse</td>
<td>38</td>
</tr>
<tr>
<td>7.6</td>
<td>Grid - Air Time Utilization</td>
<td>39</td>
</tr>
<tr>
<td>7.7</td>
<td>Grid - Per flow Throughput</td>
<td>40</td>
</tr>
<tr>
<td>7.8</td>
<td>Grid - Average Throughput</td>
<td>41</td>
</tr>
<tr>
<td>7.9</td>
<td>Grid - Aggregate Utility</td>
<td>41</td>
</tr>
<tr>
<td>7.10</td>
<td>Grid - Per Node Buffer Utilization</td>
<td>42</td>
</tr>
<tr>
<td>7.11</td>
<td>Grid - Network Buffer Utilization</td>
<td>43</td>
</tr>
<tr>
<td>7.12</td>
<td>Grid - Fairness Index</td>
<td>43</td>
</tr>
<tr>
<td>7.13</td>
<td>Aggregate Injection Rate</td>
<td>44</td>
</tr>
<tr>
<td>7.14</td>
<td>Aggregate Delay</td>
<td>45</td>
</tr>
<tr>
<td>7.15</td>
<td>Spatial Reuse</td>
<td>46</td>
</tr>
<tr>
<td>7.16</td>
<td>Air time Utilization</td>
<td>47</td>
</tr>
<tr>
<td>7.17</td>
<td>Throughput</td>
<td>48</td>
</tr>
<tr>
<td>7.18</td>
<td>Aggregate Utility</td>
<td>49</td>
</tr>
<tr>
<td>7.19</td>
<td>Buffer Occupancy</td>
<td>50</td>
</tr>
<tr>
<td>7.20</td>
<td>Buffer Occupancy</td>
<td>51</td>
</tr>
<tr>
<td>7.21</td>
<td>Fairness</td>
<td>52</td>
</tr>
<tr>
<td>A.1</td>
<td>Random Topology 1</td>
<td>58</td>
</tr>
<tr>
<td>A.2</td>
<td>Random Topology 2</td>
<td>59</td>
</tr>
<tr>
<td>A.3</td>
<td>Random Topology 3</td>
<td>59</td>
</tr>
<tr>
<td>A.4</td>
<td>Random Topology 4</td>
<td>60</td>
</tr>
<tr>
<td>A.5</td>
<td>Random Topology 5</td>
<td>60</td>
</tr>
<tr>
<td>A.6</td>
<td>Random Topology 6</td>
<td>61</td>
</tr>
<tr>
<td>A.7</td>
<td>Random Topology 7</td>
<td>61</td>
</tr>
<tr>
<td>A.8</td>
<td>Random Topology 8</td>
<td>62</td>
</tr>
</tbody>
</table>
Figure A.185 Random Topology 15 - Pure injection rate .......................... 228
Figure A.186 Random Topology 15 - Per flow Injection Rate .................... 229
Figure A.187 Random Topology 15 - Per Flow Delay .............................. 230
Figure A.188 Random Topology 15 - Spatial Reuse ............................... 231
Figure A.189 Random Topology 15 - Air Time Utilization ......................... 232
Figure A.190 Random Topology 15 - Per flow Throughput ......................... 233
Figure A.191 Random Topology 15 - Average Throughput ......................... 234
Figure A.192 Random Topology 15 - Aggregate Utility ......................... 235
Figure A.193 Random Topology 15 - Per Node Buffer Utilization ............... 236
Figure A.194 Random Topology 15 - Network Buffer Utilization ................. 237
Figure A.195 Random Topology 15 - Fairness Index .............................. 238
Figure A.196 Random Topology 16 - Pure injection rate ......................... 239
Figure A.197 Random Topology 16 - Per flow Injection Rate ..................... 240
Figure A.198 Random Topology 16 - Per Flow Delay .............................. 241
Figure A.199 Random Topology 16 - Spatial Reuse ............................... 242
Figure A.200 Random Topology 16 - Air Time Utilization ......................... 243
Figure A.201 Random Topology 16 - Per flow Throughput ......................... 244
Figure A.202 Random Topology 16 - Average Throughput ......................... 245
Figure A.203 Random Topology 16 - Aggregate Utility ........................... 246
Figure A.204 Random Topology 16 - Per Node Buffer Utilization ............... 247
Figure A.205 Random Topology 16 - Network Buffer Utilization ................. 248
Figure A.206 Random Topology 16 - Fairness Index .............................. 249
Figure A.207 Random Topology 17 - Pure injection rate ......................... 250
Figure A.208 Random Topology 17 - Per flow Injection Rate ..................... 251
Figure A.209 Random Topology 17 - Per Flow Delay .............................. 252
Figure A.210 Random Topology 17 - Spatial Reuse ............................... 253
Figure A.211 Random Topology 17 - Air Time Utilization ......................... 254
Figure A.212 Random Topology 17 - Per flow Throughput ......................... 255
Figure A.213 Random Topology 17 - Average Throughput ......................... 256
Figure A.214 Random Topology 17 - Aggregate Utility ........................... 257
Figure A.215 Random Topology 17 - Per Node Buffer Utilization ............... 258
Figure A.216 Random Topology 17 - Network Buffer Utilization ................. 259
Figure A.217 Random Topology 17 - Fairness Index .............................. 260
Figure A.218 Random Topology 18 - Pure injection rate ......................... 261
Figure A.219 Random Topology 18 - Per flow Injection Rate ..................... 262
Figure A.220 Random Topology 18 - Per Flow Delay .............................. 263
Figure A.221 Random Topology 18 - Spatial Reuse ............................... 264
Figure A.222 Random Topology 18 - Air Time Utilization ......................... 265
Figure A.223 Random Topology 18 - Per flow Throughput ......................... 266
Figure A.224 Random Topology 18 - Average Throughput ......................... 267
Figure A.225 Random Topology 18 - Aggregate Utility ........................... 268
Figure A.226 Random Topology 18 - Per Node Buffer Utilization ............... 269
Figure A.227 Random Topology 18 - Network Buffer Utilization ................. 270
Figure A.228 Random Topology 18 - Fairness Index .............................. 271
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.229</td>
<td>Random Topology 19 - Pure injection rate</td>
<td>272</td>
</tr>
<tr>
<td>A.230</td>
<td>Random Topology 19 - Per flow Injection Rate</td>
<td>273</td>
</tr>
<tr>
<td>A.231</td>
<td>Random Topology 19 - Per Flow Delay</td>
<td>274</td>
</tr>
<tr>
<td>A.232</td>
<td>Random Topology 19 - Spatial Reuse</td>
<td>275</td>
</tr>
<tr>
<td>A.233</td>
<td>Random Topology 19 - Air Time Utilization</td>
<td>276</td>
</tr>
<tr>
<td>A.234</td>
<td>Random Topology 19 - Per flow Throughput</td>
<td>277</td>
</tr>
<tr>
<td>A.235</td>
<td>Random Topology 19 - Average Throughput</td>
<td>278</td>
</tr>
<tr>
<td>A.236</td>
<td>Random Topology 19 - Aggregate Utility</td>
<td>279</td>
</tr>
<tr>
<td>A.237</td>
<td>Random Topology 19 - Per Node Buffer Utilization</td>
<td>280</td>
</tr>
<tr>
<td>A.238</td>
<td>Random Topology 19 - Network Buffer Utilization</td>
<td>281</td>
</tr>
<tr>
<td>A.239</td>
<td>Random Topology 19 - Fairness Index</td>
<td>282</td>
</tr>
<tr>
<td>A.240</td>
<td>Random Topology 20 - Pure injection rate</td>
<td>283</td>
</tr>
<tr>
<td>A.241</td>
<td>Random Topology 20 - Per flow Injection Rate</td>
<td>284</td>
</tr>
<tr>
<td>A.242</td>
<td>Random Topology 20 - Per Flow Delay</td>
<td>285</td>
</tr>
<tr>
<td>A.243</td>
<td>Random Topology 20 - Spatial Reuse</td>
<td>286</td>
</tr>
<tr>
<td>A.244</td>
<td>Random Topology 20 - Air Time Utilization</td>
<td>287</td>
</tr>
<tr>
<td>A.245</td>
<td>Random Topology 20 - Per flow Throughput</td>
<td>288</td>
</tr>
<tr>
<td>A.246</td>
<td>Random Topology 20 - Average Throughput</td>
<td>289</td>
</tr>
<tr>
<td>A.247</td>
<td>Random Topology 20 - Aggregate Utility</td>
<td>290</td>
</tr>
<tr>
<td>A.248</td>
<td>Random Topology 20 - Per Node Buffer Utilization</td>
<td>291</td>
</tr>
<tr>
<td>A.249</td>
<td>Random Topology 20 - Network Buffer Utilization</td>
<td>292</td>
</tr>
<tr>
<td>A.250</td>
<td>Random Topology 20 - Fairness Index</td>
<td>293</td>
</tr>
<tr>
<td>A.251</td>
<td>Random Topology 21 - Pure injection rate</td>
<td>294</td>
</tr>
<tr>
<td>A.252</td>
<td>Random Topology 21 - Per flow Injection Rate</td>
<td>295</td>
</tr>
<tr>
<td>A.253</td>
<td>Random Topology 21 - Per Flow Delay</td>
<td>296</td>
</tr>
<tr>
<td>A.254</td>
<td>Random Topology 21 - Spatial Reuse</td>
<td>297</td>
</tr>
<tr>
<td>A.255</td>
<td>Random Topology 21 - Air Time Utilization</td>
<td>298</td>
</tr>
<tr>
<td>A.256</td>
<td>Random Topology 21 - Per flow Throughput</td>
<td>299</td>
</tr>
<tr>
<td>A.257</td>
<td>Random Topology 21 - Average Throughput</td>
<td>300</td>
</tr>
<tr>
<td>A.258</td>
<td>Random Topology 21 - Aggregate Utility</td>
<td>301</td>
</tr>
<tr>
<td>A.259</td>
<td>Random Topology 21 - Per Node Buffer Utilization</td>
<td>302</td>
</tr>
<tr>
<td>A.260</td>
<td>Random Topology 21 - Network Buffer Utilization</td>
<td>303</td>
</tr>
<tr>
<td>A.261</td>
<td>Random Topology 21 - Fairness Index</td>
<td>304</td>
</tr>
<tr>
<td>A.262</td>
<td>Random Topology 22 - Pure injection rate</td>
<td>305</td>
</tr>
<tr>
<td>A.263</td>
<td>Random Topology 22 - Per flow Injection Rate</td>
<td>306</td>
</tr>
<tr>
<td>A.264</td>
<td>Random Topology 22 - Per Flow Delay</td>
<td>307</td>
</tr>
<tr>
<td>A.265</td>
<td>Random Topology 22 - Spatial Reuse</td>
<td>308</td>
</tr>
<tr>
<td>A.266</td>
<td>Random Topology 22 - Air Time Utilization</td>
<td>309</td>
</tr>
<tr>
<td>A.267</td>
<td>Random Topology 22 - Per flow Throughput</td>
<td>310</td>
</tr>
<tr>
<td>A.268</td>
<td>Random Topology 22 - Average Throughput</td>
<td>311</td>
</tr>
<tr>
<td>A.269</td>
<td>Random Topology 22 - Aggregate Utility</td>
<td>312</td>
</tr>
<tr>
<td>A.270</td>
<td>Random Topology 22 - Per Node Buffer Utilization</td>
<td>313</td>
</tr>
<tr>
<td>A.271</td>
<td>Random Topology 22 - Network Buffer Utilization</td>
<td>314</td>
</tr>
<tr>
<td>A.272</td>
<td>Random Topology 22 - Fairness Index</td>
<td>315</td>
</tr>
</tbody>
</table>

