Adaptive Random Network Coding in WiMAX

Jin Jin, Baochun Li
Department of Electrical and Computer Engineering
University of Toronto
{jinjin, bli}@eecg.toronto.edu

***Abstract*—The IEEE 802.16 standard, or WiMAX, has emerged as one of the strongest contenders for broadband wireless access technology. In our previous work, we proposed a protocol using random network coding in WiMAX MAC layer, which has indeed offered important advantages. In this paper, we propose adaptive random network coding in WiMAX by introducing two algorithms in order to further improve the performance. First, we utilize the channel state information feedback to dynamically construct the packets that could be better matched to the fluctuating channel conditions, so as to obtain a higher throughput. Second, we design the algorithm by adapting the number of upstream nodes, which helps to minimize the overhead of sender push of redundant coded blocks to the receiver even after the receiver has completely received the data segment. Coupled with random network coding, both algorithms offer salient performance improvement as evidenced in our simulation evaluation.**

I. Introduction

The IEEE 802.16 family of standards, or *WiMAX* [1], has been designed to facilitate high data rate communication in metropolitan-area wireless networks. It implements both packet-oriented data transmission and standard mobile telephony, and provides better performance than traditional wireless communication standards, especially for applications requiring high and stable throughput.

Specifically at the physical layer of WiMAX, Hybrid Automatic Retransmission reQuest (HARQ) has been employed to provide reliable data transmission [2]. It is a variant of the ARQ error control protocol, combining ARQ with Forward Error Correction (FEC). However, HARQ incurs overhead with its retransmissions and ACK/NACK packets. The built-in reliability sacrifices some degree of transmission *resilience* in realistic channels with varying quality conditions over time. In addition, in handover and multi-hop transmission modes in WiMAX, a mobile station is able to establish connections with two or more uplink nodes through different sub-channels. In these cases, HARQ is not able to fully utilize the wireless bandwidth, as it is designed for a point-to-point channel. Further, as HARQ is performed on all the links, it incurs additional overhead.

On the contrary, *network coding* has been originally proposed in information theory [3], and has since emerged as one of the most promising information theoretic approaches to improve network performance. It has been shown that *random linear codes* using a Galois field of a limited size are sufficient to implement network coding in a practical network setting [4]. It has recently been shown that network coding is able to significantly improve end-to-end unicast throughput in multi-hop wireless networks, when implemented above the MAC layer of IEEE 802.11 [5], [6].

In our previous work [7], we seek to investigate the usefulness of random network coding in WiMAX. We designed a MAC-layer protocol by employing random network coding, rather than the traditional HARQ at the physical layer, with the intention of taking full advantage of its rateless properties. With such properties, random network coding offers resilient and stable transmission, due to its inherent resilience to errors. Should a particular packet be lost, subsequent packets received are equally innovative and useful. Therefore, random network coding is able to adapt the rate of data transmission to coincide with the available bandwidth in time-varying wireless channel conditions. Further, in the multi-path transmission scenarios, the mobile station is able to enjoy data transmissions from several upstream nodes concurrently with random network coding, by which the wireless medium is fully utilized. These observations may lead to the future use of random network coding at the MAC layer in practical WiMAX systems.

Though random network coding has already shown its usefulness in WiMAX, there are some features which can be tuned to further improve the performance. In this paper, we propose *adaptive random network coding* in WiMAX by introducing two algorithms at the MAC layer. First, we exploit the flexibility present at the MAC layer for construction and transmission of the MAC layer packet, which is also the basic transmission unit, referred to as *block* or *coded block*, in random network coding. We seek to change the block size adaptively by tuning the tradeoff between block error rate and protocol overhead. A feedback-based approach is employed in the algorithm by estimating the channel quality using average block error rate as the metric. The dynamic manner in which the block sizes are changed to match the channel conditions helps to improve throughput over unreliable and fluctuating wireless channel conditions.

Second, we study how to further utilize the bandwidth efficiently. In handover and multi-hop scenarios, the mobile station is able to communicate with two or more upstream nodes simultaneously. When the mobile station completely receives a data segment, it will send the feedback to all upstream nodes to stop the current transmission and invoke

This work was supported in part by Bell Canada through its Bell University Laboratories R&D program, and by a grant from LG Electronics.

the transmission of the next segment. However, the upstream nodes push redundant blocks to the receiver due to the delay of feedback transmission. We design an algorithm for the mobile station to send the feedback even before it has completely received the segment, in order to prematurely stop the transmissions of a subset of upstream nodes for this segment. By adapting the number of upstream nodes, we could reduce the amount of overhead, and therefore fully utilize scarce wireless bandwidth.

