Mousie Fasil, Hussein Al-Shatri, Stefan Wilk and Anja Klein, ”A Network-Centric View on DASH in Wireless Multihop Networks,” in *Proc. of Vehicular Technology Conference VTC-Fall*, Montral, Canada, September 2016.

A Network-Centric View on DASH in Wireless Multihop Networks

Mousie Fasil[1], Hussein Al-Shatri[1], Stefan Wilk[2], and Anja Klein[1]
[1] Communications Engineering Lab, Technische Universität Darmstadt, Merckstrasse 25, 64283 Darmstadt, Germany
[2] Distributed Multimedia Systems, Technische Universität Darmstadt, Rundeturmstr. 10, 64283 Darmstadt, Germany
Email:[1]{m.fasil, h.shatri, a.klein}@nt.tu-darmstadt.de, [2]{swilk}@cs.tu-darmstadt.de

***Abstract*—Video streaming in wireless multihop networks is a challenge due to different capabilities of end-user devices and changing network conditions. This challenge is addressed at the application layer with adaptive video streaming schemes like dynamic adaptive streaming over HTTP (DASH), which is widely applied by content providers. DASH copes with diverse end-user device capabilities by storing several representations of the same video such that DASH can offer a video in multiple qualities to users. Nevertheless, adjustments in DASH are solely taking place at the application layer. Especially in wireless multihop networks, adaptions on the lower layers are of particular importance. Therefore, we propose a novel application-aware cross-layer framework which adapts network support structures at the network layer, performs resource allocation at the medium access layer, switches between communication types at the physical layer and takes into account the properties and requirements of DASH at the application layer. Furthermore, we present a unified graph model, which takes into account the application layer, the network layer, the medium access layer and the physical layer jointly. We formulate a binary linear problem which chooses the optimal video representation for each user and finds the best combination of mechanisms on the lower three layers to optimally distribute the video content through the wireless multihop network. We show that our application-aware cross-layer framework which utilizes transitions leads to gains between 15-83 % compared to conventional approaches that do not switch between different mechanisms.**

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 device capability, e.g., screen resolution and processing power, and device connectivity, e.g., channel quality [2]. The challenge in WMN is the utilization of the resources, while disseminating the video content to the users. The resource utilization depends on the device connectivity, since the distribution of the video content is adapted according to the weakest user. At the application layer (APP), adaptive video streaming technologies have been proposed in order to address this issue. In the last decade, two methodologies for adaptive video streaming systems were proposed: one is utilizing scalable video coding (SVC) which in this paper is termed SVC-based video-streaming and the other is dynamic adaptive streaming over HTTP, which throughout this paper is assumed to apply non-scalable video coding and abbreviated DASH. In SVC, a video is divided into one base layer and several enhancement layers. The enhancement layers are built upon the base layer, thus all layers have to be received in order to recover the video in the highest quality [3].

Today, the standard gaining the current momentum of both academia and industry is DASH. In DASH, multiple representations of a video are stored, each corresponding to a certain quality [4]. From a network-centric view, both adaptive video streaming approaches introduce multiple data streams, but SVC-based video-streaming and DASH utilize the network resources differently.

SVC-based video-streaming introduces a data stream for every video layer, where a video layer with a higher order is distributed when all layers with a lower order can be received. Thus, multiple dependent data streams flow through the network to the users, when SVC-based video-streaming is utilized. In DASH, each user aims to obtain the video in the best quality with respect to his device capabilities. This may result in multiple independent data streams, where users are receiving different video representations of the same content. The multiple independent data streams introduced with DASH lead to new challenges with regard to the utilization of network resources.

Since adaptive video streaming cannot properly address the variations in a WMN like changing network density, available resources and interference related to the lower layers, a cross-layer approach combining adaptive video streaming and lower layer transitions is needed. A transition is the switching between two equivalent mechanisms, e.g. switching between unicast (UC) and broadcast (BC), or two equivalent network support structures, e.g. tree structure and butterfly structure, as illustrated in Fig. 1.

