Author Mousie Fasil, Alexander Kuehne, and Anja Klein, ”Node Virtualization and Network Coding: Optimizing Data Rate in Wireless Multicast,” in *Proc. The Eleventh International Symposium on Wireless Communication Systems*, August 2014.

Node Virtualization and Network Coding: Optimizing Data Rate in Wireless Multicast

Mousie Fasil, Alexander Kuehne, Anja Klein
Communications Engineering Lab, Technische Universität Darmstadt
Merckstrasse 25, 64283 Darmstadt, Germany Email:{m.fasil, a.kuehne, a.klein}@nt.tu-darmstadt.de

***Abstract*—This paper investigates the achievable sum rate for a wireless multihop network (WMN) with unequal link capacities. We focus on a multi-source multicast scenario, where the nodes are operating in half-duplex mode. For this scenario, we propose a framework which fully utilizes the broadcast (BC) gain of the wireless medium through extended virtualization. Further, we show that our framework can switch between the routing mechanisms plain routing and network coding at the network layer (NET) and can also switch between the communication types unicast (UC), multicast (MC) and BC communications at the physical layer (PHY). We show that our framework outperforms isolated layer solutions and also currently available cross-layer approaches in the literature.**

I. Introduction

Obtaining the optimal multicast rate in wireless multihop networks (WMNs) is a challenging task. The goal is to obtain the maximum achievable rate between one source and an associated group of destinations, which are interested in the same information (e.g. audio or video). In a single source multicast scenario, where only one source is routing to a group of destinations, the problem can be expressed as a linear optimization problem [1]. The solution for a multi-source multicast scenario, where multiple sources route to their respective group of destinations, can be obtained by determining a multicast tree for each source, which is NP-hard [2].

In addition, the wireless communication introduces interference and collisions, where a collision occurs if a node tries to transmit and receive at the same time. Therefore, scheduling needs to be performed in addition to routing. Another aspect in WMNs is the utilization of the broadcast (BC) gain of the wireless medium, especially in combination with routing. The aim is to make use of the omnidirectional communication such that the network benefits by either higher achievable throughput or by minimized energy consumption.

In this paper, we focus on multicast rate maximization in WMNs for a multi-source multicast scenario. We are specifically looking into WMNs with unequal link capacities. Thus, our problem consists of three domains, namely the network layer (NET), the medium access layer (MAC) and the physical layer (PHY).

The NET problem is to obtain a solution for the routing. One approach is to use network coding [3], which allows the formulation of a linear optimization problem as in [4]. Network coding was introduced by Ahlswede et al. in [3]. In their pioneering paper, they showed that assuming information as flows is not achieving optimality in a multicast scenario. Instead of replicating and forwarding the information, the information should be combined at intermediate nodes. There are several works on wired [5], wireless [1], [4], [6], and hybrid networks [2] with regards to network coding. Nevertheless, network coding does not always provide a gain over plain routing as seen in [4]. Hence, Li et al. [4] present an approach which utilizes plain routing and network coding, but do not apply it for WMNs.

For the scheduling at the MAC, Sagduyu and Ephremides proposed a heuristic to perform conflict-free scheduling. The heuristic separates the network so that nodes do not interfere with each other and do not transmit and receive at the same time. The heuristic is used in combination with network coding and plain routing separately. Although the approach is used in a wireless network, they assume equal link capacities. In [6], the approach from Sagduyu and Ephremides is extended with virtualization. In the context of [6] and [2], virtualization utilizes the BC gain of the wireless medium by adding virtual nodes to the network. The virtual nodes are connected to multiple physical nodes over virtual links. The virtualization represents a PHY BC at the NET. Although Niati et al. address all three layers, they also assume fixed link capacities, which does not hold in a WMN.

Therefore, in this paper we present a framework which can switch between the routing mechanisms plain routing and network coding at the NET and which can also switch between the communication types unicast (UC), multicast (MC) and BC communications at the PHY. Note that this is not provided by current approaches in the literature, especially not for WMNs with unequal link capacities. Further, we evaluate the behavior of current approaches and our framework in the presence of unequal link capacities, which are represented by bottlenecks.

In this paper, we extend the virtualization approach to MC communications. We propose a new scheduler, which better utilizes the different communication types at the PHY compared to the scheduler in [1]. We combine the two optimization problems from [4] and [6], and integrate both the extended virtualization and the new scheduler into the combined optimization problem. This results in a new framework, which can cope with unequal link capacities.

