

Mousie Fasil, Hussein Al-Shatri, Stefan Wilk and Anja Klein, "Application-Aware Cross-Layer Framework: Video Content Distribution in Wireless Multihop Networks," in *Proc. of 2015 IEEE 26th Annual International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC): MAC and Cross-Layer Design (IEEE PIMRC2015 MAC & Cross-Layer Design)*, Hong Kong, China, August 2015.

©2015 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this works must be obtained from the IEEE.

# Application-Aware Cross-Layer Framework: Video Content Distribution in Wireless Multihop Networks

Mousie Fasil<sup>1</sup>, Hussein Al-Shatri<sup>1</sup>, Stefan Wilk<sup>2</sup>, and Anja Klein<sup>1</sup>

<sup>1</sup> Communications Engineering Lab, Technische Universität Darmstadt, Merckstrasse 25, 64283 Darmstadt, Germany

<sup>2</sup> Distributed Multimedia Systems, Technische Universität Darmstadt, Rundeturmstr. 10, 64283 Darmstadt, Germany

Email:<sup>1</sup>{m.fasil, h.shatri, a.klein}@nt.tu-darmstadt.de, <sup>2</sup>{swilk}@cs.tu-darmstadt.de

**Abstract**—Scalable video coding (SVC) can overcome the user heterogeneity issue, e.g., different screen resolutions or different connectivities, in video-streaming. In wireless multihop networks, the performance of SVC is not adequate, because SVC cannot adapt the lower layers. In order to adapt to changing environmental conditions, e.g., network topology, available resources or channel conditions, a cross-layer framework is required. We propose a new application-aware cross-layer framework which utilizes SVC, network structures and communication types at APP, NET, DLL and PHY layers together. Further, our application-aware cross-layer framework performs transitions at different layers to find the best combination of mechanisms. This is achieved by the following steps. First, we apply a graph-based approach to integrate all mechanisms on the different layers in a single graph. Secondly, video layers are modeled in the graph as virtual sources. Thirdly, we perform an optimal mapping from video layer data rates to physical layer rates. Fourthly, we formulate a multi-source sum rate optimization problem which chooses the best video layer distribution among users and the best combination of mechanisms at all layers. Finally, we demonstrate that our application-aware cross-layer framework outperforms current approaches.

## I. INTRODUCTION

In 2014, video-streaming was responsible for 55 % of the worldwide mobile traffic [1]. It is estimated that by the end of 2019, it will increase to three-fourths of the total worldwide mobile traffic [1]. The performance of a video-streaming service in a wireless multihop network (WMN) highly depends on the degree of user heterogeneity. User heterogeneity includes the diversity of the mobile device, concerning, e.g., screen resolution and processing power as well as device connectivity and channel conditions [2]. Hence, in the presence of one or multiple weak users the performance of a video-streaming service will be low in terms of service quality and resource utilization [3].

At the application layer (APP), adaptive video streaming technologies have been proposed in order to address device heterogeneity as well as changing network conditions [4]. One adaptive streaming approach is the video encoding technology scalable video coding (SVC) [5], [6].

SVC breaks the limitation of video-streaming by introducing video layers, which allows to serve each user with an individual video quality [5]. However, SVC cannot address the variations in a WMN like changing network topology, available resources and channel conditions. Since, SVC cannot adapt the lower layers, a cross-layer approach is needed.