There are several related works which combine adaptive video streaming with a cross-layer approach. First, we discuss the literature combining SVC with a cross-layer approach. In [5], [6], [7], SVC at the APP and network coding [8] at the network layer (NET) are utilized to maximize the performance in terms of rate or video quality. Nevertheless, APP and NET cannot adapt to variations at the medium access layer (MAC) and at the physical layer (PHY). Thus, they cannot adapt to changes of the available resources and channel conditions. In [9] and [10], a resource allocation problem at the MAC is formulated

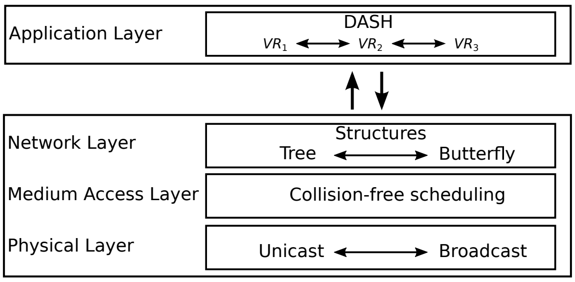


Figure 1: Application-aware cross-layer framework.

which maps SVC video layer rates to PHY rates such that every user can receive at least the base layer, while the remaining resources are allocated to the enhancement layers. Nevertheless, the presented results are limited to the one hop case and different communication types, e.g., BC and UC, are not taken into account. In [11], the advantage of transitions is demonstrated, where the proposed cross-layer framework switches between different mechanisms on the lower layers and takes into account the properties and requirements of SVC. In [12], the authors are combining DASH at the APP with cross-layer information at the MAC and the PHY. They presented results for cellular networks, but are not considering the multihop case and are not taking into account the switching between different communication types, e.g., BC and UC.

In this paper, we propose an application-aware cross-layer framework which utilizes the lower three layers and takes into account DASH at the APP, which to the best of our knowledge was not done before for WMNs. We model the different representations of a video as independent data streams. First, we show how the concept of DASH at the APP can be integrated into a unified graph altogether with the lower layers NET, MAC and PHY. We utilize the concept of transitions in the unified graph, where the framework switches between different video representations, between network support structures, e.g., tree and butterfly, and between communication types, e.g., UC and BC, simultaneously. Second, we propose a new heuristic to map the data rate requirements of the video representations at the APP to the available PHY rates. Thirdly, we formulate an optimization problem which allocates the optimal video representation to each destination.

The remainder of the paper is organized as follows. In Section II, the unified graph model of the application-aware cross-layer framework is presented. In Section III, a rate mapping heuristic and a sum rate optimization problem are developed. The simulation results are discussed in Section IV, where the proposed application-aware cross-layer framework is evaluated in terms of sum rate. Also, a comparison between the application-aware cross-layer framework against conventional schemes without the possibility to perform transitions is shown. The paper is concluded in Section V.

II. System Model

In this section, a unified graph is shown, which models DASH at the APP and the different mechanisms at the lower layers together. First the NET is discussed, which usually is modeled in a graph, where different network support structures

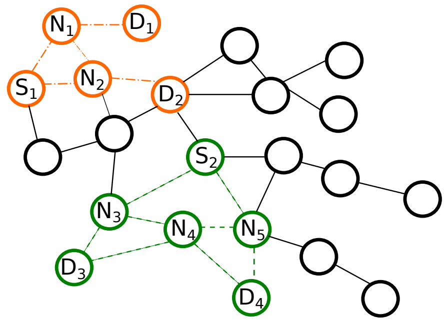


Figure 2: Network support structures in a random network: source node S_1 in the left utilizes the tree structure and source node S_2 in the middle utilizes the butterfly structure.

are utilized. Next, the different communication types UC and BC at the PHY are shown, which are integrated into the graph through the concept of virtualization [13]. In order to avoid collisions between transmitting nodes, the resulting graph containing NET and PHY mechanisms has to be split into subgraphs. Therefore, a collision-free scheduler is presented at the MAC. Finally, the integration of DASH into the graph is explained, which is achieved by modeling each video representation as an independent virtual source.

Throughout the paper, all nodes in the WMN operate in half-duplex mode and are equipped with a single omnidirectional antenna each. At the lower layers, the forwarded video is referred to as message.

A. NET Mechanisms: Network Support Structures

A WMN is modeled 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 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. The link between two nodes is denoted as a directed edge $e = (i, j)$, where i is the transmitting node and j is the receiving node.