We show that our framework outperforms isolated schemes in terms of throughput, where isolated is with regards to a single layer solution, e.g. network coding. Further, we reveal that available schemes cannot cope with unequal link capacities. In

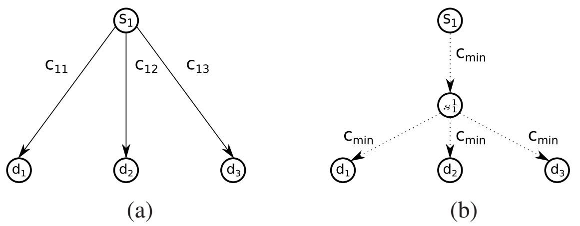

Figure 1: Virtualization utilizing the BC nature of the PHY where (a) is a four node graph G and (b) is the extended graph G^{ex}

our work, we present the advantage of performing transitions between communication types and routing mechanisms. The remainder of the paper is organized as follows. In Section II, we present the system model introducing NET, MAC and PHY. Our newly proposed scheduler and the combined optimization problem are discussed in Section III. We show simulation results in Section IV, where we evaluate the throughput in the presence of strong bottlenecks. The paper is concluded in Section V.

II. System Model

In this section, we describe the system model. First, we discuss the modeling of the WMN from the NET view, followed by an introduction of the conflict-free scheduling applied on the MAC. Next, we explain the wireless channel model from the PHY view and discuss bottlenecks and their impact on the network. Throughout the paper, we assume that all nodes in the WMN operate in half-duplex mode. Further, we assume that each node is equipped with a single omnidirectional antenna.

A. Network Layer

We model the WMN as a directed acyclic graph $G = (V, E)$. The graph G contains a set V of vertices representing nodes in the network and a set $E \subset V \times V$ of edges representing links between nodes. The set of vertices contains three subsets, the subset of sources $S \subset V$, the subset of destinations $D \subset V$ and the subset of relays $R \subset V$. A source introduces a message into the network, which is then routed through the network to the destinations. A destination can receive the message either via a direct path or via a multihop path. We define the direct path as the direct communication from a source to a destination. A multihop path includes transmissions from a source to a destination over one or multiple relays. We denote a link between two nodes as a directed edge $e = (i, j)$, where i is the transmitting node and j is the receiving node.

At the NET, we consider plain routing and network coding to route messages from the source to the group of destinations. In a WMN, network coding can reduce the number of transmissions compared to plain routing. Nevertheless, this only holds if the network topology contains multiple paths from a source to a destination. The reason is that without a distinct path for an uncoded message and at least another distinct path for the coded message, it cannot be ensured that all messages can be decoded at the destination and hence network coding becomes infeasible. In the case that network coding is not feasible, plain routing needs to be used, where messages are replicated and forwarded. The drawback is that the number of transmissions increases, which reduces the achievable throughput.

B. Medium Access Layer

In a WMN, it is necessary to separate the communications between nodes in order to avoid interference and collisions. Therefore scheduling is needed, which is done at the MAC. The scheduler determines the group of receiving nodes, by activating either an UC, a MC or a BC transmission, respectively. Node communications are separated in time in order to avoid interference and collisions of multiple packet receptions. The scheduling is link based, thus only links which do not interfere or conflict with each other are active at the same time. The scheduling decisions are represented as sub-graphs of the complete network graph G. Hence, routing is not done over G, but over each sub-graph. This results in an overall routing solution for the graph G.

Since radio transmissions in WMNs are omnidirectional, multiple nodes can receive the same message simultaneously. Thus a node can choose between different communication types depending on which one maximizes the throughput. The differentiation between UC and BC can be done by applying virtualization. Virtualization extends a given network graph by adding a virtual node and virtual links. For each node with at least two outgoing links, a virtual node is added to the network graph. The virtual node has only one incoming edge from the original node and multiple outgoing virtual links to the original receiving nodes. The link capacity of the outgoing virtual links is set to the minimum of the original outgoing links of the node. This results in an extended network graph G^{ex}, which contains the original nodes and links from G, where the links represent the UC communication links, and the virtual nodes and links which represent the BC communication.