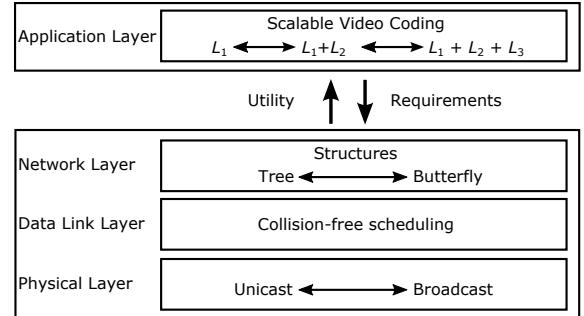


Figure 1: Application-aware cross-layer framework

Several works [7], [8] and [9] present a cross-layer approach, where they combine SVC and network coding [10]. The papers show that considering APP and network layer (NET) together increases the performance compared to approaches without a layered video distribution scheme. Nevertheless, APP and NET cannot adapt to changes in available resources and channel conditions. Thus, they cannot adapt to variations at the data link layer (DLL) and at the physical layer (PHY). In other related works [11], [12] and [13], APP is combined with DLL. At the DLL, a resource allocation problem is formulated which maps SVC video layer rates such that every user can receive at least the basic video layer and the remaining resources are allocated to the enhancement video layers. The presented results are limited to the one hop case and different communication types, e.g. broadcast (BC) and unicast (UC), where not taken into account. In [14], we present a cross-layer framework which considers NET, DLL and PHY. Further, we demonstrate the advantage of transitions, where the cross-layer framework switches between different mechanisms on different layers, but do not take into account APP requirements. In this paper, we propose the combination of our cross-layer framework and SVC, where we take into account APP, NET, DLL and PHY layers together. Our proposed application-aware cross-layer framework is illustrated in Fig. 1. First, we utilize the concept of transitions, where the application-aware cross-layer framework switches between different sets of active video layers, e.g.,  $L_1$  or  $L_1 + L_2$ , between network structures, e.g., tree and butterfly, and between communication

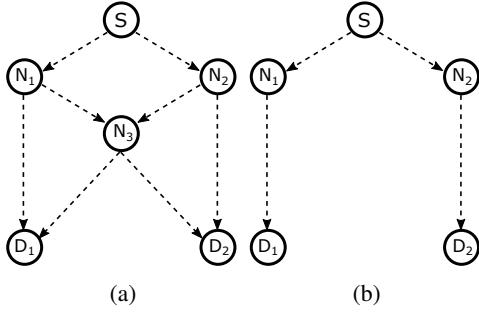


Figure 2: Network structures (a) butterfly structure and (b) tree

types, e.g., UC and BC, simultaneously. Secondly, we model the different mechanisms shown in Fig. 1 as a single graph. In this graph, video layers are represented as virtual sources. Thirdly, we propose an optimal mapping between the video layer data rates and PHY rates. Fourthly, we formulate a per video layer per destination rate optimization problem. The remainder of the paper is organized as follows. In Section II, we present the system model. We model the cross-layer framework as a graph. In Section III, we formulate a rate mapping problem to map the video layer data rate requirements to PHY rate requirements, and the sum rate optimization problem of our application-aware cross-layer framework. We show simulation results in Section IV, where we evaluate our proposed application-aware cross-layer framework against schemes without the possibility to perform transitions and our previous cross-layer approach. The paper is concluded in Section V.

## II. SYSTEM MODEL

In this section, we describe the system model and show that SVC and different mechanisms at different layers can be modeled by a single graph. We start with the NET, which naturally is modeled as a graph, where we utilize different network structures as NET mechanisms. We continue with the different communication types which are PHY mechanisms. Communication types are integrated into the graph through the concept of virtualization, see [14] and [15]. Virtualization will extend the graph such that transitions between different NET mechanisms and PHY mechanisms can be performed. The extended graph has to be split into subgraphs, in order to avoid collisions between transmitting nodes. Therefore, we describe a collision-free scheduler at the DLL. Finally, we extend the concept of virtualization to integrate SVC in the graph, by representing each video layer as a virtual source. 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. We will refer to the video forwarded in the lower network layers as messages.

### A. NET Mechanisms: Network Structures

In a WMN, it is beneficial to utilize different network structures at the NET [14], to adapt to changes in the network

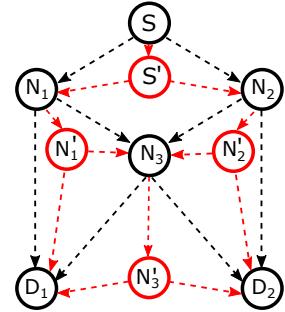


Figure 3: Extended network graph obtained by virtualization

topology. A network structure is a set of nodes in a WMN which cooperate in order to deliver the messages to the destination. We model the WMN as a directed 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 connections between the nodes. The set of nodes contains three disjoint subsets, the subset  $S \subset V$  of source nodes, the subset  $D \subset V$  of destination nodes and the subset  $N \subset V$  of relay nodes. 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.