In a WMN, it is beneficial to utilize different network support structures at the NET, in order to adapt to changes in the network topology. A network support structure is a set of nodes which cooperate in order to deliver the messages to the destination. Two network support structures are considered: the butterfly and the tree. In Fig. 2, an example of the butterfly structure and the tree structure is illustrated, where $S1$ is utilizing a tree structure and S_2 is utilizing a butterfly structure. The main difference between the two network support structures is that in the butterfly structure, network coding [8] can be utilized. In Fig. 2, using network coding, the butterfly structure combines two incoming messages from relays N_3 and N_5 at relay N_4 into one outgoing message. Another difference regarding butterfly and tree is the number of relays involved in the message forwarding. In comparison, a tree requires a lower number of relays in the forwarding than the butterfly structure, since in the butterfly, messages flow over multiple paths through different relays. However, the tradeoff between the two network support structures does not only depend on the current network topology, but also on the available PHY mechanisms and the scheduling at the MAC.

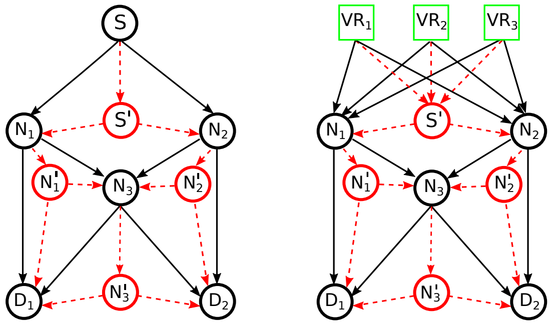
B. PHY Mechanisms: Communication Types

At the PHY, two communication types are considered to forward a message, BC and UC. In BC, a message is forwarded simultaneously to all neighbors of a node, where the rate is adjusted with respect to the weakest neighbor. In UC, a node forwards a message such that only one node is the intended receiver by adapting the rate to the intended receiver. Hence, a node can forward a message to a specific neighbor by utilizing UC, but it requires multiple time resources for multiple receivers. On the other hand, a node requires only one time resource to forward to all its neighbors with BC, but the resulting transmission rate is dominated by the weakest link. Therefore, a node needs to select the communication type that maximizes the overall rate.

In order to optimally decide which communication type to choose, the different communication types need to be included in the unified graph. The differentiation between UC and BC can be done by applying virtualization, cf. [11], [14], [13]. As an example, virtualization is applied in the graph in Fig. 3 (a). 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 in Fig. 3 (a), 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 capacities 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 a graph where the black solid edges in Fig. 3 (a) represent the UC communication links and the red dashed edges represent the BC communication links.

C. MAC: 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. Therefore, a collision-free scheduler is proposed, which splits the network graph into P collision-free subgraphs. Each subgraph $G_p \subseteq G$ contains a subset of vertices $V_p \subseteq V$ and edges $E_p \subseteq E$, which do not collide with each other. The scheduler is determining a finite amount of collision-free subgraphs, since finding all possible combinations is complex. The scheduling process is done as follows. First, the scheduler selects the node with the highest number of outgoing edges and schedules the node in subgraph G_p. In the second step, the scheduler checks in an iterative manner all remaining nodes by probing if collisions with nodes in G_p occur. Nodes are added into G_p, if they are not in conflict. This is repeated until no other node can be added to the p-th subgraph without introducing a collision. For the next subgraph G_{p+1}, the node with the highest number of outgoing edges which was not scheduled before is chosen. The above steps are repeated until every edge of each node is scheduled at least once. The obtained collision-free subgraphs are the basis for choosing the nodes and edges for the message forwarding. This is done by allocating resources to selected subgraphs, which is explained in Section III-B.



(a) (b)

Figure 3: (a) Node virtualization: extending network graph with virtual nodes and edges and (b) Source virtualization: node S is replaced by the three representations of the video as virtual sources.

D. APP: Modeling DASH Representations as Virtual Sources

In order to maximize the resource utilization on the lower layers, it is necessary to model the properties and requirements of DASH. Hence, the DASH representations are modeled as virtual sources to model the selection of different representations. This allows to model the adaption of DASH as a source selection problem at the lower layers. In DASH, a video is encoded multiple times and stored as K different video representations $VR_k \in \{VR_1, VR_2, \dots, VR_K\}$ at the source node. Thus, $VR_1, \dots, VR_K$ contain the same content, but each VR_k provides the content in a different quality, and hence, has a different data rate requirement. This allows users to choose the most suitable video representation with respect to their own capabilities. Consequently, a user can only request one video representation at a time. Each VR_k has a data rate requirement $B_k \in \{B_1, B_2, \dots, B_K\}$. Since each user can request a representation, each video representation is modeled as an independent source. The physical source is replaced by K virtual sources which represent the K video representations. As an example, the source node S of the graph in Fig. 3 (a) is replaced, where $K = 3$, hence resulting in the virtual sources VR_1, VR_2 and VR_3. S is replaced with VR_1, VR_2 and VR_3 in G as shown in Fig. 3 (b). As shown in Fig. 3 (b), VR_1, VR_2 and VR_3 are connected to the nodes through the UC and BC links of the original source which is indicated through the black solid and red dashed edges, respectively.