The concept of virtualization is illustrated in Figure 1, where network graph G of Figure 1(a) is extended to network graph G^{ex} in Figure 1(b). Figure 1(a) contains only the UC links from s_1 to d_1, to d_2 to d_3, while Figure 1(b) contains only the BC links and also the virtual node s_1^1. In our example, we have three edges with link capacity c_{11}, c_{12} and c_{13}, respectively. The resulting BC link capacity is $c_{\text{min}} = \min\{c_{11}, c_{12}, c_{13}\}$.

In the presence of a weak link, BC communication is disadvantageous, since the BC links are adjusted to the weakest link. An option is to switch to UC communications, but here the number of transmissions increases, which results also in a low throughput.

Instead, MC communications needs to be introduced, allowing to put strong links and weak links in separate groups. Therefore, we propose an extended virtualization, where nodes with more than two outgoing links are virtualized multiple times. This extended virtualization works as follows: First, determine the BC communication with the first virtualization. Next, add a new virtual node connecting a group of the nodes. Repeat,

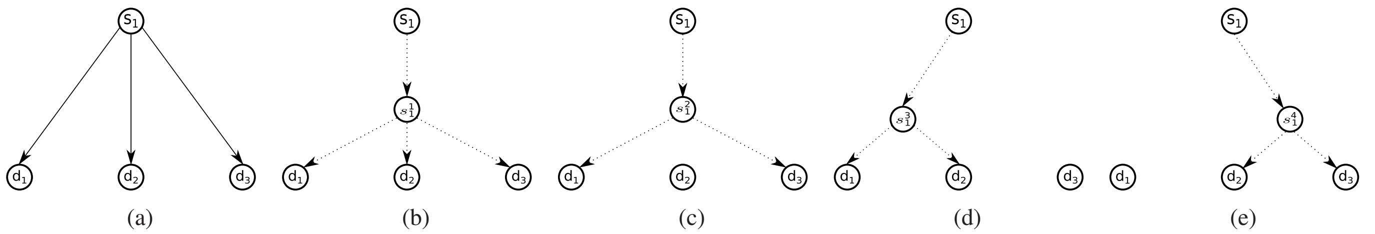

Figure 2: Virtualization of a four node network (a) without virtualization (b) BC virtualization (c) MC virtualization 1 (d) MC virtualization 2 (e) MC virtualization 3

until all groups are found. The extended virtualization fully utilizes the BC nature, since we can now distinguish between UC, BC and MC transmissions at the MAC. In conclusion, all three communication types of the PHY are available at the NET through the MAC.

The extended virtualization is shown in Figure 2, where we have node s_1 with three outgoing links, leading to four possible virtual extensions as seen in Figure 2(b)-2(e). The virtual nodes s_1^1 in 2(b) is the BC communication while the virtual nodes s_1^2 to s_1^4 represent the three possible MC transmissions in Figure 2(c)-2(e). The extended Graph G^{ex} contains all five sub-graphs. The task of the scheduler is to determine sub-graphs from G^{ex}, this cannot be done with the scheduler in [1]. Therefore, a new scheduler is needed, which we will discuss later in Section III.

C. Physical Layer

In this work, we model the PHY based on the unit disk model, which is similar to the model used in [6]. We assume omnidirectional transmissions between nodes. Two nodes are connected if the intended receiver lies within the transmission range of the sender. If an unintended receiver is in the range of the sender, then interference occurs. We assume that a node cannot transmit and receive at the same time. Further, a node is not able to receive multiple messages at the same time, since only one antenna is available at each node. In [6], the link capacity is unity and the transmission range is fixed. In our work, we assume unequal link capacity. We take into account a channel factor which depends on the distance.

In our paper, we assume that nodes can receive a message as long as they are inside a certain range from the transmitter. Let us define an inner transmission range R_i^{in} of node i and an outer transmission range R_i^{out} of node i. We define the channel factor between two nodes as

$$\gamma_{ij} = \max\left\{\frac{R_i^{\text{out}} - d_{ij}}{R_i^{\text{out}} - R_i^{\text{in}}}, 0\right\} \quad (1)$$

where d_{ij} is the distance between node i and j and is larger than or equal to the minimum distance d_0. In this paper, we set the minimum distance between nodes to $d_0 = R_i^{\text{in}}$, thus the channel factor is bounded by $0 \leq \gamma_{ij} \leq 1$. The channel factor has an effect on the link capacity where the link capacity decreases if a node moves closer to R_i^{out}, at which point the resulting link capacity will be zero due the large distance. The resulting unequal link capacities have a direct effect on the network performance, since one or multiple bottlenecks can occur.