The remainder of this paper is organized as follows. In Sec. II, we briefly review the MAC-layer random network coding protocol in WiMAX. In Sec. III, we present the intuition and design of the *adaptive block size* algorithm. In Sec. IV, we introduce the algorithm on adapting the number of upstream nodes in multi-path transmission scenarios. Our performance evaluation is presented in Sec. V. Finally, we conclude the paper in Sec. VI.

II. MAC-LAYER RANDOM NETWORK CODING

In our previous work [7], we proposed a detailed protocol to use random network coding in WiMAX MAC layer, by replacing the traditional HARQ at the physical layer. Such a MAC-layer Random Network Coding protocol, henceforth referred to as MRNC for brevity, is designed to improve the throughput performance in WiMAX.

In MRNC, when a data segment (also referred to as a *generation* or a *group* in the literature) is to be transmitted, it is divided into n blocks, denoted as $[b_1, b_2, \cdots, b_n]$, each of which has a fixed number of bytes. The sender randomly chooses a set of coding coefficients $[c_1, c_2, \cdots, c_n]$ in the Galois field GF(2^8), and produces one coded block x as a linear combination of the original data blocks:

$$x = \sum_{i=1}^{n} c_i b_i \tag{1}$$

In MRNC, the sender keeps on transmitting the coded blocks to the receiver. At the other side, the receiver only needs to "hold a bucket" to "collect" the coded blocks. For each block the receiver collects, it progressively decodes the segment using Gauss-Jordan elimination. Immediately after n linearly independent blocks have been received for a segment, the receiver is able to recover the original data segment, and sends the ACK back to the sender. This feedback will request the sender to stop the transmission of the current segment and switch to the next segment. During this process of transmission, it is not necessary to transmit ACK/NACK and retransmission packets with *each individual* packet, as in HARQ, which generate much overhead. Not surprisingly, the simulation results showed that MRNC is helpful to improve the average throughput, and it offers more stable and resilient throughput over time, due to the inherent resilience to errors with random network coding.

Random network coding also helps to fully utilize the scarce wireless bandwidth in handover and multi-hop scenarios in WiMAX. In the handover region of both cases, the mobile station (MS) is able to maintain the connections with two or more upstream nodes through separate downlink sub-channels supported by OFDM/OFDMA in WiMAX physical layer. With the advantages of random network coding, each downlink connection can be used separately for transmitting different coded blocks simultaneously without collision. MS collects the data concurrently, such that it could decode the segment after receiving a sufficient number of coded blocks. In this manner, random network coding helps to fully utilize the downlink channels, and avoid the overhead of retransmission and feedback packets generated by HARQ in each connection and each hop. As we expected, the simulation results revealed that MRNC achieves significant performance improvement over HARQ in both handover and multi-hop scenarios. This coincides with our intuition that network coding fits naturally in the multi-path transmissions in WiMAX.

III. ADAPTIVE BLOCK SIZE

In MRNC, the packet size plays a crucial role in WiMAX throughput performance. At the MAC layer, the packet is considered as a MAC layer Protocol Data Unit, which is also the basic transmission unit, referred to as *block* or *coded block*, in MRNC. We can simply represent the throughput under MRNC as $R(1-P_e)$ [7], where R is the channel rate, and P_e denotes the block error rate, since blocks are often corrupted during transmission in error prone wireless channels. It is noted that under the same bit error rate, a decreasing block size would decrease the block error rate as well. Similarly, it can be argued that if the block size increases, the resulting block error rate increases as well.

Based on the intuition above, we can see that a smaller block is helpful to achieve a lower block error rate. Of course, the flip side of the coin is the lower transmission efficiency due to a lower payload to protocol overhead ratio. On the other hand, a larger block size achieves better efficiency, but leads to a higher block error rate. Thus, we observe that both large and small block sizes have their advantages and disadvantages in MRNC. The natural question that arises is: *how do we adjust the block size to obtain better performance?*

We propose an algorithm, referred to as *adaptive block size*, to dynamically construct and transmit the blocks in MRNC. We focus on the *goodput*, rather than throughput, to obtain a better measurement on the transmission performance in the WiMAX MAC layer. Goodput is the application level throughput, *i.e.*, the number of useful bits transmitted per unit of time, excluding protocol overhead.