Now, the unified graph considers the APP, the NET, the MAC and the PHY jointly. Further different mechanisms and network support structures are modeled, which enables the utilization of transitions.

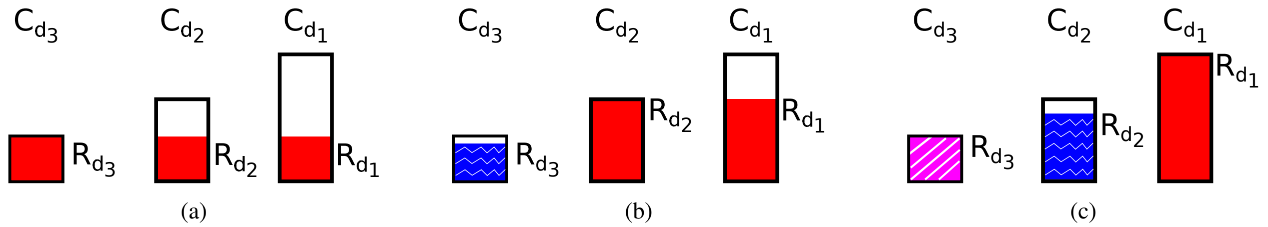


Figure 4: Toy example of the mapping heuristic (a) first step: container C_{d_3} is filled with VR_3 as well as C_{d_2} and C_{d_1} (b) second step: container C_{d_2} is filled with VR_3 as well as C_{d_1} while C_{d_3} is filled with VR_2 (c) third step: container C_{d_1} is filled with VR_3, C_{d_2} is filled with VR_2 and C_{d_3} is filled with VR_1, which is the optimal mapping solution.

III. Application-Aware Cross-Layer Framework for Wireless Multihop Networks

In the previous section, DASH representations are modeled as virtual sources in the application-aware cross-layer framework. The next step is to formulate the requirement of DASH, that a destination can only receive one video representation VR_k at a time. Therefore, a multi-source sum rate optimization problem for DASH is presented, which is a binary linear problem (BLP). In the BLP the above mentioned requirement is included as a constraint, where a destination can only be served by one virtual source at a time. However, the BLP can only be solved optimally under the condition that the data rate requirement of all video representations is mapped to PHY rate requirements. Thus, a new mapping heuristic for DASH is developed, which determines the PHY rates for every VR_k. First, the new mapping heuristic for DASH is discussed in Section III-A. With the obtained mapping solution, the BLP is solved, which is presented in Section III-B.

A. Video Representation Data Rate to PHY Rate Mapping

The aim of the heuristic is to map the required data rates of every video representation at the APP to PHY rates. This mapping problem is difficult to solve optimally, since the data rate requirements have fixed ratios and the PHY rates are continuous. Therefore, a heuristic is proposed, which takes the view of one representative subcarrier in a multi-carrier system. Based on the relative data rate requirement for each video representation VR_k, the heuristic is performing a mapping that translates the relative data rate B_k of a video representation VR_k at the APP to the corresponding PHY rate r_k.

By illustrating the end to end capacities as containers, the heuristic aims to fill the containers as much as possible with the data rate of the video, which can be imagined as being a liquid. The heuristic searches in a greedy fashion for a solution. A mapping solution contains one container filled completely with a data rate, based on this allocation the remaining data rates are calculated and the remaining containers are filled accordingly. Thus containers can only be filled to a level, which fulfill the fixed ratios between the data rates and which do not overflow the containers. The fixed ratios lead to empty spaces in the containers. Hence, the heuristic tries out all combinations of different containers filled with different data rates. In order to find a solution, which minimizes the overall empty space in the containers.