A bottleneck occurs when one or more links have a low link capacity along a route. In this paper, we assume different channel factors between nodes. This is not considered in [6], where only fixed and equal link capacities are assumed. The impact of a bottleneck depends on its location. If a bottleneck is located within the first hop of one source, then only this source has a poor performance but the remaining network is unaffected. If a bottleneck occurs at a central part of a network, which is connected to several destinations, then the complete network is affected, leading to an overall poor performance. In this case, the multicast rate of each source decreases. This can be prevented if certain paths between a source and its destinations are avoided. Thus, destinations where only weak paths are available, have to be served with a lower rate or not at all. In general, analyzing WMNs with bottlenecks is an important step to evaluate the ability of our framework to switch between BC, UC and MC and between plain routing and network coding at the respective layers.

III. Problem Formulation and proposed Framework

In the previous section, we discussed the system model, we pointed out that the PHY communication types can be represented at the MAC and the NET. In this section, we present the new scheduler and the combined optimization problem from [4] and [6] for the routing problem.

A. Scheduling Problem

We have presented the extended virtualization, which is important to cope with bottlenecks. In order to use the extended virtualization in terms of conflict-free scheduling, a new scheduler is required. Conflict-free scheduling is ensured if no transmitter is interfering with another transmitter and if no transmitter is active as a receiver simultaneously. In addition, for the extended virtualization we require that the scheduler ensures that there is no conflict when a transmitter is switching between the different communication types. The proposed schedulers in the literature do not fulfill this requirement and hence a new scheduler is needed.

Our new scheduler is inspired by the scheduler from Sagduyu and Ephremides in [1]. To clarify the differences, we shortly present the scheduler from [1]. The scheduler constructs sub-graphs by starting with a random node and setting it as a receiver. As transmitter, the node with the smallest cost metric, e.g. transmit power [1], is chosen. The next step is to again arbitrarily choose a receive node, which has not been chosen as a transmitter or receiver. The receiver chooses a transmitter with the smallest cost metric. The new pair is added to the sub-graph as long as a) the transmitter has not already been chosen as a receiver and b) the new pair does not interfere with pairs already in the sub-graph. This is done until no new pair can be added into the sub-graph. The scheduler produces additional new sub-graphs until all nodes have been chosen as transmitter and receiver at least once. The scheduler ensures that all nodes are scheduled at least once, but it does not necessarily activate all BC communications.

Our scheduler differs in two points. The first point is that we determine the sub-graphs starting from the transmitter. The second point is that we ensure that every possible BC communication is activated. The reason is when all BC communications can be scheduled conflict-free, then we can also activate UC or MC communications without any conflict. The advantage is that, our new scheduler can adapt to changing link capacities by switching between the three communication types when needed. The scheduler has two parts. In the first part, the scheduler determines the conflict-free sub-graphs also called network realizations. The scheduler determines m network realizations $N = \{N_1, \dots N_m\}$. Each realization contains only BC communications. In detail our scheduler goes through the following steps in the first part:

1) Randomly chose a node as a transmitter.
2) Activate all nodes connected with the transmitter as receivers.
3) Add another transmitter under the following conditions:
   a) The transmitter is not active as a receiver.
   b) The transmitter does not have the same receivers as any active transmitter.
4) If no transmitter can be added, store the sub-graph N_m and create a new sub-graph and start again.
5) Repeat above steps, until all nodes were activated at least once as transmitter and receiver except for source and destination nodes.

After the first part, our scheduler has determined In the first phase we have m network realizations $N = \{N_1, \dots N_m\}$, which are only the BC communications. The second part of the scheduler determines the realizations for UC and MC transmissions. We take each N_m and create additional network realizations. For the UC scheduling a network realizations contains transmitting nodes with only one active UC link. For the MC scheduling a network realizations have the form as seen in 2 and again MC transmissions are only possible for nodes with more than two outgoing links, hence we extend the virtualization as already seen in Figure 2 to add multiple virtual nodes. This allows the algorithm to react to bottlenecks, by fully enabling the BC gain. Finally, the set of network realizations contains not only the m BC transmission but the overall p network realizations with all three communication types. We express the extended network realization as $N_p^{\text{ex}} = (V_p^{\text{ex}}, E_p^{\text{ex}})$, where p is number of network realizations.