Intuitively, we are able to adaptively tune the block size in response to the channel conditions in order to improve goodput. At the MAC layer, WiMAX is capable of performing the *Aggregation* and *Fragmentation* of MAC layer data units [1], with which we could vary the block size in MRNC. Heuristically, when the channel quality becomes high, we increase the block size. The larger the block size is, the less overhead of MAC header achieves while maintaining that very few redundant coded blocks are sent. On the other hand, we could manage to raise the goodput by decreasing the block size under poor channel conditions. The smaller block size could

Feedback type	Block error rate
1	$< 5\%$
2	$< 10\%$ & $> 5\%$
3	$< 20\%$ & $> 10\%$
4	$< 40\%$ & $> 20\%$
5	$< 90\%$ & $> 40\%$
6	$> 90\%$

TABLE I
DIFFERENT FEEDBACK TYPES.

Feedback type	Block size change
1	+75 bytes
2	+50 bytes
3	-25 bytes
4	-50 bytes
5	-75 bytes
6	-100 bytes

TABLE II
ADAPTIVELY CHANGING BLOCK SIZE BASED ON THE FEEDBACK.

help to obtain a lower block error rate, thus fewer redundant coded blocks are required to transmit.

We design a feedback-based scheme to achieve such block size adaption. In MRNC, after the receiver completely receives the entire segment, it will send feedback to the sender to stop the current transmission, and invoke the transmission of the next segment. With this feedback, the receiver could explicitly report the channel state information based on the transmission of the coded blocks of the entire segment. We use the average block error rate achieved in the transmission as the metric. With such knowledge, the sender is able to dynamically adjust the block size and construct coded blocks for the transmission of the subsequent segment.

A heuristic approach is adopted for this adaptive algorithm in our design. First, we establish a finely tuned feedback granularity to represent different channel states which indicate different levels of average block error rates. We propose six types of feedback, shown in Table I, each of which identifies one of the states of the transmission quality.

Upon receiving feedback from the receiver, the sender could tune the block size according to the type of the feedback. The changes in block size accordingly are shown in Table II, where for instance, "+50" implies the algorithm will increase the block size by 50 bytes, and "-25" implies to decrease the size by 25 bytes. With this *adaptive block size* algorithm, we are able to dynamically change the block size to adapt to the fluctuating channel conditions, which will lead to higher goodput. We will verify the usefulness of this algorithm in Sec. V.

IV. ADAPTIVE UPSTREAM NODE

In WiMAX handover and multi-hop modes, the mobile station (MS) could build connections with two or more upstream nodes through different sub-channels to enjoy the multi-path transmission. In MRNC, random network coding is helpful to make full use of the available bandwidth from each upstream nodes, and thus to improve the downlink throughput. During the transmission, upon receiving a sufficient number of coded blocks, MS could decode the original segment immediately, and send feedback to all upstream nodes to stop their current transmission, and ask them to proceed to the next segment.

The problem in the current protocol is that the delay of feedback transmission will cause more redundant coded blocks to be pushed to the receiver. We define the *feedback delay* as the time difference between the time the last bit of the last coded block (after receiving this block, the entire segment could be decoded correctly by the receiver) is transmitted and the time a feedback packet is received by the senders.

Now we use a simple illustrative example to show how the *feedback delay* affects the performance. Assume that at time t, the last bit of the final innovative coded block is received. The segment could therefore be decoded at time t and all upstream nodes receive the feedback at time $t + D$, with D equal to *feedback delay*. Without explicit knowledge, during the time period from t to $t + D$, all upstream nodes continue their transmission of the coded blocks of the segment that has already been correctly received by MS. In the best case, the transmission of the next block is in progress while the feedback arrives. Upon receiving it, the upstream nodes could only switch to the transmission of the next segment after finishing the redundant transmission of the current block. We can do a back-of-envelope calculation. If the block size is 256 bytes, and there are 4 upstream nodes communicating with the MS concurrently. Thus, totally 256·4 = 1K bytes in total will be dropped. If we further consider a large scale WiMAX network with high speed data transmission and with a large number of mobile nodes being served concurrently, it becomes obvious that the scarce wireless bandwidth is underutilized due to the deficiency of the protocol. Moreover, the situation will be even worse when a larger *feedback delay* is unavoidable. As such, mitigating the effect of *feedback delay* and fully utilizing the wireless bandwidth should be critical design objectives.