The rate mapping problem is illustrated by the following three exemplary cases, see Fig. 4 (a)-(c). This toy example consists of three destinations d_1, d_2 and d_3, three video representations VR_1, VR_2 and VR_3 with data rate requirements B_1, B_2 and B_3 at the APP, where $B_1 = \frac{1}{3}B_3$ and $B_2 = \frac{2}{3}B_3$. Further, it is assumed that the end to end capacity of each destination at the PHY is known [14], where C_{d_1} is the capacity of d_1, C_{d_2} is the capacity of d_2 and C_{d_3} is the capacity of d_3. The heuristic starts by filling the smallest container C_{d_3} completely with B_3. Also C_{d_1} and C_{d_2} are filled with B_3, resulting in equal levels at the containers $R_{d_3} = C_{d_3}$, $R_{d_2} = R_{d_3}$ and $R_{d_1} = R_{d_3}$, as shown in Fig. 4 (a). Next, the second smallest container C_{d_2} is filled completely with B_3, leading to C_{d_1} being filled with B_3. Since the container C_{d_3} is smaller than C_{d_2} it cannot fit B_3 and hence is filled with $B_2 = \frac{2}{3}B_3$, as illustrated in Fig. 4 (b). In Fig. 4 (c), the heuristic continues searching and is filling C_{d_1} with B_3. Both containers C_{d_2} and C_{d_3} cannot be filled with B_3, since they are smaller than C_{d_1}. Hence, C_{d_2} is filled with $B_2 = \frac{2}{3}B_3$ and since C_{d_3} is smaller it is filled with $B_1 = \frac{1}{3}B_3$.

Out of three exemplary cases, the mapping which minimizes the overall empty space of the containers is shown in Fig. 4 (c). Thus, the heuristic sets the PHY rates to $r_3 = C_{d_1}$, $r_2 = \frac{2}{3}r_3$ and $r_1 = \frac{1}{3}r_3$. In the next section, the obtained PHY rate requirement r_k for each video representation VR_k is injected into the multi-source sum rate optimization problem to determine the maximum sum rate.

B. Multi-Source Sum Rate Optimization

As mentioned earlier, DASH representations are modeled as independent virtual sources. The independence between the virtual sources leads to competition for network resources, especially when destinations are served with different video representations. Nevertheless, the objective is to maximize the sum rate over all virtual sources 1 to K and destinations 1 to D. Thus, the utility function can be expressed as

$$\max \sum_{k}^{K} \sum_{d}^{D} r_{k,d} \tag{1}$$

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. The flow from k to d over the link from node i to node j in the p-th subgraph is defined 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

$$\sum_{p=1}^{P}\left(\sum_{j:(i,j)\in E_p} f_{i,j}^{(p)}(k,d) - \sum_{j:(j,i)\in E_p} f_{j,i}^{(p)}(k,d)\right) = \sigma_i, \quad (2)$$
$$\forall i \in V, k = \{1,...,K\}, d = \{1,...,D\}$$

where σ_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 zero otherwise. Furthermore, each flow is upper bounded by a capacity constraint. The capacity in subgraph p depends on the link capacity $c_{i,j}$ between nodes i and j and the duration the link is utilized in the p-th subgraph, which is determined by the timeshare factor τ_p. If a link is part of sub-graph G_p the indicator function $\mathbf{I}_{E_p}(i,j)$ is one, else $\mathbf{I}_{E_p}(i,j)$ is zero. This means that the link is not active in G_p. The indicator function is written as

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

Furthermore, the flow through a link is bounded by the capacity constraint, which is expressed as

$$0 \le \sum_{k}^{K} f_{i,j}^{(p)}(k,d) \le \tau_p \cdot c_{ij} \cdot \mathbf{I}_{E_p}(i,j), \quad (4)$$
$$\forall (i,j) \in E_p, p = \{1,...,P\}, d = \{1,...,D\}.$$

The DASH constraint is formulated as follows

$$\sum_{k=1}^{K} x_{k,d} \le 1 \qquad \forall d = \{1,...,D\}, \quad (5)$$

where $x_{k,d}$ is binary and equal to one if video representation k is received by destination d and zero otherwise. Eq. (5) reflects the requirement that a user can receive only one video representation at a time, by limiting the sum over $x_{k,d}$ to one. Furthermore, if $x_{k,d}$ is one, then the rate $r_{k,d}$ is equal to the previously obtained physical rate r_k. This is written as

$$r_{k,d} = x_{k,d} \cdot r_k, \quad \forall k = \{1,...,K\}, d = \{1,...,D\}. \quad (6)$$