Throughout the paper, we consider two network structures: the tree and the butterfly. In Fig. 2 we illustrate an example of the butterfly structure and the tree structure. The main difference between the two structures is that in the butterfly structure, network coding can be utilized. In Fig. 2 (a), the butterfly structure combines two incoming messages from relay  $N_1$  and  $N_2$  at relay  $N_3$  using network coding into one outgoing message. In a tree structure messages are not combined. The advantage of the tree is that the number of relays involved in the forwarding are low in comparison to the butterfly. The advantage of the butterfly is that multiple messages can be coded together at relay node. However, the achievable sum rate depends not only on the network structure, but also on the available PHY mechanisms.

### B. PHY Mechanisms: Communication Types

At the PHY, a node can choose between BC and UC to forward a message. In BC, a node forwards a message to all neighboring nodes simultaneously, where the forwarding node adjusts the rate with respect to the weakest neighbor. In UC, a node forwards a message such that only one node is the intended receiver. Hence, UC can achieve high transmission rates, but it requires multiple time resources for multiple receivers. On the other hand, BC needs only one time resource but it achieves a lower transmission rate. Therefore, a node needs to choose between the different communication types such that the overall rate is maximized.

In order to optimally decide which communication type to choose, the different communication types need to be included in the graph. The differentiation between UC and BC can be done by applying virtualization, cf. [14], [15]. As an example, we perform virtualization on the graph shown in Fig. 3.

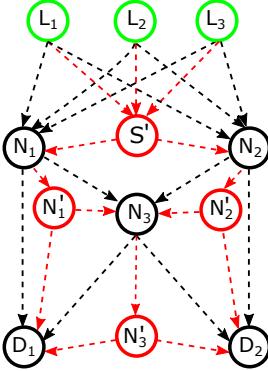


Figure 4: Source virtualization node  $S$  is replaced by the video layers as virtual sources

Virtualization extends a given graph by adding virtual nodes and virtual links to the graph. For each node with at least two outgoing links, a virtual node is added to the network graph. For instance, the virtual node  $S'$  has one incoming edge from the original node  $S$  and two outgoing virtual links are added between  $S'$  and the original receiving nodes  $N_1$  and  $N_2$ . The capacity of the virtual links are set to the minimum of the original outgoing links between the transmitting node, e.g.,  $S$  and the receiving nodes, e.g.,  $N_1$  and  $N_2$ . This results in an extended network graph  $G^{\text{ex}} = (V^{\text{ex}}, E^{\text{ex}})$  as seen in Fig. 3, where the black edges represent the UC communication links, while the red edges represent the BC communication links. In the same manner PHY multicast can be taken into account, see [14].

### C. DLL: Collision-free Scheduling

In a WMN, it is necessary to coordinate the communications between nodes in order to avoid collisions. A collision occurs when a node is transmitting and receiving at the same time or when a node is receiving multiple messages at the same time. This collision-free scheduling is done at the medium access layer, which is a sub-layer of the DLL. The collision-free scheduler splits  $G^{\text{ex}}$  into  $p$  subgraphs. Each subgraph  $G_p^{\text{ex}}$  contains a subset of vertices  $V_p^{\text{ex}}$  and edges  $E_p^{\text{ex}}$ , which do not conflict with each other. The union over all  $p$  sub-graphs

$$\bigcup_{p=1}^P G_p^{\text{ex}} = G^{\text{ex}}$$

is the extended graph shown in Fig. 3. The scheduler allocates resources to each  $G_p^{\text{ex}}$  such that the sum rate is maximized and collisions are avoided.

Now, we can optimally choose the network structure and the PHY mechanism over  $G^{\text{ex}}$ . Further, we can perform transitions when the environmental conditions change, namely we can switch between the butterfly and the tree structure at the NET, while at the PHY we can switch between UC and BC.

### D. APP: Video Layers as Virtual Sources in a Graph

In SVC, a video is divided into  $K$  video layers  $L_k \in \{L_1, L_2, \dots, L_K\}$ . Each video layer has a specific data rate requirement  $B_k \in \{B_1, B_2, \dots, B_K\}$ . Further, we assume that the video layers have to be received in successive order.