xvi
Figure A.273 Random Topology 23 - Pure injection rate .......................... 316
Figure A.274 Random Topology 23 - Per flow Injection Rate ...................... 317
Figure A.275 Random Topology 23 - Per Flow Delay ............................... 318
Figure A.276 Random Topology 23 - Spatial Reuse ................................. 319
Figure A.277 Random Topology 23 - Air Time Utilization ......................... 320
Figure A.278 Random Topology 23 - Per flow Throughput ......................... 321
Figure A.279 Random Topology 23 - Average Throughput ......................... 322
Figure A.280 Random Topology 23 - Aggregate Utility ............................ 323
Figure A.281 Random Topology 23 - Per Node Buffer Utilization ............... 324
Figure A.282 Random Topology 23 - Network Buffer Utilization ................. 325
Figure A.283 Random Topology 23 - Fairness Index ................................ 326
Figure A.284 Random Topology 24 - Pure injection rate .......................... 327
Figure A.285 Random Topology 24 - Per flow Injection Rate ...................... 328
Figure A.286 Random Topology 24 - Per flow Delay ............................... 329
Figure A.287 Random Topology 24 - Spatial Reuse ................................ 330
Figure A.288 Random Topology 24 - Air Time Utilization ......................... 331
Figure A.289 Random Topology 24 - Per flow Throughput ......................... 332
Figure A.290 Random Topology 24 - Average Throughput ......................... 333
Figure A.291 Random Topology 24 - Aggregate Utility ............................ 334
Figure A.292 Random Topology 24 - Per Node Buffer Utilization ............... 335
Figure A.293 Random Topology 24 - Network Buffer Utilization ................. 336
Figure A.294 Random Topology 24 - Fairness Index ................................ 337
Figure A.295 Random Topology 25 - Pure injection rate .......................... 338
Figure A.296 Random Topology 25 - Per flow Injection Rate ...................... 339
Figure A.297 Random Topology 25 - Per flow Delay ............................... 340
Figure A.298 Random Topology 25 - Spatial Reuse ................................ 341
Figure A.299 Random Topology 25 - Air Time Utilization ......................... 342
Figure A.300 Random Topology 25 - Per flow Throughput ......................... 343
Figure A.301 Random Topology 25 - Average Throughput ......................... 344
Figure A.302 Random Topology 25 - Aggregate Utility ............................ 345
Figure A.303 Random Topology 25 - Per Node Buffer Utilization ............... 346
Figure A.304 Random Topology 25 - Network Buffer Utilization ................. 347
Figure A.305 Random Topology 25 - Fairness Index ................................ 348
Figure A.306 Random Topology 26 - Pure injection rate .......................... 349
Figure A.307 Random Topology 26 - Per flow Injection Rate ...................... 350
Figure A.308 Random Topology 26 - Per flow Delay ............................... 351
Figure A.309 Random Topology 26 - Spatial Reuse ................................ 352
Figure A.310 Random Topology 26 - Air Time Utilization ......................... 353
Figure A.311 Random Topology 26 - Per flow Throughput ......................... 354
Figure A.312 Random Topology 26 - Average Throughput ......................... 355
Figure A.313 Random Topology 26 - Aggregate Utility ............................ 356
Figure A.314 Random Topology 26 - Per Node Buffer Utilization ............... 357
Figure A.315 Random Topology 26 - Network Buffer Utilization ................. 358
Figure A.316 Random Topology 26 - Fairness Index ................................ 359
Chapter 1

Introduction

1.1 Back-pressure and utility maximization framework

Wireless mesh networks [3, 16] are increasingly being used as a scalable topology for internet access. [23] surveys challenged in designing wireless mesh networks. Interference in local neighborhood limits the capacity of multihop wireless networks. At a time in a local neighborhood, only one link can be scheduled for transmission at a given frequency band. 802.11 tries to be fair in bandwidth distribution using MACA [1] method but, fails in case of multihop wireless networks.

The backpressure scheduling/congestion control for multihop wireless networks was first proposed by Tassiulas et al. in [27]. It introduced the concept of congestion control based using per-destination queues (PDQs). The paper proved that to maintain queue stability, that is, the invariant that ingress rate never exceeds egress rate, while providing optimal throughput to a flow, the contention for medium access between the nodes should be resolved such that the product of the rate at which a queue is served and the backlog of the queue is maximized. Here the backlog refers to the difference between the length of a PDQ at current hop and the next hop.

Network utility maximization problem formalized in [15] states that the injection rates into the flows can be selected so that the utility of flows can be maximized. The utility of a flow indicates desired effect of the flow on the network and is represented as a concave function of aggregate throughput of the flow. It was proved in [9, 12, 18, 21] that the back-pressure policy can solve the problem of network utility maximization. The network utility represents desired effect of a flow, and hence, utility maximization framework can introduce fair bandwidth distribution across the flows.

Transport layer congestion control, like TCP, cannot take into account medium access latencies due to link schedulers and hence, is inadequate. [6] proved that isolated congestion control
and link scheduling may prevent queue lengths from converging. Thus, it validated joint ap-
proaches to congestion control and scheduling. [4] provided a theoretical joint approach.

1.2 Channel width adaptation

Channel width adaptation was first explored for the case of two communicating nodes by [8]. As noted by [17], with fixed channel width allocations in 802.11 only three 20MHz wide channels are available. [8] studied effect of varying channel width on peak throughput, transmission range and energy consumption. Since then various works have shown that adaptive channel width allocation can provide better throughput and fairer bandwidth distribution across the network. But, to the best of our knowledge this is the first attempt to study channel width allocation in conjunction with backpressure congestion control/scheduling framework.

For this work, we limit ourselves to fixed transmission power. We assume existence of an NIC that allows fully granular channel width utilization and the transmission rate thus obtained obeys Shannon’s theorem.

1.3 Our Choice of TDM

We aim to study the effect of purely variable channel widths on the backpressure based joint scheduling and congestion control. A contention resolution scheme like CSMA/CA has a potential of introducing unfairness in scheduling decisions. Also, owing to back-offs the slotted time scheduling decisions would intuitively underutilize the air time, i.e. the medium might remain idle introducing further throughput reductions. To leave these factors out, we chose to use TDM based LAN as a basis of our experiments.

Another reason for not using CSMA/CA is explained in section 4.4. The inter-frame spacings are absolute in time in case of 802.11. Because of this the increase obtaines in transmission rate with channel width is sub-linear. This also has potential to introduce aberrations in comparison of static and variable channel width schemes. This prompted us against the use of CSMA/CA for initial study. An adaptation of our scheme for CSMA/CA networks is left as a future work.

1.4 Organization

The remaining thesis is organized as follows. Chapter 2 describes related work. Chapter 3 explains background work about joint backpressure scheduling and utility based rate control and about effects of channel width adaptation. Chapter 4 describes our network model. Chapter 5 describes a greedy channel width allocation strategy and its adaptation for 802.11n [2] channels.
Chapter 6 describes evaluation method and Chapter 7 presents the results. Finally, Chapter 8 presents conclusions and future work.
Chapter 2

Related Work

In this work we aim to study the effect of varying channel width in case of backpressure based joint scheduling and congestion control. Thus, these fields become our basic building blocks. Hence, we review some related work in these domains.

2.1 Back-pressure scheduling and congestion control

Back-pressure based scheduling for maximum throughput was explored in [27, 28]. The utility maximization framework was proposed in [15]. Joint approaches to scheduling and congestion control were explored in [4, 29]. The back-pressure based scheduling/congestion control was shown to solve the problem of network utility maximization in [9, 12, 18, 21]. Throughput optimization in case of network utility maximization framework was studied in conjunction with OFDMA rate control by [13]. [11] proved that significant performance gain can be obtained if small packets are aggregated at IP layer. [4, 29, 11] use per-destination queues (PDQs) to implement back-pressure. Packet queues are arranged by destination. The backlog of a PDQ is just the difference of queue length of PDQ for a destination at current node and that at next hop node. These studies also use the notion of MAC priority, that is, assigning priority based on the backlog. While [4] implement MAC priorities by modifying the CWmin as defined in IEE 802.11 [19] standard, [29] uses 4 levels of differential service using modified madwifi driver. [11] uses MAC priority mechanism similar to IEEE 802.11 e [19] standard.

In our work, we prioritize flows based upon backlog. The notion of MAC priorities we use pertains to channel widths allocated.
2.2 Channel Width Adaptation

Channel width adaptation is a relatively recent domain introduced in [8]. Here the case of two communicating nodes was considered and the algorithm *samplewidth* was given to adapt to optimal channel width. One important observation here is that, for 802.11 networks, the proportionally between channel width and transmission rate as governed by Shannon’s theorem is only approximately followed. In our experiments we do not make use of 802.11 for data transmission and assume that the linear relationship holds. One more important observation here is that narrower channels have larger range for same transmission power because of better resistance to delay spread and decrease in minimum power required for successful reception. We use interference minimization framework to work around this issue as explained in Chapter 4.

Changes in neighborhood and hence possible routes according to varying channel widths were studied in [7]. A channel width allocation scheme for AP based wireless LANs was provided in [20]. This scheme allocates spectrum to APs based on traffic requirements and solves the problem of APs being bandwidth bottleneck. [30] applies concepts in game theory to prove that optimal channel width allocation in terms of throughput without starvation can be achieved by using cost measures to achieve dominant strategy equilibrium. [7] provides a joint approach to routing and channel width adaptation in wireless mesh networks but does not consider scheduling and congestion control. [17] provides a joint approach to topology design, spectrum allocation and routing.

[24] provided an approach to improve aggregate throughput by using channel width based on predicted packet size. In contrast, we study spectrum allocation in conjunction with scheduling and congestion control for throughput optimization as well as fair bandwidth allocation.
Chapter 3

Backpressure Framework

In this chapter we review the basic concepts of backpressure and network utility maximization in detail to set up a basic framework for our variable channel width model.

3.1 Back-pressure

Back-pressure congestion control/scheduling [27] uses PDQs to obtain optimal throughput while maintaining queue stability. Similar to [4], let, \( Q^i_d \) denote the PDQ for destination \( d \) at node \( i \) and \( q^i_d \) be its length.

Let, \( n(i, d) \) denote the next hop from node \( i \) to destination \( d \).

Let, \( r_{i,n(i,d)} \) denote the rate at which \( Q^i_d \) is served, i.e., the rate at which \( Q^i_d \) is scheduled for transmission.

The back-pressure policy for optimal throughput proposed in [27] states that, a policy for maximum throughput requires that links should be scheduled for transmission so that the product of backlog of a PDQ with maximum backlog and the rate at which it is served summed per node over the network is maximized, i.e.

\[
\text{maximize } \sum_i \left( \left( q^i_d - q^{n(i,d)}_d \right)_{\text{max}} \times r_{i,n(i,d)} \right)
\]

(3.1)

Here, \( q^i_d - q^{n(i,d)}_d \) is known as backlog of PDQ \( d \) at node \( i \).

Also, \( w^i_d = (q^i_d - q^{n(i,d)}_d) \times r_{i,n(i,d)} \) is known as urgency weight of PDQ \( d \) at node \( i \). (This is the reason scheduling schemes based on this are referred to as differential backlog scheduling schemes)[4]

This represents two scheduling decisions. First, within a node, a PDQ with maximum backlog is selected as candidate for transmission. Second, in a local neighborhood, a link with maximum urgency weight for such a candidate is scheduled. Time is considered to be slotted and
scheduling decision is taken at the beginning of every slot. In a perfectly slotted environment like TDMA, this will result in maximum throughput as well as stabilized queues.

### 3.2 Utility Maximization Framework

The Network Utility Maximization (NUM) problem was formalized in [15]. Let, in a set of all flows $F$ in a network, every flow $f$ be served at an injection rate $x_f$, and throughput $X_f$ has a concave utility function $U_f(X_f)$ associated with it which represents the desired effect of the flow on the network.

Further, let $C_i$ be capacity of the network $i$ and $X_i$ be the set of flows passing through $i$. Then the NUM problem is

$$\text{maximize } \sum_{f \in F} U_f(X_f) \quad (3.2)$$

subject to

$$\sum_{f \in F} x_f \leq C_i \quad (3.3)$$

### 3.3 The Relation between Throughput and Utility

As an example to understand the difference between the policy for optimal throughput and the NUM problem, consider figure 3.1 where a part of a multihop wireless network is shown. Consider the average transmission rates of these queues to be $r_1$, $r_2$ and so on. Let the throughputs achieved by these flows be $R_1$, $R_2$ and so on.