B. Routing Problem

Next, we elaborate how to solve the routing problem. We are interested in maximizing the sum rate in the WMN. In the considered WMN, multiple sources want to route message to multiple destinations. Hence, each source introduces independent data into the network. In order to utilize network coding, a relay receiving multiple independent messages will combine these messages and forward them. In WMNs with unequal link capacities, network coding is not always the optimal mechanism, since network coding needs independent paths, where the uncoded and the coded message can be routed independently. Hence, in case that independent paths are not available, network coding is not available and plain routing must be used. Thus, we need an optimization problem which can switch between the two routing mechanisms. Therefore, we combine the optimization problem from [4] and [6]. We choose the utility function as

$$\sum_{s \in S} \log(1 + r_s) \tag{2}$$

which is adopted from [2]. We maximize the sum rate which is constrained by the maximum message flow in the network. We define the flow from a source to destination over the link from node i to node j in the p-th network realizations as $f_{i,j}^{(p)}(l, d)$. The flow is upper bounded by the link capacity. The link capacity depends on the channel factor γ_{ij} and how often the link is active. This can be determined by the number of network realizations the link is included and how often the network realization is activated. For the latter, we define the timeshare τ_p, which is associated with N_p^{ex}. A link is included in N_p^{ex} if the indicator function $\mathbf{I}_{E_p^{\text{ex}}}((i, j))$ is one, else it is zero. Therefore, the indicator function is written as

$$\mathbf{I}_{E_p^{\text{ex}}}((i, j)) = \begin{cases} 1, & \text{if}(i, j) \in E_p^{\text{ex}} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Thus, the link capacity $c_{i,j}$ between node i and j depends on γ_{ij}, τ_p and $\mathbf{I}_{E_p^{\text{ex}}}((i, j))$ and is defined as follows

$$c_{i,j} = \sum_{p=1}^{P} \tau_p \cdot \gamma_{ij} \cdot \mathbf{I}_{E_p^{\text{ex}}}((i, j)). \tag{4}$$

Therefore, the sum of flows $f_{i,j}^{(p)}(l, d)$ through a link are bounded by

$$0 \leq \sum_{s \in S} f_{i,j}^{(p)}(s, d) \leq \tau_p \cdot \gamma_{ij} \cdot \mathbf{I}_{E_p^{\text{ex}}}((i, j)) \tag{5}$$

At each node flow conservation must hold, which expresses that any incoming flows into a node must depart from the node, except for the sources and the destinations. Flow conservation is expressed as follows

$$\sum_{p=1}^{P} \left(\sum_{j:(i,j) \in E_p^{\text{ex}}} f_{i,j}^{(p)}(s, d) - \sum_{j:(j,i) \in E_p^{\text{ex}}} f_{j,i}^{(p)}(s, d) \right) = \sigma_i \tag{6}$$

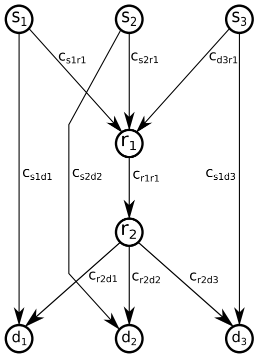

Figure 3: Multicast scenario with three sources and three destinations

where σ_i is r_s when it is a source, $-r_s$ when it is a destination and 0 otherwise. As in [6], we normalize and bound τ_p by

$$\sum \tau_p = 1 \tag{7}$$

and

$$0 \leq \tau_p \leq 1. \tag{8}$$

The linear constraints shown in (5)-(8) are the combination of the constrains described in [4] and [6]. The merging of the linear constraints make the switching from plain routing to network coding and the scheduling over different network realizations N_p^{ex} possible, which was not done before. The overall optimization problem is expressed as

$$\max \sum_{s \in S} \log(1 + r_s) \tag{9}$$

$$\text{s.t.}(5) - (8). \tag{10}$$

By solving the problem, we obtain the multicast rate for each source r_s, the timeshares τ_p for each realization and the flow $f_{i,j}(s,d)$ at each link for the respective source and destination pair.