Finally, the timeshares are normalized and bounded as

$$\sum_{p=1}^{P} \tau_p = 1 \quad (7)$$
$$0 \le \tau_p \le 1, \qquad \forall p = \{1,...,P\}. \quad (8)$$

The multi-source optimization problem expressed in Eq. (1) - (8) is a BLP. By solving the BLP, the maximum sum rate in the system, a binary matrix containing $x_{k,d}$ and the timeshare factor τ_p for each sub-graph are obtained.

IV. Simulation Results

In this section, the performance of the application-aware cross-layer framework in a scenario with one video source is investigated. It is assumed that three different video representations are available and that they have relative data rate requirements of $B_1 = \frac{1}{3}$, $B_2 = \frac{2}{3}$ and $B_3 = 1$. The simulation is performed for different network sizes, details are given in Table I. The evaluation is done over 100 snapshots of random networks, where the nodes are uniformly distributed in an area with a map size of 15 m by 5 m. The performance of the proposed application-aware cross-layer framework (CrossDASH) is evaluated in terms of sum rate. CrossDASH is compared against three reference schemes. The upper bound is allocating the resources such that the maximum sum rate is achieved, without considering DASH. In more detail, the upper bound does not uphold the constraints in Eq. (5) and Eq. (6) of Section III. In order to show the advantage of transitions, two additional schemes are considered which cannot perform transitions. The first one is using the butterfly structure at the NET and BC at the PHY, which is abbreviated with BBC. The second scheme is utilizing the tree structure at the NET and UC at the PHY, which is abbreviated with TUC.

In Fig. 5, the cumulative distribution function (cdf) of the sum rate achieved in the system is shown for the case of three users and a network size of 15 nodes. First, the outage capacity at 10 % is analyzed, where CrossDASH provides an outage capacity of 0.15 bits/s/Hz. This is a gain of 66 % compared to BBC and a gain of 50 % compared to TUC. The median of sum rate for CrossDASH is 0.27 bits/s/Hz, which is a gain of 50 % compared to BBC and a gain of 18 % compared to TUC. The cdf shows that the sum rate achieved with CrossDASH aligns with the upper bound, but also TUC performs very close to the upper bound for high sum rate values. CrossDASH achieves a maximum sum rate of 0.94 bits/s/Hz, as well as the upper bound and TUC. Furthermore, CrossDASH achieves a gain of 38 % compared to BBC in terms of maximum sum rate. The proposed scheme CrossDASH outperforms BBC in terms of maximum sum rate, since BBC can only utilize BC at the PHY which can introduces more conflicts leading to an inefficient scheduling at the MAC. Also, CrossDASH outperforms TUC and BBC in terms of outage capacity, because CrossDASH performs transitions at the lower layers. In addition, CrossDASH performs very close to the upper bound, due to fact that the mapping is performing very well.

In Fig. 6, a comparison of the average sum rate is shown for CrossDASH, TUC and BBC for the case of 3 users, for an increasing number of nodes in the network. As the number of nodes increases, the average sum rate in bits/s/Hz increases for CrossDASH and TUC, while for BBC the average sum rate declines. Fig. 6, as the size of the network increases the gain of CrossDASH increases from 33 % for 10 nodes, to 68 % for 20 nodes and to 83 % for 30 nodes compared to BBC in terms of average sum rate. This is due to the fact that BBC

Transmit power P_T	0 dBm
Noise power P_N	-30 dBm
Path loss exponent α	4
Average channel gain $\mathrm{E}\{\lvert h\rvert^2\}$	1
Distance $d_{i,j}$ between node i and j	1 m - 5 m
Number of nodes	10-30
Number of destinations	3