This means, without receiving the first video layer, the second video layer cannot be decoded and so on and so forth. Let us assume that each video layer contains individual information and hence, each video layer has independent messages to forward to the destinations. Therefore, we can model each video layer as an individual source, which we represent by virtualizing the physical source as multiple virtual sources. We replace the original source in  $G^{\text{ex}}$  with virtual sources as shown in Fig. 4. For every video layer  $L_k$ , we add a virtual source to  $G^{\text{ex}}$  where they replace the original source node  $S$ . The result is shown in Fig. 4, the virtual sources are connected to the nodes through the UC and BC links of the original source, which is indicated through the black and red edges. We have shown that an application-aware cross-layer framework can be modeled with a graph-based approach. We represent video layers as virtual sources, in order to consider the APP in the graph, which was not done before. Further, the cross-layer framework is capable of switching between network structures at the NET and between communication types at the PHY. This allows us to fully utilize transitions at the lower layers, which was not applied before for video-streaming in WMNs.

## III. APPLICATION-AWARE CROSS-LAYER FRAMEWORK FOR WIRELESS MULTIHOP NETWORKS

In the previous section, we showed that the application-aware cross-layer framework can be represented in a single graph. Based on this graph, we formulate a multi-source sum rate optimization problem, where video layers are the sources and the destinations can receive a video layer  $k$  only if they can receive all the video layers 1 to  $k - 1$ . The multi-source sum rate optimization problem can be expressed as a binary linear problem (BLP). The solution of the BLP contains the optimal sum rate and the optimal number of subscribers for every video layer.

However, the BLP can only be solved optimally under the condition that video layer data rate requirements are mapped to PHY rate requirements. Thus, a mapping problem has to be formulated and solved, before the BLP can be solved. The mapping problem can be expressed as a binary non-linear problem (BNLP).

Therefore, we discuss the BNLP first in Section III-A. We formulate the mapping problem and shortly elaborate how a heuristic approach can be applied to find the mapping. With the obtained mapping solution, we can solve the BLP, which we formulate in Section III-B.

### A. Video Layer Date Rate to Physical Layer Rate Mapping

Since the required video layer data rate is much higher than the PHY rate, it is important to map the required video layer data rate requirement to PHY rates. Throughout this paper, we investigate the optimization problems from the point of one representative subcarrier, i.e., average achievable PHY rate. We formulate a rate mapping problem based on the number of video layers  $L_1$  to  $L_K$ , the relative ratios of the video layer data rates  $B_1$  to  $B_k$  and the achievable capacity on

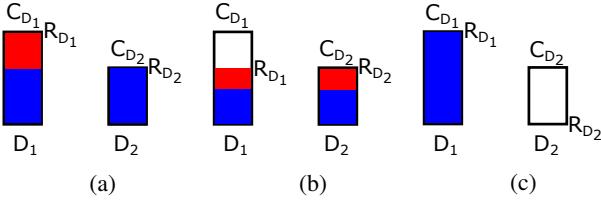


Figure 5: Mapping the video layer data rate requirement to PHY rate requirement where (a) is optimal and (b) and (c) are suboptimal in terms of sum rate

each subcarrier. We illustrate the rate mapping problem, by the following toy example. Let us assume a scenario with two destinations  $d_1$  and  $d_2$ , one sub-carrier and two video layers  $L_1$  and  $L_2$  with unequal data rate requirement  $B_1$  and  $B_2$ , where  $B_1 = \frac{1}{2}B_2$ . Further, let us assume the achievable physical rate at the destinations  $d_1$  and  $d_2$  are  $R_{d_1}$  and  $R_{d_2}$ , respectively. The question now is how to allocate  $L_1$  and  $L_2$  such that the sum of  $R_{d_1}$  and  $R_{d_2}$  is maximized. In Fig. 5, we illustrate three possible solutions. In Fig. 5 (a), we map  $L_1$  and  $L_2$  such that  $d_1$  can receive them both, but  $d_2$  only receives  $L_1$ , while in Fig. 5 (b), we map  $L_1$  and  $L_2$  such that  $d_1$  and  $d_2$  can receive both video layers, and in Fig. 5 (c), we map in such way that only  $d_1$  can receive  $L_1$  and  $d_2$  receives nothing. The solution presented in Fig. 5 (a) achieves the maximum sum rate in comparison to the solution in Fig. 5 (b) and (c). Therefore, the solution presented in Fig. 5 (a) is optimal in terms of sum rate maximization.