The policy for optimal throughput dictates that the scheduling decisions should try to achieve transmission rates that maximize the node wise sum of the product of queue backlog and queue transmission rate for the queue with maximum backlog at that node. That is, in this case, maximize the following expression

$$20 \times r_1 + 25 \times r_3 + 18 \times r_5 + \ldots..$$

On the other hand NUM dictates that if $R$ is system bandwidth, the scheduling scheme should achieve transmission rates so as to

$$\text{maximize} \quad U(R_1) + U(R_3) + U(R_4) + U(R_5) + U(R_6) \ldots..$$
Figure 3.1: Optimal throughput policy vs NUM. The figures in parentheses represent the queue backlogs

subject to

$$r_1 + r_2 + r_3 + r_4 + r_5 + r_6 + .... \leq R$$

Even though the problems proposed above are distinct, it has been proved that a policy for optimal throughput also solves the problem of network utility maximization, under concave utility functions[9, 12, 18, 21]. This reconciliation is intuitive. Higher backlog would be an indication of a faster flow. So if faster flows are permitted to inject packets at higher rate and served accordingly, the network utility would also increase. On the other hand, as long as utility function remains concave across the flows (faster flow always has higher utility), we cannot obtain higher network utility by serving faster flow with a lower transmission rate.

3.4 Elastic, semi-elastic and inelastic flows

Flows are typically classified as elastic, semi-elastic and inelastic [4, 26]:

A flow whose source cannot control injection rate (in other words, a flow that has fixed bandwidth requirement), is called as inelastic flow. Constant bit rate applications like voice calls are good examples of inelastic flows.

A flow that has a minimum injection rate required, but can also inject packets at a higher rate should bandwidth be available or rate control scheme dictate, is called a semi-elastic flow. Theoretically, all TCP flows can be modeled as semi-elastic as delays could trigger a timeout.

A flow that has no constraint on required injection rate is called an elastic flow. As an
example a UDP flow can be considered as elastic flow. In our simulations, we consider purely MAC layer frames for the sake of simplicity. But those can also be thought of as MAC layer resultant frames of a UDP stream with lower layer header/trailer added and packets re-arranged in frames of size we choose. We choose 1500 Bytes long frames.

We limit this work to elastic flows. Experiments with inelastic and semi-elastic flows as well as considerations for 802.11 MAC are left as future work.
Chapter 4

System Model

4.1 Terminology

4.1.1 PDQ, Queue and Flow

A PDQ is a queue of packets for a specific destination at some node. Hereafter, the terms PDQ and queue are used interchangeably. The term flow is a set of PDQs for a destination from source to destination.

4.1.2 Backlog

The backlog of a queue is defined as the difference of queue length of a PDQ at current hop and next hop.

4.1.3 Urgency Weight

The urgency weight of a queue is defined as the product of its backlog and instantaneous rate at which it is served.

4.1.4 Max weight or max weight first scheduling

A scheduling policy that gives precedence to queue in a local neighborhood that has the highest backlog is referred to as max weight scheduling or max weight first scheduling in this document. Thus, TDM-maxwt refers to a scheduling policy for TDM that considers queue with highest backlog in a local neighborhood as a scheduling candidate.
4.1.5 MAC Priority

This work considers the notion of MAC priority as a measure of channel width. Despite the similarity in nomenclature, this is different from what is used in 802.11 e [19] and relevant literature. In [29, 11, 19] the term MAC priority is used to represent a priority level that adjusts the CSMA/CA back off contention window i.e. higher priority frame would have shorter contention window. In case of TDM, back-off mechanism is irrelevant. We use MAC priority to represent a candidate for allocated channel width, following above convention.

4.1.6 Activation Vector

We define activation vector to consist of a list of pairs, each pair consisting of a queue scheduled for transmission for a time slot and the channel width it is allocated. The activation vector is a result of scheduling policy as well as channel width allocation scheme per time slot.

4.1.7 Source and Sink

By convention, source is the source of a flow and the last PDQ, i.e. PDQ at the destination node for the destination node is termed as sink. A special property of the sink is that the queue length of the sink is always considered to be 0, i.e., the destination application can always consume packets at the rate they arrive.

4.2 Network Model

4.2.1 Notation

Consider a network represented by a graph \( G(V, E) \) with \( V \) nodes or vertices and \( E \) representing the links.

Let, \( Q^i_d \) and \( q^i_d \) denote the PDQ and its length respectively, for destination \( d \) at node \( i \).

\( n(i, d) \) denote the next hop from node \( i \) for destination \( d \).

\( b^i_d = q^i_d - q^{n(i,d)}_d \) denote the backlog for PDQ for destination \( d \) at node \( i \).

\( r_{i,n(i,d)} \) denote the rate at which PDQ \( Q^i_d \) is served.

\( w^i_d = (q^i_d - q^{n(i,d)}_d) \times r_{i,n(i,d)} \) be the urgency weight of the PDQ \( Q^i_d \).

\( F \) be the set of flows across the network.

\( x_f \) be the injection rate, \( X_f \) be the throughput and \( U_f(X_f) \) be the concave utility function for flow \( f \).

\( \omega^i_d \) be the channel width allocated to PDQ \( d \) at node \( i \) and \( W \) be the total frequency spectrum width.

\( \Omega(.) \) be a function that maps a transmission rate to channel-width satisfying it, so that
\[ \Omega(r_{i,n(i,d)}) = \omega_d^i \]

### 4.2.2 Assumptions

We assume routing to be orthogonal to our scheme. For our experiments we have assumed routes to be fixed. This assumption can be relaxed by having the PDQ data structure also store next hop identifier.

We assume existence of a NIC called GRANWID, capable of channel width allocation in continuous domain. Also, transmission power is not discretized and can take any continuous value. Every node has a dedicated 802.11 NIC for signaling, e.g., neighbor discovery, transmitter-receiver association, channel width claim and notification. This NIC works on a dedicated frequency band, which is orthogonal to the frequency band used for transmission by the PDQs. For theoretical considerations, time is slotted and the channel width allocation algorithm runs at the beginning of every slot.

We assume that every node is capable of transmitting from any of its PDQ simultaneously. Though this seems unrealistic, we defer the discussion about feasibility of realization of our model till next chapter.

Finally, we consider the data plane, which corresponds to the NICs that perform actual transmission/reception of packets from PDQs, to have a very thin MAC. This means that there is no carrier sense or back off. Once set to transmit, the NIC would go on transmitting packets from the PDQ assigned, separated by guard intervals. For example, consider figure 4.1. For the sake of simplicity every link serves a single PDQ in this example. Let \( l_{xy} \) be the PDQ at \( x \) with next hop \( y \) and \( r_{xy} \) be transmission rate at which it is served. Then the figures in boxes denote the backlog of the queue. The urgency weight of queue \( l_{AB} \) is \( 150 \times r_{AB} \).

Now, we turn our attention to different components of our network model.

### 4.2.3 Channel Width And Transmission Rate

According to Shannon’s theorem [14], for a fixed transmission scheme,

\[ r \propto w \tag{4.1} \]

where \( r \) is transmission rate and \( w \) is channel width

Thus, \( \Omega^{-1}(.) \) exists. Also, higher channel width translates to higher transmission rate.

As observed in [8], this linear relationship does not hold in 802.11 networks because the inter-frame spacings are of fixed duration across channel widths. We relax this model by not using 802.11 MAC for data transmission. It is assumed that inter-frame spacings can be calculated
and constant of above proportionality can be adjusted accordingly.

Consider a slotted time scheduling algorithm with time slots of duration \( slotlen \). The scheduling duration for a slot is taken at the beginning of the slot. For the sake of simplicity, we disregard the time overhead \( (\Theta) \) of the algorithm. Note that this does not weaken the performance model, as to take \( \Theta \) into account, we just have to replace \( slotlen \) with \( slotlen + \Theta \).

### 4.2.4 The Interference Minimization Scheme

The relationship between the minimum received power \( s_{\text{min}} \) expressed in dBm required to receive a signal correctly with SINR and channel width \( \omega \) is given as [14, 7]

\[
    s_{\text{min}} = \sinr + 10 \log_{10} (K \times T_0 \times \omega) + N_F \quad (4.2)
\]

Here, \( K \) is Boltzmann’s constant, \( 1.38 \times 10^{20} \), \( N_F \) is receiver noise figure and \( T_0 \) is absolute room temperature, considered \( 290^\circ K \) in our simulations.

Also, the relationship between transmission range, distance and channel width is given as [25, 7]

\[
    \text{dist} = 10 \left( \frac{P_T - P_R - 20 \log_{10} \left( \frac{4\pi\omega d_0}{c} \right)}{10 \alpha} \right) \quad (4.3)
\]

where \( P_R \) is received power, \( d_0 \) is reference distance, \( \alpha \) is path-loss factor and \( c \) is the speed
of light in air. Putting $P_R = s_{\text{min}}$ makes $\text{dist}$ the transmission range $\text{trange}$. Putting $\text{dist} = \text{distance between } i \text{ and } n(i,d)$ enables us to compute the transmission power required for successful transmission with minimum possible transmission range. With this transmission range, we can compute the interference range $\text{irange}$, in keeping with extensive experimental experience reported in literature, as

$$\text{irange} = \text{trange}(1 + \delta) \quad (4.4)$$

Figure 4.2 depicts this relationship.

### 4.3 The Card GRANWID

Now we turn our attention to model the card GRANWID. Going along the observations in [8, 7] we assume that at a width of 80 MHz, GRANWID requires received power to be -63dBm to receive a frame correctly ($s_{\text{min}}$). Based upon this, we can model $s_{\text{min}}$ as a function of channel
width $\omega$ as

$$s_{\text{min}}(\omega) = -63 - 10 \log_{10} \left( \frac{80}{\omega} \right)$$  \hspace{1cm} (4.5)

The transmission power is assumed to be fully discretized and is calculated as

$$P_T = 10\alpha \log_{10} \text{trange} + s_{\text{min}} + 20 \log_{10} \left( \frac{4\pi\omega d_0}{c} \right)$$  \hspace{1cm} (4.6)

GRANWID is assumed to conform to linear relationship between channel width and transmission rate as per Shannon’s theorem. The constant of proportionality depends upon the modulation being used. For the purpose of our experiments, we do not change the modulation scheme. We fix the constant of proportionality $\omega_{\text{const}}$ corresponding to the value required to obtain 54 Mbps transmission rate at 20 MHz channel. In other words,

$$\omega_{\text{const}} = 2.7 \text{Mbps/MHz} = 0.3375 \text{MBps/MHz}$$  \hspace{1cm} (4.7)

Using this we get table 4.1 for $s_{\text{min}}$, $\text{trange}$ and $\text{irange}$ and $P_T = 18 \text{dBm}$ and $\alpha = 4$. We get graph 4.3 for channel width vs transmission power when interference range is fixed at 100 m.

Table 4.1: Variations in Received Power, Transmission Range and Interference Range with Channel Width

<table>
<thead>
<tr>
<th>Width(MHz)</th>
<th>Min Received Power(dBm)</th>
<th>Transmission Range(m)</th>
<th>Interference Range(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>-63.0000</td>
<td>18.2983</td>
<td>25.6176</td>
</tr>
<tr>
<td>60</td>
<td>-64.2493</td>
<td>22.7046</td>
<td>31.7865</td>
</tr>
<tr>
<td>40</td>
<td>-66.0103</td>
<td>30.7739</td>
<td>43.0835</td>
</tr>
<tr>
<td>20</td>
<td>-69.0206</td>
<td>51.7554</td>
<td>72.4576</td>
</tr>
<tr>
<td>10</td>
<td>-72.0309</td>
<td>87.0419</td>
<td>121.8586</td>
</tr>
<tr>
<td>5</td>
<td>-75.0412</td>
<td>146.3864</td>
<td>204.9409</td>
</tr>
</tbody>
</table>

get graph 4.3 for channel width vs transmission power when interference range is fixed at 100 m.
4.4 Lower bound on Throughput

Now we assert a lower bound on throughput of scheduling policy with variable channel width in comparison with that of scheduling policy with fixed channel width. Considering throughput obtained by optimal scheduling policy with fixed channel width as a lower bound, we state our claim as-

Aggregate network throughput obtained using scheduling scheme with optimal channel width assignment policy can be no worse than that of the optimal fixed channel width policy.