To solve the above problem, we propose an algorithm by adapting the number of upstream nodes, referred to as *adaptive upstream node*, to effectively utilize the bandwidth. This algorithm enables the MS to send feedback before it has completely received all the coded blocks for decoding the segment. Therefore, the upstream nodes is able to prematurely stop when segment downloading is almost completed. This *adaptive upstream node* algorithm is designed to favor upstream nodes with better bandwidth to complete the download, and stop the upstream nodes with lower data rates. It will gradually stop more nodes based on the completion timing estimates in the transmission process, in order to avoid the transmission of redundant packets.

We design a heuristic approach for such an algorithm. When 3/4 coded blocks from a segment are successfully received, the MS could stop the upstream nodes with the lowest throughput. The number of upstream nodes to be prevented is one half of the nodes associated to the MS. Upon receiving the feedback, these nodes will stop the transmission of the current segment, but start to transmit the coded blocks of the next segment. If the entire data transmission is completed, the connections will be released, and the bandwidth may be reallocated. Meanwhile, the remaining upstream nodes with higher throughput shall keep transmitting the coded blocks from the current

Conditions	Actions
received 3/4 coded blocks	stop 1/2 of upstream nodes with the lowest throughput, and ask them to transmit blocks from the next segment
received 7/8 coded blocks	stop another 1/4 of upstream nodes with the lowest throughput, and ask them to transmit blocks from the next segment
received 7/8 coded blocks	the remaining 1/4 of upstream nodes take turns to transmit blocks from two segments
received all coded blocks	all upstream nodes transmit the blocks of the next segment

TABLE III
ADAPTIVE UPSTREAM NODE ALGORITHM.

segment. After finishing the transmission for 7/8 coded blocks, the MS will only maintain 1/4 of the connections with the highest throughput.

We wish to tune the design of the algorithm to further improve bandwidth utilization. At the same time that the MS sends feedback to a certain set of upstream nodes when 7/8 coded blocks are received, the feedback is also sent to the remaining 1/4 upstream nodes to inform them that the transmission for the current segment is almost done. Upon receiving this signal, they take turns to transmit the coded blocks from the current segment and from the next segments. Consider the scenario when the last coded block from the current segment is transmitted, the following data block to be sent should be the block from the next segment. Normally when feedback arrives after *feedback delay*, the transmission of blocks from the next segment has already been started, which is useful. Thus, the wasted bandwidth is reduced. The entire heuristic algorithm is summarized in Table III.

V. PERFORMANCE EVALUATION

We are now ready to show extensive simulations to study the performance of our adaptive algorithms. For this purpose, we use Matlab and ns-2 WiMAX simulator which is the only simulator for WiMAX that is available to be used in both academia and industry. The parameters used in the simulation are configured based on the WiMAX technical specification document [1].

A. Evaluating the Adaptive Block Size Algorithm

We first focus on the goodput performance in single-hop transmissions, and seek to examine the usefulness of the adaptive block size algorithm in this scenario. For a fair comparison, we try to find out the optimized block size for MRNC without the adaptive scheme (referred to as regular MRNC) through an extensive simulation.

The simulation is carried out under the following scenarios. One mobile station (MS) moves around the service area of a cell randomly. Its initial speeds (in km/h) and directions (in degrees) are generated with a uniform distribution of $U[10, 50]$ and $U[0, 360]$, respectively. The MS will change its speed and direction after a certain amount of time with an exponential distribution, with a mean value of 10 seconds. The new speed is uniformly generated with $U[10, 50]$ if the current speed is beyond 10 km/h; otherwise, it is obtained using $U[v-10, v+10]$, where v is the current speed. The new direction is obtained from a Gaussian distribution with the mean as the current direction, and a standard deviation of 40 (degrees). The initial location of the MS is randomly chosen in the service region. The design of this simulation scenario aims to provide realistic time-varying channel conditions in WiMAX.