Our aim is a solution as seen in Fig. 5 (a), where the video layer data rate requirements are mapped to PHY rate requirements such that the sum rate is maximized. Therefore, we formulate a rate mapping problem where  $u_{k,d}$  is a binary value which determines if destination  $d$  receives video layer  $k$  and  $r_k$  is the PHY rate requirement of video layer  $L_k$ .

$$\max \sum_d \sum_k u_{k,d} \cdot r_k, \quad (1)$$

subject to

$$\sum_d \sum_k u_{k,d} \cdot r_k \leq c_d, \quad (2)$$

$$\sum_k u_{k,d} = 1, \quad \forall d \in \{1, 2, \dots, D\} \quad (3)$$

$$r_k = \sum_{u=1}^k b_u, \quad \forall k \in \{1, 2, \dots, K\} \quad (4)$$

$$u_{k,d} \in \{0, 1\}, \quad (5)$$

$$r_K > r_{K-1} > \dots > r_1. \quad (6)$$

We solve the rate mapping problem in (1) - (6) with the following heuristic. We begin with the destination with the highest capacity  $c_d$  and allocate all video layers to it. Based on this allocation we determine the number of video layers and the corresponding rate at all other nodes. Further, we determine the achievable sum rate. In the second step, we allocate all video layers to the user with the second highest  $c_d$  and similarly determine the achievable sum rate. If the

sum rate is greater than before, we store the new video layer allocation and repeat the step. The heuristic stops when the achievable sum rate decreases. At this step, we have achieved the maximum rate and therefore the best mapping of the layers to PHY rates.

In the next section, the obtained PHY rate requirement  $r_k$  for each video layer  $L_k$  are injected into the multi-source sum rate optimization problem to determine the maximum sum rate and the optimal number of video layer subscribers.

### B. Multi-Source Sum Rate Optimization

We can formulate the video-streaming scenario in a WMN as a multi-source problem, since each video layer is represented as a virtual source. Our goal is to maximize the sum rate in the network and to maximize the number of destination nodes which receive  $L_1$  to  $L_K$ . Our optimization problem is similar to [14], with the extension that interdependences between sources exists. We maximize the sum rate over all virtual sources 1 to  $K$  and destinations 1 to  $D$ . Thus, our utility function can be expressed as

$$\max \sum_k^K \sum_d^D r_{k,d} \quad (7)$$

where  $r_{k,d}$  expresses the rate achieved between virtual source  $k$  and destination  $d$ . The rate between  $k$  and  $d$  is constrained by the maximum flow in the network. We define the flow from  $k$  to  $d$  over the link from node  $i$  to node  $j$  in the  $p$ -th subgraph as  $f_{i,j}^{(p)}(k, d)$ . At each node, the flow conservation must hold, which expresses that any incoming flow into a node must depart from the node, except for virtual sources and destinations. The flow conservation constraint is expressed by,

$$\sum_{p=1}^P \left( \sum_{j:(i,j) \in E_p^{\text{ex}}} f_{i,j}^{(p)}(k, d) - \sum_{j:(j,i) \in E_p^{\text{ex}}} f_{j,i}^{(p)}(k, d) \right) = \sigma_i, \quad (8)$$

$$\forall i \in V^{\text{ex}}, k = \{1, \dots, K\}, d = \{1, \dots, D\}$$

where  $\sigma_i$  is equal to  $r_{k,d}$  when it is a virtual source, equal to  $-r_{k,d}$  when it is a destination and equal to 0 otherwise. Further, each flow is upper bounded by a capacity constraint, where the capacity in subgraph  $p$  depends on the link capacity  $c_{i,j}$  between nodes  $i$  and  $j$  and the duration the link is active in the  $p$ -th subgraph, which is determined by the timeshare  $\tau_p$ . A link is included in a sub-graph  $G_p^{\text{ex}}$  if the indicator function  $\mathbf{I}_{E_p^{\text{ex}}}((i,j))$  is one, else it is zero. The indicator function is written as

$$\mathbf{I}_{E_p^{\text{ex}}}((i,j)) = \begin{cases} 1, & \text{if } (i,j) \in E_p^{\text{ex}} \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