We can provide the proof of correctness of above claim as follows.

4.4.1 Corollary I

A scheduling solution with fixed channel width is also a valid solution for variable channel width policy. This is trivially true.

4.4.2 Corollary II

For every instance of fixed channel width allocation, at least one variable channel width allocation instance exists whose aggregate throughput is no worse than that of the fixed channel width allocation instance.

For a set of flows $F'$ in a local neighborhood according to our assumption of available bandwidth,

$$\sum_{f \in F'} \Omega(x_f) = W' \leq W \quad (4.8)$$
Consider a channel width allocation scheme that utilizes complete spectrum in a local neighborhood. Any such scheme that distributes the free channel width of $W - W'$ across PDQs depending upon $U_f(x_f)$ would yield throughput no worse than that of fixed channel width allocation policy. Same can be said of optimal fixed width channel allocation scheduling. Same is true for utility function, since utility function is a concave function of the injection rate.

### 4.4.3 Inference

This theorem enables us to imagine scheduling solutions as a Venn diagram 4.4. But this does not necessarily mean that variable channel width schemes yield strictly better throughput. In fact figure 4.4 can be considered as plot of policies yielding increasing throughput going upwards. This means, as we found out while investigating candidates for channel width allocation, a variable channel width policy designed without proper investigation can still yield a worse throughput. Considering localized neighborhood, this is again intuitive to explain. Assume transmission over $l_{AB}$ can interfere with transmission over all other links. Again consider figure 4.1. Consider using a scheme that allocates channel width proportional to backlog. $l_{BC}$ will be scheduled with a width of $\frac{250 \times 60}{750} = 20$ MHz. Whereas $l_{DA}$ which cannot reach C will get 34.28 MHz. This has defeated utility maximization. Now $l_{DE}$ which cannot interfere with C will get 5.72 MHz. Now, $l_{AB}$ has no spectrum left. Depending upon packet injection from source, this scheme again has a potential to downgrade aggregate throughput.
Chapter 5

Greedy Channel Width Allocation Algorithm

In this chapter we first discuss various design choices for the design of our scheme. Then the algorithms are explained followed by an example for each.

5.1 MAC Priorities

For channel width assignment, we use MAC priority mechanism similar (but not identical) to [29, 11]. Here, linear quantization of MAC priority levels is used. If there are \(\text{maxlevels}\) levels of MAC priorities, \(\text{maxb}\) and \(\text{minb}\) are maximum and minimum weights of queues in neighborhood of a queue with weight \(\text{curr}\), then the MAC priority \(\text{currprio}\) of the current queue is calculated as:

\[
\text{currprio} = \frac{(\text{curr} - \text{minb}) \times \text{maxlevels}}{\text{maxb} - \text{minb}}
\]  

(5.1)

[29] considers 8 different MAC priorities and backlog of a queue as its weight. We use a similar scheme with backlog of a queue as its weight. We define \(\text{maxlevels}\) to be values \(\omega_i\) can take. Thus, in case of Gwidcont, the MAC priority takes any value (continuous) between 0 and 60, which is the spectrum width. On the other hand, Gwiddis allows 4 different MAC priorities, corresponding to permissible channel widths, 5, 10, 20 and 40 MHz.

5.2 Utility-based Rate Control or Source Rate Control

The injection rate of a flow should be controlled based upon its utility. We use similar mechanism as in [4, 11] at source node to control the flow injection rate. Packets generated at a
source node $src$ are injected into the queue for destination $d$ as long as

$$U'_f(x_f) > \beta \times q_{d}^{src}$$

(5.2)

where $U'_f(x_f)$ is the first derivative of the utility function of a flow applied to the instantaneous injection rate. The term source rate control is alternatively used to refer to this utility based rate control mechanism.

[5] proved that up to a certain limit, the user satisfaction of web browsing is proportional to the log of throughput obtained. Since then log $X_f$ has been used in many other works as the choice for the utility function [4, 29, 11]. We conform with this convention. $\beta$ is a small constant. Its value is 0.00000001 for our experiments.

Based upon our interference minimization scheme, we now design a greedy channel allocation algorithm for continuous channel width allocation. The source node first decides how to inject packets into the PDQ. For this, all the packets generated are enqueued in a queue called genq. We assume packet size to be fixed at $p$ bytes. The number of packets to be injected into the PDQ in a time slot $nadd$ at source is governed by the rate control as explained before. We have

$$x_f = \frac{nadd}{slotlen}$$

(5.3)

$$U'(x_f) = U'(\log_{10} x_f)$$

(5.4)

$$= \frac{1}{x_f \ln(10)}$$

(5.5)

$$= \frac{slotlen}{nadd \times \ln(10)}$$

(5.6)

If $qlen$ is current length of PDQ (in number of packets), then the maximum value of ($nadd$) is constrained as

$$\frac{slotlen}{nadd \times \ln(10)} > \beta(nadd + qlen)$$

(5.7)

$$(nadd)^2 + \beta qlen \ln(10) \times nadd - \frac{slotlen}{\beta \ln(10)} < 0$$

(5.8)

We form and solve the quadratic equation of $nadd$ and set $nadd$ as the positive root.

Thus, we get following algorithm for source rate control.
Algorithm 1: Source Rate Control

\[
\begin{align*}
a & \leftarrow 1 \\
b & \leftarrow \beta qlen \ln(10) \\
c & \leftarrow -\frac{\text{slotlen}}{\beta \ln(10)} \\
nadd & \leftarrow \frac{-b + \sqrt{b^2 - 4ac}}{2a} \\
\text{if } nadd > \text{genq.length} \text{ then} & \\
& \quad nadd \leftarrow \text{genq.length} \\
\text{end if} \\
\text{return } nadd
\end{align*}
\]

5.3 Testing for Interference

In absence of 802.11 we cannot rely on career sense or RTS/CTS mechanism for ensuring non-interfering transmission in local neighborhood. A queue 'A' can interfere with transmission of queue 'B' if the distance between transmitter of 'A' and receiver of 'B' is less than irange of 'A'. Note that this removes exposed terminal problem as well as hidden terminal problem. To illustrate this, consider transmitter receiver pairs (A, B) and (C, D). Let A has higher backlog compared to C. Further assume that A and C are outside each other’s interference range. Now, if transmission from A and C can collide at B (hidden terminal), above test for interference will dictate that C should not transmit on the channel used by A. Now assume A and C lie in each other’s interference range, but B and D are outside interference ranges of C and A respectively (exposed terminal). In this case, above test would dictate that the transmission over links AB and CD would not interfere. Hence channel reuse is possible.

5.4 Choice of Time Slot Duration

With the transmission rate conforming to 54 Mbps at 20 MHz, a node can transmit approximately 14,000 packets of size 1500 Bytes per second utilizing complete 60 MHz spectrum. We found out that with large number of packets being transmitted, the difference between fairness experienced by the schemes was reduced to an extent. On the other hand, if the time slot is too low, e.g. 1 ms, then the node cannot transmit enough packets to have reasonably accurate scheduling decision. This results in very slow converging of injection rate and throughput. As per our observations, 1 ms time slot took more than 10,000 simulated seconds to converge, while simulation for 1 s time slot converged in approximately 100 s. By trial and error, we settled on 10 milisecond time slot.
5.5 Calculating the Activation Vector

As we mentioned above, the activation vector is a product of scheduling as well as channel allocation. Scheduling is simple. Queues are simply maintained in sorted order based upon their backlogs, maximum backlog first. While traversing this array, a queue is selected for transmission only if some spectrum is available. The width assigned is a minimum of available width and the width as dictated by channel allocation. We now discuss candidates for channel allocation.

5.5.1 Greedy Backlog Reversing

This greedily requests a width upto \( \Omega(b_d^i) \). If channel and packets both are available, the queue backlog becomes negative with same magnitude. This scheme is called greedy backlog reversing because of its behavior in case channel width requested by a queue is available, and no new packets are scheduled to arrive to the queue. In this case, after transmission at the beginning of the next time slot, the backlog of the queue becomes negative with the same magnitude, i.e. \(-b_d^i\). As expected, this scheme proved to be very unfair and no significant improvement in throughput was observed compared to simple TDM-maxwt. We chose to discard this scheme.

5.5.2 Local Proportional Channel Width Assignment

This is a scheme that assigns channel width to flows based upon proportion of total backlog in local neighborhood. This scheme sums up backlogs of queues in local neighborhood as \( b_{sum} \). The channel width requested is \( \Omega\left(\frac{b_d^i}{b_{sum}}\right) \). This scheme did not improve fairness much, but underutilized the spectrum in a local neighborhood. So, the throughput does not improve. This is again intuitively obvious. If queue 'A' leaves 35 MHz for other queues in neighborhood, it is possible that those queues will not be scheduled because of other higher priority queues not in neighborhood of 'A'. As an end result, the 35 MHz spectrum is simply wasted. We discarded this scheme as well.

5.5.3 MAC Priorities

As already explained in section 4.1.5, this scheme uses linear quantization to map queue backlog into requested channel width. This scheme is clearly not optimal. One very obvious sub-optimality comes from the observation that the queue with maximum backlog in a local neighborhood may get complete available spectrum if enough packets are available for transmission. Nevertheless, our observations show an improvement in throughput and fairness. This is our scheme of choice.
This enables us to calculate assigned width as follows. Note that \( NPDQ \) is the number of PDQs in network sorted in descending order of backlogs. We assume that, for a wireless mesh network, the nodes have an idea of neighborhood within the interference range. As noted by [22], this can be achieved by maintaining long term average of transmission power vs received power. The neighborhood of a node with longest hop distance \( h_{\text{max}} \) would be \( h_{\text{max}}(1 + \delta) \). So every node has to maintain a neighbor list of nodes within this distance. Every node has to keep a track of channel width available in the neighborhood.

### Algorithm 2: Allocate Channel Width-Continuous

\[
\begin{align*}
\text{min} & \leftarrow \text{minimum backlog in neighborhood} \\
\text{max} & \leftarrow \text{maximum backlog in neighborhood} \\
i.\text{remwid} & \leftarrow \text{unclaimed bandwidth in neighborhood} \\
\text{for} \text{ queues 1 to i do} \\
\quad \text{if} \ j.\text{assigned} > 0 \ \text{and} \ \text{dist}(i.tx, j.rx) > \text{irange} \ \text{and} \ j.\text{remwid} < \text{remwid} \ \text{then} \\
\qquad i.\text{remwid} & \leftarrow j.\text{remwid} \\
\text{end if} \\
\text{end for} \\
\text{req} & \leftarrow \frac{(b_i - \text{min}) \times 60}{\text{max} - \text{min}} \\
\text{if} \ \text{req} < i.\text{remwid} \ \text{then} \\
\quad \text{req} & \leftarrow \text{remwid} \\
\text{end if} \\
\text{if} \ \text{req} > \Omega(i.\text{length}) \ \text{then} \\
\quad \text{req} & \leftarrow \Omega(i.\text{length}) \\
\text{end if} \\
i.\text{assigned} & \leftarrow \text{req} \\
\text{for} \ \text{queues j from i to NPDQ do} \\
\quad \text{if} \ \text{dist}(i.tx, j.rx) > \text{irange} \ \text{then} \\
\qquad j.\text{remwid} & \leftarrow j.\text{remwid} - \text{min}(j.\text{remwid}, i.\text{remwid}) \\
\text{end if} \\
\text{end for}
\end{align*}
\]