IV. Simulation Results

We investigate the proposed framework for a WMN network with three sources and three destinations as depicted in Figure 3. We analyze the performance of the network under different bottleneck cases. We look into three cases, namely case 1 first hop, case 2 intermediate hop and case 3 last hop bottlenecks. The first hop case has two variants. The first variant is that the middle route s_1 to r_1 is a bottleneck and the second variant is that the direct link from source node s_1 to destination node d_1 is a bottleneck. The intermediate hop case covers the middle link r_1 to r_2. The last case deals with the bottleneck from r_2 to d_1. We will focus on single link bottlenecks, hence all other links in the network have a fixed link capacity of 1. Note that the WMN in Figure 3 is symmetric, i.e., it is sufficient to only consider the three cases mentioned above. We plot the sum rate over the channel factor γ of the bottleneck, where the sum rate is the sum over the multicast rates of the sources and the channel factor γ of the bottleneck is in the range between $0 \leq \gamma \leq 1$.

We show results of our framework in comparison to three benchmark schemes. The first benchmark scheme uses plain routing at the NET and BC scheduling at the MAC (PRBCS). PRBCS enforces that all sources in the network have the same multicast rate. The second benchmark scheme uses network coding at the NET and BC scheduling at the MAC (NCBCS). The third scheme is a greedy routing approach at the NET and also uses BC scheduling at the MAC (GRBCS). The GRBCS uses plain routing to achieve the maximum sum rate in the network, in contrast to the first scheme where each source achieves the same multicast rate. In order to demonstrate the advantage of extended virtualization, we also compare our proposed framework with the framework from [6] in case 3. For the framework from [6], we consider three utility functions, namely the sum-log rate utility, the sum rate utility and the max-min utility.

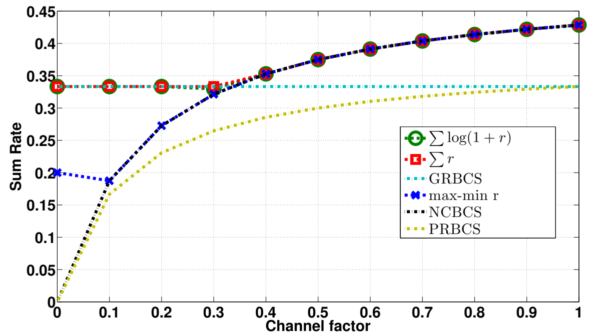

Figure 4: Sum rate vs γ for case 1a) intermediate link

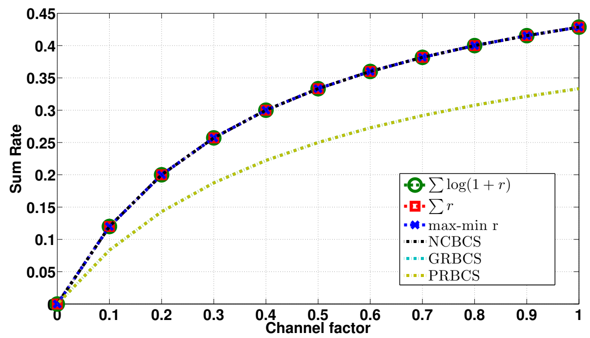

Figure 5: Sum rate vs γ for case 2 intermediate hop

A. Case 1 - First Hop

The first variant of the first hop case is illustrated in Figure 4. We see that PRBCS provides the lowest performance, which is limited to the maximum achievable multicast rate of the weakest source. The next two curves show the performances of the NCBCS and our framework with a max-min utility. Both schemes have the same performance when the channel factor γ is in the region of $0.1 \leq \gamma \leq 1$. Only for an outage of the link for $\gamma = 0$, the max-min curve is better. Here, the max-min can activate a source, while PRBCS and NCBCS

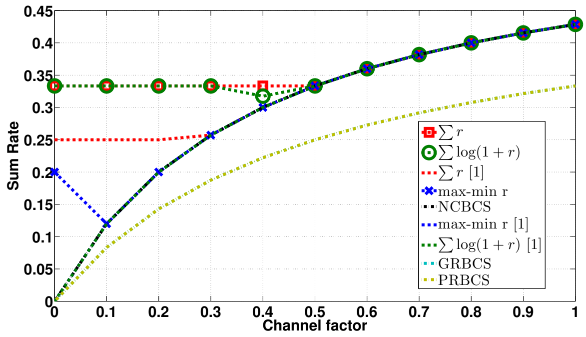

Figure 6: Sum rate vs γ for case 3 last hop

cannot. The reasons are that NCBCS only works if all three source nodes can apply network coding and thus for $\gamma = 0$, the multicast rate of each source is zero, for PRBCS equality must hold and since the weakest source achieves a multicast rate of zero the overall rate is zero. In contrast, the max-min is not constrained by equality and therefore the multicast rates are individually maximized. The upper three curves represent the GRBCS, our framework with sum rate utility and with sum log utility, respectively. The GRBCS achieves a constant sum rate, which our framework outperforms by switching from plain routing to networking coding. The switching occurs as soon as γ is greater than 0.3, where network coding achieves higher rates compared to plain routing. In Figure 4 we can can see that our framework can utilize the advantages of both isolated approaches.