We express the capacity constraint as

$$0 \leq \sum_k^K f_{i,j}^{(p)}(k, d) \leq \tau_p \cdot c_{i,j} \cdot \mathbf{I}_{E_p^{\text{ex}}}((i,j)), \quad (10)$$

$$\forall (i,j) \in E_p^{\text{ex}}, p = \{1, \dots, P\}, d = \{1, \dots, D\}.$$

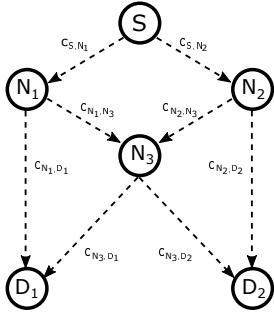


Figure 6: Scenario with one source and two destinations

SVC requires that video layers are received in successive order. This APP constraint is described by

$$x_{k,d} \geq x_{k+1,d} \quad \forall k = \{1, \dots, K-1\}, d = \{1, \dots, D\}, \quad (11)$$

where  $x_{k,d}$  is equal to one if video layer  $k$  is received by destination  $d$  and zero otherwise. Thus, if  $x_{k,d}$  is one, then the rate  $r_{k,d}$  is equal to the physical rate  $r_k$ . This is written as

$$r_{k,d} = x_{k,d} \cdot r_k, \quad \forall k = \{1, \dots, K\}, d = \{1, \dots, D\}. \quad (12)$$

Finally, the time shares are normalized and bounded as

$$\sum_{p=1}^P \tau_p = 1 \quad (13)$$

$$0 \leq \tau_p \leq 1, \quad \forall p = \{1, \dots, P\}. \quad (14)$$

The multi-source optimization problem expressed in Eq. (7 - 14) is a BLP. By solving the BLP, we obtain the maximum sum rate in the system, a binary matrix containing  $x_{k,d}$ , the timeshare  $\tau_p$  for each sub-graph and the flow  $f_{i,j}(k, d)$ .

#### IV. SIMULATION RESULTS

We investigate the performance of the proposed application-aware cross-layer framework for a scenario with one source and two destinations. We assume that a video is separated into three video layers  $L_1$ ,  $L_2$  and  $L_3$ . Further, we assume that the video layers have equal data rate requirements  $B_1 = B_2 = B_3$ . The achievable sum rate is considered as a performance measure. We determine the link capacity between nodes  $i$  and  $j$  as

$$c_{i,j} = \log_2(1 + \frac{|h|^2}{d_{i,j}^\alpha} \cdot \frac{P_T}{P_N}) \quad (15)$$

where  $\alpha$  is the path loss exponent,  $h$  is the channel gain,  $P_T$  is the transmit power,  $P_N$  is the noise power at the receiver and  $d_{i,j}$  is the distance between node  $i$  and  $j$  in meters. Moreover, we assume that the received noise power at all nodes is equal to  $P_N$ . The simulation parameters are given in Table I.

For the scenario shown in Fig. 6, we determine the upper bound, where the sum rate is determined without considering the video layers. In addition, we compare our application-aware cross-layer framework with our previous cross-layer framework [14], which serves both users with equal rate.