#### 5.6 The Algorithm \( \text{Gwid} \)

Now we formalize our channel width allocation scheme. For the purpose of our simulations, we have assumed that a central controller which takes care of TDM synchronization, runs the following algorithm. It is trivial to convert the activation vector calculation into a broadcast based distributed one. This part is not considered for our experiments.
Algorithm 3: Gwid - Continuous Spectrum Allocation

\[ NQ \leftarrow \text{list of queues in the network in descending order of backlog} \]

\[ \text{for queues 0 to NPDQ do} \]
\[ \quad \text{if i.tx is source then} \]
\[ \quad \quad nadd \leftarrow \text{Source Rate Control} \]
\[ \quad \quad \text{append } nadd \text{ packets to queue} \]
\[ \quad \text{end if} \]
\[ \text{end for} \]
\[ \text{for queues 0 to NPDQ do} \]
\[ \quad \text{Allocate channel Width} \]
\[ \text{end for} \]
\[ \text{Begin Transmission for duration slotlen} \]

5.7 Discrete Channel Width Allocation

Now we provide approximation of Gwid for 802.11 n [2] channels. Here, the channel widths are limited to 5, 10, 20 and 40 MHz channels. We choose these channel widths for realism, since these are the standard available channel widths in IEEE 802.11 n [2] standard. Thus, we have only 4 priority levels. The structure of the algorithm remains the same as 3. The only different algorithm is channel width allocation.
Algorithm 4: Allocate Channel Width-Discrete

\[
\begin{align*}
\text{min} & \leftarrow \text{minimum backlog in neighborhood} \\
\text{max} & \leftarrow \text{maximum backlog in neighborhood} \\
i\.\text{remwid} & \leftarrow \text{unclaimed bandwidth in neighborhood} \\
\text{widarr} & \leftarrow [5, 10, 20, 40] \\
\text{for} & \text{ queues 1 to i do} \\
& \text{ if } j\.\text{assigned} > 0 \text{ and dist}(i\text{.tx}, j\text{.rx}) > \text{irange} \text{ and } j\.\text{remwid} < \text{remwid} \text{ then} \\
& \qquad i\.\text{remwid} \leftarrow j\.\text{remwid} \\
& \text{ end if} \\
\text{ end for} \\
\text{reqind} & \leftarrow (b_i - \text{min}) \times 4 \frac{\text{max} - \text{min}}{\text{max} - \text{min}} \\
\text{req} & \leftarrow \text{widarr}[\text{reqind}] \\
a & \leftarrow 0 \\
\text{while } & \text{widarr}[\text{reqind} - a - 1] > \Omega(i\.\text{length}) \text{ do} \\
& \quad a \leftarrow a + 1 \\
\text{end while} \\
\text{if } & \text{req} < i\.\text{remwid} \text{ then} \\
& \quad \text{req} \leftarrow \text{element of widarr closest to and less than remwid} \\
\text{end if} \\
i\.\text{assigned} & \leftarrow \text{req} \\
\text{for } & \text{ queues j from i to NPDQ do} \\
& \text{ if dist}(i\text{.tx}, j\text{.rx}) > \text{irange} \text{ then} \\
& \qquad j\.\text{remwid} \leftarrow j\.\text{remwid} - \min(j\.\text{remwid}, i\.\text{remwid}) \\
& \text{ end if} \\
\text{end for}
\end{align*}
\]
5.8 Example

To help understand the two schemes better, we provide examples of each case. Consider figure 5.1. For the sake of simplicity consider that each flow spans two hops in anti-clockwise fashion. Every node can interfere with reception of all others. 60 MHz spectrum is available. The card allows transmission of one packet per MHz of spectrum in one time slot. Also, every source injects 15 packets per time slot. Initially all queues are empty. Note that sinks are not shown.

5.8.1 Gwid-cont

After first time slot \( q_A^C, q_B^D, q_C^A \) and \( q_B^D \) all have 15 packets each. For \( q_A^C \) requests width is 60MHz. But since it has only 15 packets, the width allocated will be tuned down to \( \Omega(15) \), i.e. 15 MHz. Similarly, other three queues will also get 15 MHz each. After injecting packets after second slot all the eight queues have 15 packets each. Now the queues with receiver at destination, that is \( q_B^D, q_B^D, q_B^D \) have a backlog of 15 packets each and other queues have a backlog of 0 packets. So these are scheduled for transmission. Now after injecting packets \( q_A^C, q_B^D, q_C^A \) have a backlog (and queue length) of 30 packets each. Now \( q_A^C \) and \( q_B^D \) will be scheduled with 30 MHz each, etc.

5.8.2 Gwid-dis

Again, after first time slot \( q_A^D, q_D^C, q_A^C \) and \( q_B^D \) all have 15 packets each. This makes each node at priority 4 and requested width 60. But a 15 packet transmission can be satisfied with 20 MHz. So queues \( q_A^C, q_D^C, q_A^C \) are scheduled with 20 MHz each. Now, after injecting packets, \( q_B^D \) has 30 packets, \( q_B^D \) is empty and other queues have 15 packets each. So, \( q_B^D \) is scheduled with 40 MHz and \( q_B^D \) is scheduled with 20 MHz. Now after injecting packets, we have \( q_B^D, q_A^D, q_B^D \) and \( q_A^D \) with 30 packets \( q_B^D \) empty and other queues with 15 packets each. Now, \( q_B^D \) gets 40 MHz and \( q_B^D \) gets 20 MHz, etc. Contrasting with the above illustrates the difference between the two channel width allocation algorithms.
5.9 Realization Feasibility

The closest approximation for a single node transmitting at different channels with proprietary NICs can be a multi-radio node. But for continuous channel width allocation scheme to always be feasible, this would mean having arbitrary number of radios or being able to dynamically add another radio in negligible time. Both of these assumptions are unrealistic.

An implicit assumption about encoding scheme is that guard frequency interval is not needed between adjacent channels. One way to realize this is like OFDM, dividing channel into orthogonal sub-channels. But this itself discretizes the unit channel allocation to a single sub-channel. So, a continuous channel width allocation scheme is practically unrealizable with today’s NICs. However, with software defined radios, some other encoding schemes might be possible which would allow continuous channel width transmission.

One more assumption we make while using interference minimization framework is that transmission power can be varied in continuous domain independently of the channel width and transmission rate. This assumption also does not comply to today’s NICs those support discrete power levels depending upon the width and rate supported. And these values also vary with modulation supported. While we assume a fixed modulation scheme, continuous and independent varying of channel width and transmission power for a given transmission rate is still very strong assumption.

The discrete adaptive channel allocation can be approximated with multiple radios. If we fix the channel widths to 5, 10, 20 and 40 MHz, then the worst case number of radios requires
for 60 MHz spectrum is 12, which is abnormal, but not impossible. In this case the only limiting factor would be TDM. So, a CSMA/CA approximation of Gwidd-dis is vaguely feasible to be realized with 12 NICs per node.

Lastly, we note that with the advancements in software defined radios, a single SDR card (with sufficient sampling rate) could be used to transmit different data streams on different channels (by computing the superposed waveform and constructing it with samples).
Chapter 6

Experimental Setup

6.1 Metrics of Interest

1. Throughput:

   This is average throughput per flow as well as average throughput across the network. With provision for better spectrum utilization, we expect to get better throughput with both Gwid continuous and discrete channel width allocation schemes compared to fixed channel width allocation scheme.

2. Aggregate Network Utility:

   Network utility is again the sum of utility function of flows across the network over a given time. The choice of utility function is in line with [5], which proved that upto a saturation point, user satisfaction for web browsing is proportional to logarithm of assigned bandwidth. So we compute

   \[ \sum_f \log_{10}(x_f) \]

   This metric has been used by studies [4], [29] and [11].

3. Fairness in Bandwidth Distribution:

   We use Jain’s fairness index [10] to measure fairness of bandwidth distribution. It measures fairness as

   \[ \frac{\left( \sum_{i=1}^{\left| F \right|} x_i \right)^2}{\sum_{i=1}^{\left| F \right|} x_i^2} \]

   where \( F \) is the set of all flows across the network and \( x_i \) is the throughput of flow \( i \). The index lies between 0 and 1. The closer the index to 1, the fairer the throughput.
distribution.

4. Pure injection rate:

This is the sum of packets injected by the source node into PDQ across the network per time slot.

\[ \sum \limits_{q \in F} n_{\text{inj}}^q \]

5. Buffer Occupancy:

For each node we measure the buffer occupancy as

\[ \sum \limits_{Q \in F'} q_d^i \]

where \( F' \) is set of flows passing through node \( i \). The back-pressure scheduling/congestion control attempts to stabilize queue lengths across network. So we expect to get stable buffer utilization per node as well as per flow.

6. Pure Spatial Reuse:

This means the number of PDQs scheduled for transmission across the network.

7. Air Time Utilization:

This is the number of packets transmitted per time slot across the network.

### 6.2 Cases to Compare

We compare the algorithms \( Gwid - cont \) and \( Gwid - dis \) against the algorithms \( TDM - \text{maxwt} \) and \( TDM - \text{rand} \). \( TDM - \text{maxwt} \) and \( TDM - \text{rand} \) use fixed 20 MHz channels. While \( TDM - \text{maxwt} \) schedules the links with maximum backlogs for transmission per time-slot, \( TDM - \text{rand} \) picks links randomly per time slot. The comparison with \( TDM - \text{maxwt} \) represents benefit obtained in back-pressure scheduling/congestion control scheme because of channel-width adaptation. On the other hand, comparison with \( TDM - \text{rand} \) represents benefit obtained by both, the back-pressure framework as well as channel-width adaptation.

### 6.3 Simulation Methodology

Because of difficulty of having arbitrary number of cards or dynamically adding a card if required, Opnet or NS were not feasible option to simulate continuous channel allocation scheme. So we chose to write a custom simulator in C to evaluate the performance of our schemes and
compare it with the cases $TDM − maxwt$ and $TDM − rand$. A rough sketch of components of simulator have been shown in figure 6.1. Next we give a brief description

6.3.1 Packet Generation

Packet generation is implemented as a poisson packet arrival process with mean number of packets generated per time slot as a parameter listed.

6.3.2 Rate Control

This is the same rate control algorithm as algorithm 1. For $TDM − rand$ there is no source rate control. This is done by not running the rate control and injecting all packets generated into the PDQ at source.

6.3.3 Arrange PDQs

This is just a simple sort in descending order of backlog. For $TDM − rand$ links are to be picked randomly. This is achieved by randomly shuffling the priority queue of the PDQs and marking only the first PDQ picked for a link as a candidate for scheduling.

6.3.4 Activation Vector

A set of links and assigned channel width for the time slot becomes activation vector. Algorithms 2 and 4 state how it is calculated for variable channel width allocation case. For $TDM − maxwt$ the algorithm changes slightly as algorithm 5 For $TDM − rand$ The basic structure remains similar to $TDM − maxwt$, but the array $NQ$ which is sorted list of PDQs in descending order of other three, now becomes a randomly shuffled array. The shuffle marks only the first PDQ encountered from a link as algorithm 6.