Table I: Simulation parameters

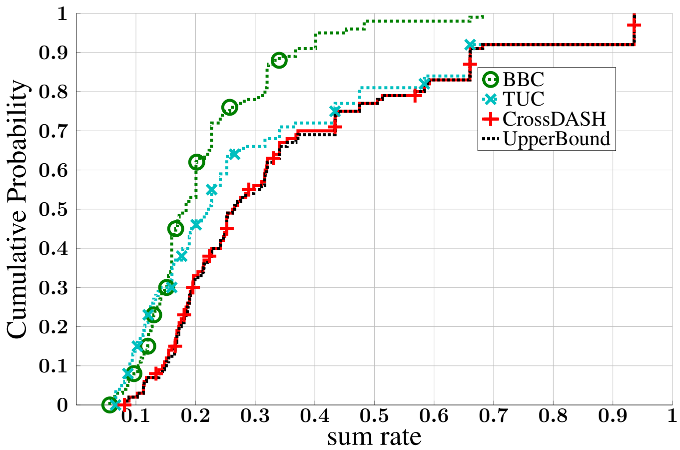


Figure 5: Empirical cumulative distribution function of the sum rate for BBC, TUC, CrossDASH and upper bound.

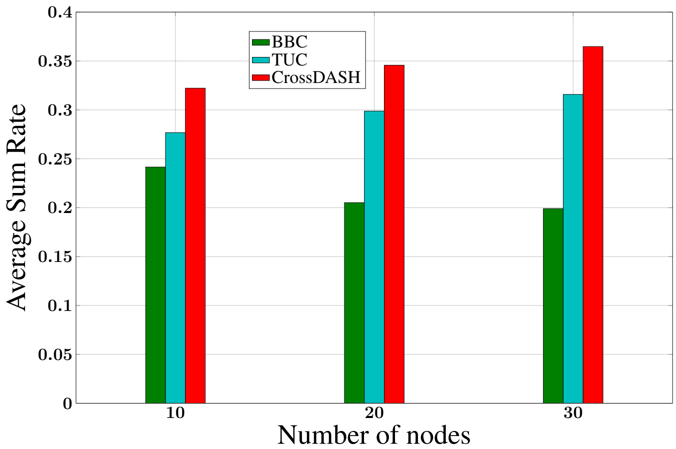


Figure 6: Average sum rate of BBC, TUC and CrossDASH over changing network size.

can only utilize BC, which leads to low rates in networks with high density of nodes. Further, CrossDASH achieves a steady gain of 15% compared to TUC over all network sizes, since CrossDASH can perform transitions.

V. Conclusion

In this paper, the flexibility of DASH at the APP was combined with network support structures at the NET and communication types at the PHY into a unified graph. The proposed application-aware cross-layer framework can adapt the APP, NET, MAC and PHY jointly to provide the optimal combination of video representation and mechanisms. In addition, a new mapping heuristic for DASH was presented which is solely based on the relative data rate requirements of the video representations. The obtained PHY rates for the video representations were fed into the formulated multi-source optimization problem, which can handle DASH by treating the video representations as independent virtual sources. In conclusion, the proposed scheme CrossDASH performs very close to the upper bound, outperforms TUC and BBC in terms of outage capacity and outperforms BBC in terms of maximum sum rate. CrossDASH achieves gains of up to 66 % in terms of outage capacity and significant gains of up to 38 % in terms of of maximum achievable sum rate, since transitions at the lower layers are utilized. Furthermore, CrossDASH can handle networks with high density and achieves significant gains up to 83% with regard to the sum rate compared to schemes which cannot perform transitions.

VI. Acknowledgment

This work has been funded by the German Research Foundation (DFG) as part of project C01 and C03 within the Collaborative Research Center (CRC) 1053 – MAKI. The authors would like to thank Sabrina Müller (B03), Alexander Frömmgen (A02) and Denny Stohr (B01) for their instructive inputs on optimization and DASH.

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 heterogenous access networks," in *Proc. ACM 15th International Conference on Multimedia*. ACM, 2007, pp. 37–46.

[3] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, 2007, pp. 1103–1120.

[4] J. Nightingale, Q. Wang, C. Grecos, and S. Goma, "Video adaptation for consumer devices: opportunities and challenges offered by new standards," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 157–163, December 2014.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] M. Fasil, H. Al-Shatri, S. Wilk, and A. Klein, "Application-Aware Cross-Layer framework: Video content distribution in wireless multihop networks," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Hong Kong, P.R. China, Aug. 2015, pp. 1088–1093.

[12] M. Zhao, X. Gong, J. Liang, W. Wang, X. Que, and S. Cheng, "Scheduling and resource allocation for wireless dynamic adaptive streaming of scalable videos over http," in *IEEE International Conference on Communications (ICC), 2014*, June 2014, pp. 1681–1686.

[13] 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.

[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.