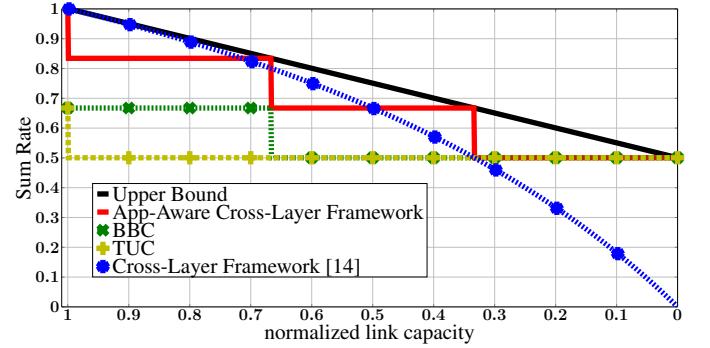


Figure 7: Sum rate vs. normalized link capacity for bottlenecks at  $D_2$

Furthermore, we look at two additional schemes. The first one is using the butterfly structure at the NET and BC at the PHY, which we abbreviate with BBC. The second scheme is utilizing the tree structure at the NET and UC at the PHY, which we abbreviate with TUC. The rate mapping is done beforehand and is not changed throughout the simulation. For our first evaluation, we analyze the sum rate for the bottleneck case at destination  $D_2$ . First, we normalize the link capacities  $c_{i,j}$  by  $c_{\max}$ , where  $c_{\max}$  is the capacity calculated for an average distance of  $d_{i,j} = 10$  m. This results, in a normalized link capacity  $\gamma_{i,j}$  which ranges between  $0 \leq \gamma_{i,j} \leq 1$ . For the bottleneck case at destination  $D_2$ , we set all  $\gamma_{i,j}$  to one except the normalized link capacities to  $D_2$  which we tune simultaneously between  $0 \leq \gamma_{D_2} \leq 1$ . In Fig. 7, we plot the sum rate over  $\gamma_{D_2}$ . The results show that our application-aware cross-layer framework outperforms BBC and TUC for  $\gamma_{D_2} \geq 0.325$  and the cross-layer framework between  $\gamma_{D_2} = 0.5$  and  $\gamma_{D_2} = 0$ . The cross-layer framework is close to the upper bound until  $\gamma_{D_2} = 0.7$ . For decreasing  $\gamma_{D_2}$ , the sum rate achieved by the cross-layer framework degrades and becomes 0 at  $\gamma_{D_2} = 0$ . This due to the fact that the cross-layer framework distributes the video with respect to the smallest rate of  $D_1$  and  $D_2$ .

The deactivation of video layers is shown in Fig. 7. The application-aware cross-layer framework provides  $L_1$ ,  $L_2$  and  $L_3$  to  $D_1$  and  $D_2$  at  $\gamma_{D_2} = 1$ , while BBC and TUC provide only  $L_1$  to  $D_2$  and  $L_1$ ,  $L_2$  and  $L_3$  to  $D_1$ . As  $\gamma_{D_2}$  decreases, more and more video layers are deactivated at  $D_2$ . In the range  $1 \geq \gamma_{D_2} > 0.7$ ,  $D_2$  receives  $L_1$  and  $L_2$ , in the range  $0.7 \geq \gamma_{D_2} \geq 0.325$ ,  $D_2$  receives only  $L_1$  and thereafter, no video layer is received at  $D_2$ . However,  $D_1$  receives all three layers over the whole range of  $\gamma_{D_2}$ .

In Fig. 8, we analyze the performance of our application-

Transmit power ( $P_T$ )	0 dBm
Noise power ( $P_N$ )	-80 dBm
Path loss exponent ( $\alpha$ )	4
Average channel gain ( $E\{ h ^2\}$ )	1
Distance between node $i$ and $j$ ( $d_{i,j}$ )	10 m - 100 m

Table I: Simulation parameters

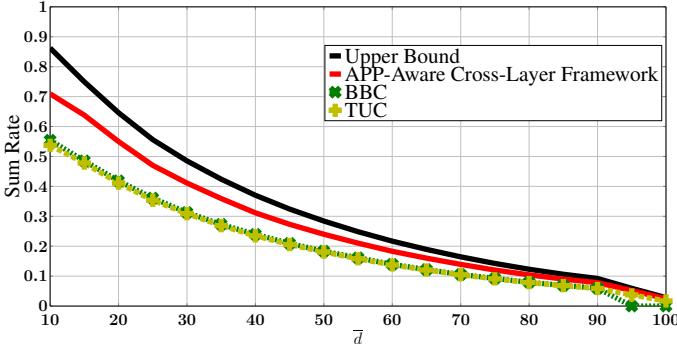


Figure 8: Sum rate vs. average distance between nodes

aware cross-layer framework, BBC and TUC over a changing average distance between all nodes  $i$  and  $j$ , where the average distance  $\bar{d}_{i,j}$  is in the range between 10 and 100 meters. Our application-aware cross-layer framework outperforms both BBC and TUC. For a low average distance, the application-aware cross-layer framework outperforms BBC and TUC, because it can perform transitions. With increasing average distance, the capacity between the nodes decreases which leads to a decrease in sum rate. The performance of the application-aware cross-layer framework decreases, since the opportunities to perform transitions decrease as well. At high average distance, the probability of a link disconnection is high which results in zero sum rate for BBC at roughly  $\bar{d}_{i,j} = 95$  m and for TUC and the application-aware cross-layer framework at  $\bar{d}_{i,j} = 100$  m.