6.3.5 Transmission

Since we have already taken care of interference this is simple packet transfer as algorithm 7
Figure 6.1: Simulation Components
Algorithm 5: Allocate Channel Width-Fixed

\( \min \leftarrow \text{minimum backlog in neighborhood} \)
\( \max \leftarrow \text{maximum backlog in neighborhood} \)
\( \text{i}.\text{remwid} \leftarrow \text{unclaimed bandwidth in neighborhood} \)
\( \text{for queues 1 to i do} \)
\( \quad \text{if } \ j.\text{assigned} > 0 \text{ and } \text{dist}(\text{i}.\text{tx}, \text{j}.\text{rx}) > \text{irange} \text{ and } \text{j}.\text{remwid} < \text{remwid} \text{ then} \)
\( \quad \quad \text{i}.\text{remwid} \leftarrow \text{j}.\text{remwid} \)
\( \quad \text{end if} \)
\( \text{end for} \)
\( \text{req} \leftarrow 20\text{MHz} \)
\( \text{if } \text{req} < \text{i}.\text{remwid} \text{ then} \)
\( \quad \text{req} \leftarrow \text{element of widarr closest to and less than remwid} \)
\( \text{end if} \)
\( \text{i}.\text{assigned} \leftarrow \text{req} \)
\( \text{for queues j from i to NPDQ do} \)
\( \quad \text{if } \text{dist}(\text{i}.\text{tx}, \text{j}.\text{rx}) > \text{irange} \text{ then} \)
\( \quad \quad \text{j}.\text{remwid} \leftarrow \text{j}.\text{remwid} – \min(\text{j}.\text{remwid}, \text{i}.\text{remwid}) \)
\( \quad \text{end if} \)
\( \text{end for} \)

Algorithm 6: TDM - Random

\( \text{NQ} \leftarrow \text{list of queues in the network shuffled randomly} \)
\( \text{for queues 0 to NPDQ do} \)
\( \quad \text{if } \text{i}.\text{tx} \text{ is source then} \)
\( \quad \quad \text{nadd} \leftarrow \text{genq.length} \)
\( \quad \quad \text{append nadd packets to queue} \)
\( \quad \text{end if} \)
\( \text{end for} \)
\( \text{for queues 0 to NPDQ do} \)
\( \quad \text{Allocate channel Width-Fixed} \)
\( \text{end for} \)
Begin Transmission for duration slotlen

Algorithm 7: Transmission

\( \text{NQ} \leftarrow \text{list of queues in the network} \)
\( \text{for queues 0 to NPDQ do} \)
\( \quad \text{if } \text{i}.\text{assigned} \neq 0 \text{ then} \)
\( \quad \quad \text{Dequeue } \Omega^{-1}(\text{i}.\text{assigned}) \text{ packets from i} \)
\( \quad \quad \text{Append } \Omega^{-1}(\text{i}.\text{assigned}) \text{ packets to queue } q_{\text{n}(\text{i}.\text{tx}, \text{i}.\text{dest})}^{\text{n}(\text{i}.\text{tx}, \text{i}.\text{dest})} \)
\( \quad \text{end if} \)
\( \text{end for} \)
6.4 Network Setup

We evaluate the schemes in case of a grid topology and in case of 30 random topologies.

6.4.1 Topology Grid

Here we have a square 7×7 grid. Consecutive nodes along a side are separated by 100m. We then select 10 source-destination pairs at random. The routes are chosen to be simple Manhattan so as to increase sharing of links. Table 6.1 lists the flows used for these experiments. The source-destination pairs were picked randomly.

6.4.2 Random Topologies

We generated 30 topologies using a 2-D poisson point process with mean 50 nodes in a 600m × 600m square area. These topologies are depicted in the appendix. This number of topologies was intended to better calculate the confidence intervals over measured parameters using a normal distribution with Chi-square method. Then for each topology, 10 source-destination pairs were randomly chosen. The routes were formed using shortest path routing. To establish connectivity, a minimum spanning tree was constructed and the longest distance link was noted. Then we formed node neighborhood considering this distance as node range. On this neighborhood set and source-destination pairs, shortest path routing was applied to get final flow description. Means for packet arrival rate were, again, picked randomly between 1000 and 1700 packets per slot. The confidence interval $ci$ for a mean $\mu$ and standard deviation $sd$ for 30 runs is calculated as-

\[
    ci = 1.96 \times \frac{sd}{\sqrt{30}}
\]

The confidence interval becomes [$\mu - ci, \mu + ci$].

33
Table 6.1: Flow Setup - Topology Grid

<table>
<thead>
<tr>
<th>Flow No</th>
<th>Source</th>
<th>Destination</th>
<th>Generation rate (Packets/slot)</th>
<th>Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28</td>
<td>2</td>
<td>1410</td>
<td>28, 21, 22, 15, 8, 9, 2</td>
</tr>
<tr>
<td>2</td>
<td>43</td>
<td>4</td>
<td>1580</td>
<td>43, 36, 37, 30, 31, 24, 25, 18, 11, 4</td>
</tr>
<tr>
<td>3</td>
<td>46</td>
<td>21</td>
<td>1112</td>
<td>46, 39, 38, 31, 30, 23, 22, 21</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>29</td>
<td>1432</td>
<td>7, 14, 15, 22, 29</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>30</td>
<td>1042</td>
<td>5, 12, 11, 18, 25, 24, 31, 30</td>
</tr>
<tr>
<td>6</td>
<td>21</td>
<td>31</td>
<td>1586</td>
<td>21, 22, 29, 30, 31</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>34</td>
<td>1273</td>
<td>8, 9, 10, 17, 18, 25, 26, 33, 34</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>41</td>
<td>1530</td>
<td>5, 12, 19, 20, 27, 33, 40</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>41</td>
<td>1750</td>
<td>0, 7, 8, 15, 16, 23, 24, 31, 32, 39, 40, 41</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>47</td>
<td>1129</td>
<td>1, 8, 9, 16, 17, 24, 25, 32, 33, 40, 47</td>
</tr>
</tbody>
</table>
Chapter 7

Results

Now we summarize the results of our experiments.

7.1 Topology - Grid

7.1.1 Aggregate Injection Rate

![Aggregate Injection Rate](image)

Figure 7.1: Grid - Pure injection rate

The aggregate injection rate fig 7.1 is a good measure to better understand the point at
which the simulation stabilizes, that is, the point after which injection rates and throughput converge. This also helps us to understand how long the simulation should be run so as to get more accurate numerical results. As we can see the injection rates converge at around 1000 second mark. Thus validating our choice of running simulation for 10000 simulated seconds.

7.1.2 Per Flow Injection Rate

![Graph showing injection rates for different flows](image.png)

Figure 7.2: Grid - Per flow Injection Rate

Per-flow injection rate is an indication of buffer utilization stability. If injection rate of a flow does not exceed its throughput, it indicates buffer size for corresponding queues does not grow. Fig 7.2 compared with fig 7.7 verifies that in our experiments queues have stabilized. Also, we get higher injection rates for variable channel width assignment. On an average, injection rates for Gwid-cont were higher than Gwid-dis, following the trend shown by throughput.
7.1.3 Per Flow Delay

Back-pressure framework can provide optimal throughput, average per flow packet delay can be very high. This delay depends upon number of hops. We, indeed see an increase in delay with number of hops, but at the same time, the delays suffered in variable channel width schemes are significantly lower than TDM-maxwt or TDM-rand scheme. In general the delay for Gwid-dis is found to be less than that of Gwid-cont. This is because of a cap on maximum channel width, Gwid-dis scheduled more links as evident from fig 7.5. Hence packets tend to reach destination in shorter chunks, but faster. Hence, even though Gwid-cont has more throughput compared to Gwid-dis, it has slightly higher delay. The aggregate network delay fig 7.4 is not representative measure for a single experiment, but this, nevertheless can be used to check the confidence limits for delay averaged over multiple runs.
7.1.4 Spatial Reuse

It was intuitive that variable channel width solutions would be able to schedule more queues compared to fixed channel width allocation. Also, in absence of any cap on maximum channel width, queue with local maximum backlog may hog complete spectrum preventing other queues from being scheduled. So, number of queues scheduled should be higher in case of Gwid-dis compared to Gwid-cont. Note that this factor is mitigated in case of uniform neighborhood topology like a grid. So, this difference is observed to be small.
7.1.5 Air time Utilization

This is another measure of how fast PDQs move forward. This has a direct correlation with average network throughput and also with number of links scheduled. As expected Gwid-cont moved forward highest number of packets followed closely by Gwid-dis. Fixed channel width schemes fail to transmit those many packets.
7.1.6 Throughput

The motivation for variable channel width allocation was to obtain higher throughput. As expected, due to higher amount of packets transmitted per time slot combined with more number of links scheduled, per flow throughput was significantly higher in case of variable channel width allocation. In particular, Gwid-cont achieved the highest throughput. This factor is particular of uniform neighborhood topology, like grid. This is because of the fact that the disparity in neighborhood, where locally maximum backlogged queue would hog complete spectrum in case of Gwid-cont, is a scenario mitigated in case of grid. In other words, in case of Gwid-cont, local maximum prevents, on an average, less number of other queues from being scheduled in grid topology compared to non-uniform neighborhood topologies.
7.1.7 Aggregate Network Utility

Aggregate Network Utility unsurprisingly follows the trend similar to average throughput. This is because network utility is a concave function of throughput.
7.1.8 Buffer Occupancy

One of the basic properties of the backpressure framework is to have low buffer utilization. Low buffer utilization is also a measure of packets moving forward swiftly, that is low buffering time. Because of more number of links scheduled, the per node buffer utilization is lowest in the case of Gwid-dis followed by Gwid-cont.

Network buffer utilization is another indicator of converging injection rates, albeit averaged over each node that has some PDQ.

Figure 7.10: Grid - Per Node Buffer Utilization
7.1.9 Fairness

As expected variable channel width allocation schemes improve fairness over TDM-maxwt. The factor of locally maximum backlogged queue hogging spectrum, though mitigated, reduces fairness of Gwid-cont compared to Gwid-dis. At the same time, TDM-rand has good fairness because on an average every link gets equal chance to transmit. So the fairness increase comes at the cost of severely reduced throughput.
7.2 Random Topologies

Now we list and explain the results averaged over 30 runs with confidence limits.

7.2.1 Aggregate Injection Rate

The aggregate injection rate again converges near 1000 sec mark.
As expected, Gwid-dis has the best delay properties. This is because with increase in non-uniformity, having upper limit on allocated channel width prevents locally maximum backlogged queue from blanking out spectrum for other queues.
7.2.3 Spatial Reuse

As expected Gwid-dis has the best spatial reuse scheduling maximum number of queues in a time slot. The difference between Gwid and Gwid-cont, though with overlapping confidence intervals, is now more than that in case of a grid.
7.2.4 Air time Utilization

The difference between Gwid-cont and Gwid-dis is less significant compared to other metrics. This is because with discretized channel width, it is still possible to waste some spectrum in case a queue can get scheduled but does not have enough packets to transmit. Note that this case can arise even with high packet arrival rates when an intermediate queue has transmitted enough packets but has not received enough packets to make up for the spare channel width. Also, discretization due to packet size also wastes some (though very small) amount of channel width.
As explained while discussing Grid case, the throughput achieved by Gwid-dis is the highest. Hence our intuition that some upper limit on maximum channel width is required is validated.
7.2.6 Network Utility

Network Utility expectedly follows the trend shown by throughput.
7.2.7 Buffer Occupancy

Again, buffer occupancy has direct correlation with the number of links scheduled. This means, the scheme that scheduled more queues per time slot has lower buffer utilization per node.
The network buffer occupancy again indicates the point where the simulation stabilizes, that is, a balance is established between queue ingress and egress rates and thus, between source injection rate and throughput.
7.2.8 Fairness

![Bar chart showing fairness index for different methods]

Figure 7.21: Fairness

Gwid-dis has the highest fairness. The fairness of TDM-rand drops and confidence limits also become large because the uniformity in neighborhood disappears.
Chapter 8

Conclusion and Future Work

8.1 Conclusion

We have proposed the approach of using variable width channels to efficiently use wireless mesh network spectrum in conjunction with network utility maximization using back-pressure, and designed scheduling and channel width allocation algorithms for this purpose. Our results show that this approach can indeed provide performance benefits.

8.1.1 Throughput and Utility

Per flow throughput does not necessarily grow with increase in granularity of channel width allocation. Though discrete channel allocation may waste some spectrum, it is offset in case of non uniform neighborhood by the phenomenon of local maximum. This mandates that with channel allocation scheme like ours, continuous channel allocation does not scale well. At the same time, this also validates the need to research channel width allocation schemes more because firstly, discrete variable channel allocation still performs better than static allocation. Secondly, air time utilization and spatial utilization of continuous channel width allocation scheme is still comparable to that of discrete scheme.

Network utility is concave function of throughput. So, it follows the trend of throughput.

8.1.2 Fairness

Intuitively, increased granularity of channel width allocation should represents increased fairness, as the transmission rate resulting, now, can correspond better to the relative backlog of a queue in its local neighborhood. But, it is a general observation that channel widths discretized to 802.11 n [2] channels have higher fairness index. This can be explained as follows. In a congested neighborhood, where multiple queues with comparable backlogs are present, in case
of fully granular channel width allocation, the higher priority nodes will be scheduled with high widths and the nodes with comparable backlogs may starve. With discretized channel width allocation, the channel allocation cap is set at 40 MHz, and the remaining width could be distributed among other queues in neighborhood. This logic is backed up by the observation that Gwid-dis schedules more links on an average compared to Gwid-cont.