We could obtain an average signal-to-noise ratio (SNR) through a 1000-second simulation. We then perform the simulation 100 times independently to obtain 100 average SNRs. Further, the average value of this 100 average SNRs could be obtained, and be used to fairly estimate the average channel quality. Under such an average value, we could obtain goodput performance on deploying regular MRNC with different block sizes. Based on the simulation results, we have obtained the highest goodput when using a block size of 128 bytes. We use it in the following performance evaluation of both regular MRNC and HARQ.

Now, we are ready to examine the benefits of the adaptive block size algorithm by comparing three protocols: MRNC with the adaptive block size algorithm (referred to as adaptive MRNC), regular MRNC and HARQ. All are used to transfer a large file in the downlink over 1000 seconds between the same base station (BS) and MS pair. In the simulation, MS moves around the service area using the same way as the previous simulation.

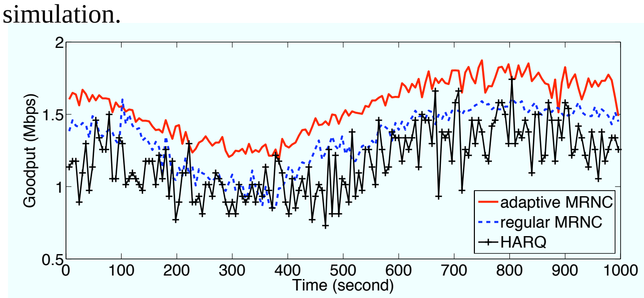


Fig. 1. Goodput performance in single-hop transmissions.

Fig. 1 shows the downlink goodput performance of four protocols in the simulation. We observe that the adaptive MRNC could help to achieve more than 20% goodput improvement over the regular MRNC, and further over 30% gain compared with HARQ. Meanwhile, it has a smaller variance in goodput over time compared with the regular MRNC, and over HARQ as well. The smaller goodput variance indicates that the *adaptive block size* algorithm could help to achieve more resilient and stable transmission. The results are shown in Table IV in details. This observation coincides with our intuition and is not a surprise: it shows its ability to adapt to time-varying channels and help in maintaining a higher goodput.

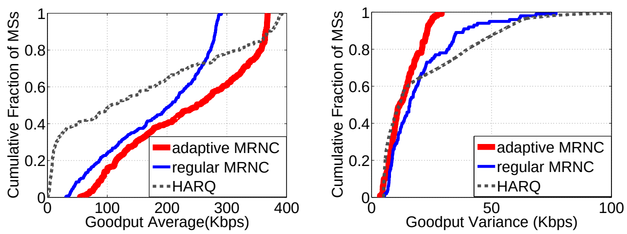
	Goodput Average	Goodput Variance
adaptive MRNC	1.55 Mbps	0.15 Mbps
regular MRNC	1.29 Mbps	0.18 Mbps
HARQ	1.18 Mbps	0.22 Mbps

TABLE IV
GOODPUT PERFORMANCE RESULTS IN A SINGLE-HOP TRANSMISSION.

With the objective of becoming even more realistic, we seek to extend our evaluation to a large-scale scenario. The simulation is performed with the same parameter setting, but with a large number of MSs active in the service region concurrently.

The arrival process of new MS connections is assumed to be a Poisson process with a mean of 5 connections/cell/second. The MS active time duration is exponentially distributed with a mean of 100 seconds. Every active MS moves around the service area using the same way as the previous simulations. We also run the simulation for 1000 seconds, and the downlink goodput at the MS side is examined for all three protocols. From the results, there are a total of 4990 MSs that have ever been active in the service area, with 455 MSs active simultaneously on average.

Fig. 2 plots the CDF of the average goodput and its variance. Not surprisingly, adaptive MRNC outperforms regular MRNC and HARQ by 28.4% and 57.7% respectively with respect to average goodput, due to its effective adaptation to time-varying channels. The results also show that adaptive MRNC has a smaller variance in goodput over time, by achieving 20% gain over the regular MRNC, and 55% gain over HARQ.



(a) (b)

Fig. 2. (a) CDF of goodput average. (b) CDF of goodput variance.