## V. CONCLUSION

In this paper, we propose an application-aware cross-layer framework for WMNs. We combine APP, NET, DLL and PHY together. This is achieved by modeling the different layers and their respective mechanisms in a single graph. We integrate SVC through the concept of virtual sources, where we add virtual sources for every video layer. In addition, we show how a mapping from the data rate requirement to a PHY rate requirement is performed. Finally, we formulate a multi-source sum rate problem as a BLP and solve it accordingly. Our simulation results, show that an application-aware cross-layer framework outperforms schemes without the possibility to perform transitions and surpasses our previously proposed underlay cross-layer framework, in the case of heterogeneity of user capabilities.

## VI. ACKNOWLEDGMENT

This work has been funded by the German Research Foundation (DFG) as part of project C01 within the Collaborative Research Center (CRC) 1053 – MAKI. The authors would like to thank Sabrina Müller (B03) for the insightful discussions about non-linear optimization, Julius Rückert (C03) and Denny Stohr (B01) for their instructive inputs on scalable video coding.

## REFERENCES

- [1] “White paper: cisco visual networking index data traffic forecast update, 2014-2019,” Cisco, Tech. Rep., 2015.
- [2] X. Zhu, P. Agrawal, J. Pal Singh, T. Alpcan, and B. Girod, “Rate allocation for multi-user video streaming over heterogeneous access networks,” in *Proc. ACM 15th International Conference on Multimedia*. ACM, 2007, pp. 37–46.
- [3] L. Zhou, B. Geller, B. Zheng, A. Wei, and J. Cui, “System scheduling for multi-description video streaming over wireless multi-hop networks,” *IEEE Transactions on Broadcasting*, vol. 55, no. 4, pp. 731–741, December 2009.
- [4] Y. Ye and P. Andrivon, “The scalable extensions of hevc for ultra-high-definition video delivery,” *IEEE MultiMedia*, vol. 21, no. 3, pp. 58–64, July 2014.
- [5] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the H.264/AVC standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, September 2007.
- [6] Y. Sánchez de la Fuente, T. Schierl, C. Hellge, T. Wiegand, D. Hong, D. De Vleeschauwer, W. Van Leekwijck, and Y. Le Louédec, “iDASH: improved dynamic adaptive streaming over http using scalable video coding,” in *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, ser. MMSys ’11. New York, NY, USA: ACM, 2011, pp. 257–264. [Online]. Available: <http://doi.acm.org/10.1145/1943552.1943586>
- [7] J. Zhao, F. Yang, Q. Zhang, Z. Zhang, and F. Zhang, “Lion: Layered overlay multicast with network coding,” *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 1021–1032, October 2006.
- [8] S. Lakshminarayana and A. Eryilmaz, “Multirate multicasting with intralayer network coding,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 4, pp. 1256–1269, August 2013.
- [9] H. Cui, D. Qian, X. Zhang, C. Jing, and Y. Sun, “Joint source-network coding optimization for video streaming over wireless multi-hop networks,” in *Proc. IEEE Vehicular Technology Conference (VTC Spring)*, May 2012, pp. 1–5.
- [10] 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, July 2000.
- [11] J. Villalon, P. Cuenca, L. Orozco-Barbosa, Y. Seok, and T. Turletti, “Arsa: a cross-layer auto rate selection multicast mechanism for multirate wireless lans,” *IET Communications*, vol. 1, no. 5, pp. 893–902, October 2007.
- [12] S. Kwack, H. Seo, and B. G. Lee, “Suitability-based subcarrier allocation for multicast services employing layered video coding in wireless OFDM systems,” in *Proc. IEEE Vehicular Technology Conference (VTC Fall)*, September 2007, pp. 1752–1756.
- [13] H. Deng, X. Tao, T. Xing, and J. Lu, “Resource allocation for layered multicast streaming in wireless OFDMA networks,” in *Proc. IEEE International Conference on Communications (ICC)*, June 2011, pp. 1–5.
- [14] M. Fasil, A. Kuehne, and A. Klein, “Node virtualization and network coding: Optimizing data rate in wireless multicast,” in *Proc. International Symposium on Wireless Communications Systems (ISWCS)*, August 2014, pp. 573–578.
- [15] 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.