8.1.3 Delay

Adaptive channel width allocation schemes were found to reduce average per packet delay. However, the dependency of delay on number of hops could not be eliminated by our schemes.

8.2 Future Work

The schemes we discussed so far assume TDM framework. A paralles scheme for CSMA/CA networks needs to be developed. Such a scheme would be capable of being implemented on proprietary 802.11 NICs.

Also, we do not consider the limitations posed by hardware on parallel transmissions. A multi-radio node can remove this impediment to an extent and discretized channel width allocation can provide a crude cap on number of radios required per node (12). A variant for single-radio networks or for nodes with fixed number of radios is left as future work.
REFERENCES


Appendix A

Random Topologies and Simulation Results

We have studied our algorithms for a wide range of topologies and network scenarios. These results are consistent with and similar to the results already presented. For completeness, we provide the detailed results of these additional experiments here.

A.1 Topologies

Figure A.1: Random Topology 1
Figure A.2: Random Topology 2

Figure A.3: Random Topology 3
Figure A.4: Random Topology 4

Figure A.5: Random Topology 5
Figure A.6: Random Topology 6

Figure A.7: Random Topology 7
Figure A.8: Random Topology 8

Figure A.9: Random Topology 9
Figure A.10: Random Topology 10

Figure A.11: Random Topology 11
Figure A.12: Random Topology 12

Figure A.13: Random Topology 13
Figure A.14: Random Topology 14

Figure A.15: Random Topology 15
Figure A.16: Random Topology 16

Figure A.17: Random Topology 17
Figure A.18: Random Topology 18

Figure A.19: Random Topology 19
Figure A.20: Random Topology 20

Figure A.21: Random Topology 21
Figure A.22: Random Topology 22

Figure A.23: Random Topology 23
Figure A.24: Random Topology 24

Figure A.25: Random Topology 25
Figure A.26: Random Topology 26

Figure A.27: Random Topology 27
Figure A.28: Random Topology 28

Figure A.29: Random Topology 29
Figure A.30: Random Topology 30
A.2 Results

A.2.1 Random Topology 1

Figure A.31: Random Topology 1 - Pure injection rate
Figure A.32: Random Topology 1 - Per flow Injection Rate
Figure A.33: Random Topology 1 - Per Flow Delay
Figure A.34: Random Topology 1 - Spatial Reuse
Figure A.35: Random Topology 1 - Air Time Utilization
Figure A.36: Random Topology 1 - Per flow Throughput
Figure A.37: Random Topology 1 - Average Throughput
Figure A.38: Random Topology 1 - Aggregate Utility
Figure A.39: Random Topology 1 - Per Node Buffer Utilization
Figure A.40: Random Topology 1 - Network Buffer Utilization
Figure A.41: Random Topology 1 - Fairness Index
A.2.2 Random Topology 2

Figure A.42: Random Topology 2 - Pure injection rate
Figure A.43: Random Topology 2 - Per flow Injection Rate
Figure A.44: Random Topology 2 - Per Flow Delay
Figure A.45: Random Topology 2 - Spatial Reuse
Figure A.46: Random Topology 2 - Air Time Utilization
Throughput (Mbps)

Flow #

Gwid-cont
Gwid-dis
TDM-maxwt
TDM-rand

Figure A.47: Random Topology 2 - Per flow Throughput
Figure A.48: Random Topology 2 - Average Throughput
Figure A.49: Random Topology 2 - Aggregate Utility
Figure A.50: Random Topology 2 - Per Node Buffer Utilization

Average # of packets Buffered per Node

- Gwid-cont
- Gwid-dis
- TDM-maxwt
- TDM-rand

H
Figure A.51: Random Topology 2 - Network Buffer Utilization
Figure A.52: Random Topology 2 - Fairness Index
A.2.3 Random Topology 3

Figure A.53: Random Topology 3 - Pure injection rate
Figure A.54: Random Topology 3 - Per flow Injection Rate
Figure A.55: Random Topology 3 - Per Flow Delay
Figure A.56: Random Topology 3 - Spatial Reuse
Figure A.57: Random Topology 3 - Air Time Utilization
Figure A.58: Random Topology 3 - Per flow Throughput
Figure A.59: Random Topology 3 - Average Throughput
Figure A.60: Random Topology 3 - Aggregate Utility
Figure A.61: Random Topology 3 - Per Node Buffer Utilization
Figure A.62: Random Topology 3 - Network Buffer Utilization
Figure A.63: Random Topology 3 - Fairness Index
A.2.4 Random Topology 4

Figure A.64: Random Topology 4 - Pure injection rate
Figure A.65: Random Topology 4 - Per flow Injection Rate
Figure A.66: Random Topology 4 - Per Flow Delay
Figure A.67: Random Topology 4 - Spatial Reuse
Figure A.68: Random Topology 4 - Air Time Utilization
Figure A.69: Random Topology 4 - Per flow Throughput
Figure A.70: Random Topology 4 - Average Throughput
Figure A.71: Random Topology 4 - Aggregate Utility
Figure A.72: Random Topology 4 - Per Node Buffer Utilization
Figure A.73: Random Topology 4 - Network Buffer Utilization
Figure A.74: Random Topology 4 - Fairness Index
A.2.5 Random Topology 5

Figure A.75: Random Topology 5 - Pure injection rate
Figure A.76: Random Topology 5 - Per flow Injection Rate
Figure A.77: Random Topology 5 - Per Flow Delay
Figure A.78: Random Topology 5 - Spatial Reuse
Figure A.79: Random Topology 5 - Air Time Utilization
Figure A.80: Random Topology 5 - Per flow Throughput
Figure A.81: Random Topology 5 - Average Throughput
Figure A.82: Random Topology 5 - Aggregate Utility
Figure A.83: Random Topology 5 - Per Node Buffer Utilization
Figure A.84: Random Topology 5 - Network Buffer Utilization
Figure A.85: Random Topology 5 - Fairness Index
A.2.6 Random Topology 6

Figure A.86: Random Topology 6 - Pure injection rate
Figure A.87: Random Topology 6 - Per flow Injection Rate
Figure A.88: Random Topology 6 - Per Flow Delay
Figure A.89: Random Topology 6 - Spatial Reuse
Figure A.90: Random Topology 6 - Air Time Utilization
Figure A.91: Random Topology 6 - Per flow Throughput
Figure A.92: Random Topology 6 - Average Throughput
Figure A.93: Random Topology 6 - Aggregate Utility
Figure A.94: Random Topology 6 - Per Node Buffer Utilization
Figure A.95: Random Topology 6 - Network Buffer Utilization
Figure A.96: Random Topology 6 - Fairness Index
A.2.7 Random Topology 7

Figure A.97: Random Topology 7 - Pure injection rate
Figure A.98: Random Topology 7 - Per flow Injection Rate
Figure A.99: Random Topology 7 - Per Flow Delay
Figure A.100: Random Topology 7 - Spatial Reuse
Figure A.101: Random Topology 7 - Air Time Utilization
Figure A.102: Random Topology 7 - Per flow Throughput
Figure A.103: Random Topology 7 - Average Throughput
Figure A.104: Random Topology 7 - Aggregate Utility
Figure A.105: Random Topology 7 - Per Node Buffer Utilization
Figure A.106: Random Topology 7 - Network Buffer Utilization
Figure A.107: Random Topology 7 - Fairness Index
A.2.8 Random Topology 8

Figure A.108: Random Topology 8 - Pure injection rate
Figure A.109: Random Topology 8 - Per flow Injection Rate
Figure A.110: Random Topology 8 - Per Flow Delay
Figure A.111: Random Topology 8 - Spatial Reuse
Figure A.112: Random Topology 8 - Air Time Utilization
Figure A.113: Random Topology 8 - Per flow Throughput
Figure A.114: Random Topology 8 - Average Throughput
Figure A.115: Random Topology 8 - Aggregate Utility
Figure A.116: Random Topology 8 - Per Node Buffer Utilization
Figure A.117: Random Topology 8 - Network Buffer Utilization
Figure A.118: Random Topology 8 - Fairness Index
A.2.9 Random Topology 9

Figure A.119: Random Topology 9 - Pure injection rate
Figure A.120: Random Topology 9 - Per flow Injection Rate
Figure A.121: Random Topology 9 - Per Flow Delay
Figure A.122: Random Topology 9 - Spatial Reuse
Figure A.123: Random Topology 9 - Air Time Utilization
Figure A.124: Random Topology 9 - Per flow Throughput
Figure A.125: Random Topology 9 - Average Throughput
Figure A.126: Random Topology 9 - Aggregate Utility
Figure A.127: Random Topology 9 - Per Node Buffer Utilization
Figure A.128: Random Topology 9 - Network Buffer Utilization
Figure A.129: Random Topology 9 - Fairness Index
A.2.10 Random Topology 10

Figure A.130: Random Topology 10 - Pure injection rate
Figure A.131: Random Topology 10 - Per flow Injection Rate
Figure A.132: Random Topology 10 - Per Flow Delay
Figure A.133: Random Topology 10 - Spatial Reuse
Figure A.134: Random Topology 10 - Air Time Utilization
<table>
<thead>
<tr>
<th>Flow #</th>
<th>Gwid-cont</th>
<th>Gwid-dis</th>
<th>TDM-maxwt</th>
<th>TDM-rand</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure A.135: Random Topology 10 - Per flow Throughput
Figure A.136: Random Topology 10 - Average Throughput
Figure A.137: Random Topology 10 - Aggregate Utility
Figure A.138: Random Topology 10 - Per Node Buffer Utilization
Figure A.139: Random Topology 10 - Network Buffer Utilization
Figure A.140: Random Topology 10 - Fairness Index
A.2.11 Random Topology 11

Figure A.141: Random Topology 11 - Pure injection rate
Figure A.142: Random Topology 11 - Per flow Injection Rate
Figure A.143: Random Topology 11 - Per Flow Delay
Figure A.144: Random Topology 11 - Spatial Reuse
Figure A.145: Random Topology 11 - Air Time Utilization
Figure A.146: Random Topology 11 - Per flow Throughput
Figure A.147: Random Topology 11 - Average Throughput
Figure A.148: Random Topology 11 - Aggregate Utility
Figure A.149: Random Topology 11 - Per Node Buffer Utilization
Figure A.150: Random Topology 11 - Network Buffer Utilization
Figure A.151: Random Topology 11 - Fairness Index
Figure A.152: Random Topology 12 - Pure injection rate
Figure A.153: Random Topology 12 - Per flow Injection Rate
Figure A.154: Random Topology 12 - Per Flow Delay
Figure A.155: Random Topology 12 - Spatial Reuse
Figure A.156: Random Topology 12 - Air Time Utilization
Figure A.157: Random Topology 12 - Per flow Throughput
Figure A.158: Random Topology 12 - Average Throughput
Figure A.159: Random Topology 12 - Aggregate Utility
Figure A.160: Random Topology 12 - Per Node Buffer Utilization
Figure A.161: Random Topology 12 - Network Buffer Utilization
Figure A.162: Random Topology 12 - Fairness Index
A.2.13 Random Topology 13

Figure A.163: Random Topology 13 - Pure injection rate
### Figure A.164: Random Topology 13 - Per flow Injection Rate

<table>
<thead>
<tr>
<th>Flow #</th>
<th>Injection Rate (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
</tr>
</tbody>
</table>