The performance with respect to the sum rate for the second variant of the first hop does not differ from the first variant. Therefore, we do not present the results in this paper, due to space limitations.

B. Case 2 - Intermediate Hop

The intermediate hop has the strongest impact on our framework, since the bottleneck involves an essential forwarding node. In this case, the best strategy is to keep the number of transmissions as low as possible. Therefore, network coding at the NET and BC at the MAC is the best strategy. Our framework follows this strategy for all three utility functions. In Figure 5, we see that our framework has the same performance as NCBCS. As already said, the number of transmission must be low and hence PRBCS is outperformed by all the schemes which use network coding, as in case 1. Further, we see that PRBCS and GRBCS achieve the same sum rate over the complete range of γ.

From case 2, we can conclude that bottlenecks at essential links cannot be compensated and that network coding is the optimal mechanism, which our framework chooses.

C. Case 3 - Last Hop

The third case shows the gain achieved by the extended virtualization, which is illustrated in Figure 6. For this case, we added the performance curves of the framework from [6], where the virtualization is performed only once. Here, again we see that PRBCS, NCBCS and the max-min utility provide the lowest performance, compared to the sum rate and sum log utilities, for a low γ. Further, as in case 2, the GRBCS overlaps with the benchmark routing. As soon as γ is greater than 0.5, the scheme of [6] and NCBCS perform as well as our framework, at this point the best strategy is to use network coding and BC communications. Nevertheless, our framework outperforms all other schemes. For both the sum and sum log utility, our framework achieves the highest sum rate up to a γ of 0.5, at which point all schemes converge to the network coding solution. The reason for the better performance, for a low γ is caused by the extended virtualization. Since with the extended virtualization, UC and MC communications replace the low performance BC communication.

V. Conclusion

In this paper, we analyzed the impact of unequal link capacities in WMNs. We presented a cross-layer approach, which can cope with unequal link capacities, e.g., in the form of bottlenecks. We combined currently available optimization problems to enable transitions between routing mechanisms and communication types at the respective layer, where the proposed new scheduler enables the switching between the communication types of the PHY. We emphasized that extended virtualization is important to fully utilize the inherent BC capabilities of the wireless medium. Extended virtualization allows us to represent UC, MC and BC communications of the PHY via the MAC at the NET. Further, the simulation results show the advantage of switching between communication types and routing mechanisms and the gains achievable applying extended virtualization especially for low channel factors.

VI. Acknowledgment

This work has been funded by the German Research Foundation (DFG) in the Collaborative Research Center (SFB) 1053 „MAKI – Multi-Mechanisms-Adaptation for the Future Internet“.

References

[1] Y. Sagduyu and A. Ephremides, “On Joint MAC and Network Coding in Wireless Ad Hoc Networks,” *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3697–3713, Oct 2007.

[2] J. Yuan, Z. Li, W. Yu, and B. Li, “A Cross-Layer Optimization Framework for Multihop Multicast in Wireless Mesh Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 11, pp. 2092–2103, Nov 2006.

[3] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, “Network Information Flow,” *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul 2000.

[4] Z. Li, B. Li, D. Jiang, and L. C. Lau, “On Achieving Optimal Throughput with Network Coding,” in *Proceedings IEEE INFOCOM 2005.*, vol. 3, March 2005, pp. 2184–2194 vol. 3.

[5] A. Ramamoorthy, J. Shi, and R. Wesel, “On the Capacity of Network Coding for Random Networks,” *IEEE Transactions on Information Theory*, vol. 51, no. 8, pp. 2878–2885, Aug 2005.

[6] R. Niati, A. Banihashemi, and T. Kunz, “Throughput and Energy Optimization in Wireless Networks: Joint MAC Scheduling and Network Coding,” *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 1372–1382, March 2012.