B. Evaluating the Adaptive Upstream Node Algorithm

We next try to identify the performance gain offered by the adaptive upstream node algorithm in the handover scenario. We first perform a simple simulation that provides hints on the benefits of the algorithm. A total of 4 BSs are deployed in the service area, and the distance between BSs is 3000 meters. The layout of the cell sites are shown in Fig. 3, and the simulation is run with the same setting as previous ones. The maximum bandwidth allocated to the MS is 500 Kbps. Fig. 4 shows that MRNC with the adaptive upstream node algorithm preserves a substantial amount of bandwidth as compared to the regular MRNC, due to its effective utilization of the wireless bandwidth. In particular, the preserved bandwidth could reach 22.4 Kbps on average.

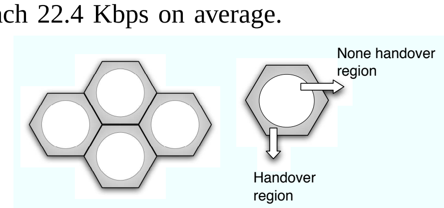


Fig. 3. The simulation scenario of WiMAX handover.

In the next experiment, we intend to extend the evaluation in a more practical setting. The simulation is performed in the cellular system above, but with the same setting as the large-scale simulation in the single-hop scenario. The simulation is run for 1000 seconds, and there have been a total of 20244 MSs ever active, with 490 MSs active during the same time epoch in each cell on average. The simulation result shows that the average preserved bandwidth on the adaptive upstream node algorithm could reach 16.3 Kbps per MS on average. The result coincides with our intuition that adaptive upstream node algorithm could effectively assist to fully utilize the scarce wireless bandwidth.

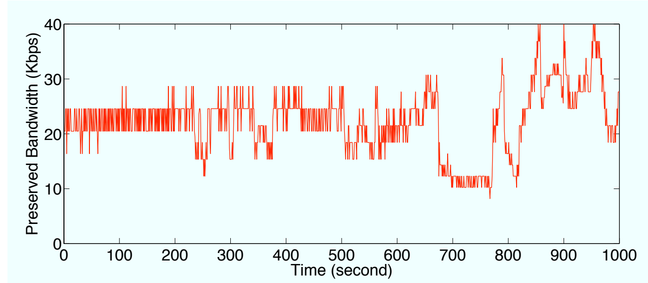


Fig. 4. Preserved bandwidth by MRNC: the adaptive upstream node algorithm over the regular MRNC in the WiMAX handover scenario.

VI. Concluding Remarks

It has been studied that random network coding is helpful in the WiMAX MAC layer. The proposed MAC-layer random network coding protocol, which took full advantage of random linear codes, has already demonstrated its ability to achieve a higher downlink throughput and better transmission resilience. In this paper, we have designed an adaptive random network coding by introducing two new heuristics: *adaptive block size* and *adaptive upstream node*. The *adaptive block size* algorithm aims to provide a more flexible scheme for data transmission by dynamically adjusting the block size in response to the channel conditions. Meanwhile, the *adaptive upstream node* algorithm could help to efficiently utilize the scarce wireless bandwidth, particularly in multi-path transmission scenarios. In our performance evaluation, we have observed that both algorithms could substantially improve the performance, with respect to both average goodput and resilience. Our studies have supported the argument of using random network coding in the WiMAX MAC layer.

References

[1] C. Eklund, R. B. Marks, K. L. Stanwood, and S. Wang, "IEEE Standard 802.16: A Technical Overview of The WirelessMAN^{TM} Air Interface for Broadband Wireless Access," *IEEE Communications Magzine*, vol. 40, no. 6, pp. 98 – 107, 2002.

[2] D. J. Costello, J. Hagenauer, H. Imai, and S. B. Wicker, "Application of Error-Control Coding," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 2531–2560, 1998.

[3] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[4] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The Benefits of Coding Over Routing in a Randomized Setting," in *Proc. of International Symposium on Information Theory (ISIT)*, 2003.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in The Air: Practical Wireless Network Coding," in *Proc. of ACM SIGCOMM*, 2006.

[6] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing," in *Proc. of ACM SIGCOMM*, 2007.

[7] J. Jin, B. Li, and T. Kong, "Is Random Network Coding Helpful in WiMAX?" in *Proc. of IEEE INFOCOM*, 2008.