Legend:
- Gwid-cont
- Gwid-dis
- TDM-maxwt
Figure A.165: Random Topology 13 - Per Flow Delay
Figure A.166: Random Topology 13 - Spatial Reuse
Figure A.167: Random Topology 13 - Air Time Utilization
Figure A.168: Random Topology 13 - Per flow Throughput
Figure A.169: Random Topology 13 - Average Throughput
Figure A.170: Random Topology 13 - Aggregate Utility
Figure A.171: Random Topology 13 - Per Node Buffer Utilization
Figure A.172: Random Topology 13 - Network Buffer Utilization
Figure A.173: Random Topology 13 - Fairness Index
Figure A.174: Random Topology 14 - Pure injection rate
Figure A.175: Random Topology 14 - Per flow Injection Rate
Figure A.176: Random Topology 14 - Per Flow Delay
Figure A.177: Random Topology 14 - Spatial Reuse
Figure A.178: Random Topology 14 - Air Time Utilization
Figure A.179: Random Topology 14 - Per flow Throughput
Figure A.180: Random Topology 14 - Average Throughput
Figure A.181: Random Topology 14 - Aggregate Utility
Figure A.182: Random Topology 14 - Per Node Buffer Utilization
Figure A.183: Random Topology 14 - Network Buffer Utilization
Figure A.184: Random Topology 14 - Fairness Index
A.2.15 Random Topology 15

Figure A.185: Random Topology 15 - Pure injection rate
Figure A.186: Random Topology 15 - Per flow Injection Rate
Figure A.187: Random Topology 15 - Per Flow Delay
Figure A.188: Random Topology 15 - Spatial Reuse
Figure A.189: Random Topology 15 - Air Time Utilization
Figure A.190: Random Topology 15 - Per flow Throughput
Figure A.191: Random Topology 15 - Average Throughput
Figure A.192: Random Topology 15 - Aggregate Utility
Figure A.193: Random Topology 15 - Per Node Buffer Utilization
Figure A.194: Random Topology 15 - Network Buffer Utilization
Figure A.195: Random Topology 15 - Fairness Index
A.2.16 Random Topology 16

Figure A.196: Random Topology 16 - Pure injection rate
Figure A.197: Random Topology 16 - Per flow Injection Rate
Figure A.198: Random Topology 16 - Per Flow Delay
Figure A.199: Random Topology 16 - Spatial Reuse
Figure A.200: Random Topology 16 - Air Time Utilization
Figure A.201: Random Topology 16 - Per flow Throughput
Figure A.202: Random Topology 16 - Average Throughput

Average Throughput (Mbps)
Figure A.203: Random Topology 16 - Aggregate Utility
Figure A.204: Random Topology 16 - Per Node Buffer Utilization
Figure A.205: Random Topology 16 - Network Buffer Utilization
Figure A.206: Random Topology 16 - Fairness Index
Figure A.207: Random Topology 17 - Pure injection rate
Figure A.208: Random Topology 17 - Per flow Injection Rate
Figure A.209: Random Topology 17 - Per Flow Delay
Figure A.210: Random Topology 17 - Spatial Reuse
Figure A.211: Random Topology 17 - Air Time Utilization
Figure A.212: Random Topology 17 - Per flow Throughput
Figure A.213: Random Topology 17 - Average Throughput
Figure A.214: Random Topology 17 - Aggregate Utility
Figure A.215: Random Topology 17 - Per Node Buffer Utilization
Figure A.216: Random Topology 17 - Network Buffer Utilization
Figure A.217: Random Topology 17 - Fairness Index
Figure A.218: Random Topology 18 - Pure injection rate
Figure A.219: Random Topology 18 - Per flow Injection Rate
Figure A.220: Random Topology 18 - Per Flow Delay
Figure A.221: Random Topology 18 - Spatial Reuse
Figure A.222: Random Topology 18 - Air Time Utilization
Figure A.223: Random Topology 18 - Per flow Throughput
Figure A.224: Random Topology 18 - Average Throughput
Figure A.225: Random Topology 18 - Aggregate Utility
Figure A.226: Random Topology 18 - Per Node Buffer Utilization
Figure A.227: Random Topology 18 - Network Buffer Utilization
Figure A.228: Random Topology 18 - Fairness Index
Figure A.229: Random Topology 19 - Pure injection rate
Figure A.230: Random Topology 19 - Per flow Injection Rate
Figure A.231: Random Topology 19 - Per Flow Delay
Figure A.232: Random Topology 19 - Spatial Reuse
Figure A.233: Random Topology 19 - Air Time Utilization
Figure A.234: Random Topology 19 - Per flow Throughput
Figure A.235: Random Topology 19 - Average Throughput
Figure A.236: Random Topology 19 - Aggregate Utility
Figure A.237: Random Topology 19 - Per Node Buffer Utilization
Figure A.239: Random Topology 19 - Fairness Index
Figure A.240: Random Topology 20 - Pure injection rate
Figure A.241: Random Topology 20 - Per flow Injection Rate
<table>
<thead>
<tr>
<th>Gwid-cont</th>
<th>Gwid-dis</th>
<th>TDM-maxwt</th>
<th>TDM-rand</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure A.242: Random Topology 20 - Per Flow Delay
Figure A.243: Random Topology 20 - Spatial Reuse
Figure A.244: Random Topology 20 - Air Time Utilization
Figure A.245: Random Topology 20 - Per flow Throughput
Figure A.246: Random Topology 20 - Average Throughput
Figure A.247: Random Topology 20 - Aggregate Utility
Figure A.248: Random Topology 20 - Per Node Buffer Utilization
Figure A.249: Random Topology 20 - Network Buffer Utilization
Figure A.250: Random Topology 20 - Fairness Index
A.2.21 Random Topology 21

Figure A.251: Random Topology 21 - Pure injection rate
Figure A.252: Random Topology 21 - Per flow Injection Rate
Figure A.253: Random Topology 21 - Per Flow Delay
Figure A.254: Random Topology 21 - Spatial Reuse
Figure A.255: Random Topology 21 - Air Time Utilization
Figure A.256: Random Topology 21 - Per flow Throughput
Figure A.257: Random Topology 21 - Average Throughput
Figure A.258: Random Topology 21 - Aggregate Utility
Figure A.259: Random Topology 21 - Per Node Buffer Utilization
Figure A.260: Random Topology 21 - Network Buffer Utilization
Figure A.261: Random Topology 21 - Fairness Index
Figure A.2.22: Random Topology 22 - Pure injection rate
Figure A.263: Random Topology 22 - Per flow Injection Rate
Figure A.264: Random Topology 22 - Per Flow Delay
Figure A.265: Random Topology 22 - Spatial Reuse
Figure A.266: Random Topology 22 - Air Time Utilization
Figure A.267: Random Topology 22 - Per flow Throughput
Figure A.268: Random Topology 22 - Average Throughput
Figure A.269: Random Topology 22 - Aggregate Utility
Figure A.270: Random Topology 22 - Per Node Buffer Utilization
Figure A.271: Random Topology 22 - Network Buffer Utilization
Figure A.272: Random Topology 22 - Fairness Index
Figure A.273: Random Topology 23 - Pure injection rate
Figure A.274: Random Topology 23 - Per flow Injection Rate
Figure A.275: Random Topology 23 - Per Flow Delay
Figure A.276: Random Topology 23 - Spatial Reuse
Figure A.277: Random Topology 23 - Air Time Utilization
Figure A.278: Random Topology 23 - Per flow Throughput
Figure A.279: Random Topology 23 - Average Throughput
Figure A.280: Random Topology 23 - Aggregate Utility
Figure A.281: Random Topology 23 - Per Node Buffer Utilization
Figure A.282: Random Topology 23 - Network Buffer Utilization
Figure A.283: Random Topology 23 - Fairness Index
Figure A.284: Random Topology 24 - Pure injection rate
Figure A.285: Random Topology 24 - Per Flow Injection Rate
Figure A.286: Random Topology 24 - Per Flow Delay
Figure A.287: Random Topology 24 - Spatial Reuse
Figure A.288: Random Topology 24 - Air Time Utilization
Figure A.289: Random Topology 24 - Per flow Throughput
Figure A.290: Random Topology 24 - Average Throughput
Figure A.291: Random Topology 24 - Aggregate Utility
Figure A.292: Random Topology 24 - Per Node Buffer Utilization
Figure A.293: Random Topology 24 - Network Buffer Utilization
Figure A.294: Random Topology 24 - Fairness Index
Figure A.295: Random Topology 25 - Pure injection rate
Figure A.296: Random Topology 25 - Per flow Injection Rate
Figure A.297: Random Topology 25 - Per Flow Delay
Figure A.298: Random Topology 25 - Spatial Reuse
Figure A.299: Random Topology 25 - Air Time Utilization
Figure A.300: Random Topology 25 - Per flow Throughput
Figure A.301: Random Topology 25 - Average Throughput
Figure A.302: Random Topology 25 - Aggregate Utility
Figure A.303: Random Topology 25 - Per Node Buffer Utilization
Figure A.304: Random Topology 25 - Network Buffer Utilization
Figure A.305: Random Topology 25 - Fairness Index
Figure A.306: Random Topology 26 - Pure injection rate
Figure A.307: Random Topology 26 - Per flow Injection Rate
Figure A.308: Random Topology 26 - Per Flow Delay
Figure A.309: Random Topology 26 - Spatial Reuse
Figure A.310: Random Topology 26 - Air Time Utilization
Figure A.311: Random Topology 26 - Per flow Throughput
Figure A.312: Random Topology 26 - Average Throughput
Figure A.313: Random Topology 26 - Aggregate Utility
Figure A.314: Random Topology 26 - Per Node Buffer Utilization
Figure A.315: Random Topology 26 - Network Buffer Utilization
Figure A.316: Random Topology 26 - Fairness Index
Figure A.317: Random Topology 27 - Pure injection rate
Figure A.318: Random Topology 27 - Per flow Injection Rate
Figure A.319: Random Topology 27 - Per Flow Delay
Figure A.320: Random Topology 27 - Spatial Reuse
Figure A.321: Random Topology 27 - Air Time Utilization
Figure A.322: Random Topology 27 - Per flow Throughput
Figure A.323: Random Topology 27 - Average Throughput
Figure A.324: Random Topology 27 - Aggregate Utility
Figure A.325: Random Topology 27 - Per Node Buffer Utilization
Figure A.326: Random Topology 27 - Network Buffer Utilization
Figure A.327: Random Topology 27 - Fairness Index
Figure A.328: Random Topology 27 - Pure injection rate
Figure A.329: Random Topology 27 - Per flow Injection Rate
Figure A.330: Random Topology 27 - Per Flow Delay
Figure A.331: Random Topology 27 - Spatial Reuse
Figure A.332: Random Topology 27 - Air Time Utilization
Figure A.333: Random Topology 27 - Per flow Throughput
Figure A.334: Random Topology 27 - Average Throughput
Figure A.335: Random Topology 27 - Aggregate Utility
Figure A.336: Random Topology 27 - Per Node Buffer Utilization
Figure A.337: Random Topology 27 - Network Buffer Utilization
Figure A.338: Random Topology 27 - Fairness Index
Figure A.339: Random Topology 29 - Pure injection rate
Figure A.340: Random Topology 29 - Per flow Injection Rate
Figure A.341: Random Topology 29 - Per Flow Delay
Figure A.342: Random Topology 29 - Spatial Reuse
Figure A.343: Random Topology 29 - Air Time Utilization
Figure A.344: Random Topology 29 - Per flow Throughput
Figure A.345: Random Topology 29 - Average Throughput
Figure A.346: Random Topology 29 - Aggregate Utility
Figure A.347: Random Topology 29 - Per Node Buffer Utilization
Figure A.348: Random Topology 29 - Network Buffer Utilization
Figure A.349: Random Topology 29 - Fairness Index
Figure A.350: Random Topology 30 - Pure injection rate
Figure A.351: Random Topology 30 - Per flow Injection Rate
Figure A.352: Random Topology 30 - Per Flow Delay
Figure A.353: Random Topology 30 - Spatial Reuse
Figure A.354: Random Topology 30 - Air Time Utilization
Figure A.355: Random Topology 30 - Per flow Throughput
Figure A.356: Random Topology 30 - Average Throughput
Figure A.357: Random Topology 30 - Aggregate Utility
Figure A.358: Random Topology 30 - Per Node Buffer Utilization
Figure A.359: Random Topology 30 - Network Buffer Utilization
Figure A.360: Random Topology 30 - Fairness Index