Optimizing Video Quality in Dense Small-Cell Wireless Networks with Packet Overhearing

Dimitrios Kosmanos*, Antonios Argyriou*, and Leandros Tassiulas†
*Department of Electrical and Computer Engineering, University of Thessaly, Volos, 38221, Greece.
†Yale Institute for Network Science (YINS), Yale University, New Haven, 06511 CT, USA.

***Abstract*—The heterogeneous wireless network (HetNet) paradigm is based on the deployment of low power small cell base stations (SCBS) close to the user in parallel with a macrocell BS (MBS) that provides umbrella coverage. However, since these SCBSs are expected to be deployed in significant numbers, their increased density creates new problems but also new optimization opportunities. In this paper we design a video streaming system for HetNet configurations that allow the SCBSs to opportunistically cooperate, while their backhaul link can be either wireless or wired. The first component of our system is an algorithm that is executed at each SCBS and is responsible for opportunistically overhearing packet transmissions from the MBS and other SCBSs and forwarding them to the users. The second component of our system is a rate allocation algorithm that is executed at the video streaming server. Our system offers different performance benefits depending on the HetNet backhaul configuration. Our performance evaluation with high quality 4K videos, indicates that in the wireline backhaul case video streaming experiences lower delay, while in the wireless backhaul case our system improves the capacity that is available for video communication.**

***Index Terms*—Heterogeneous cellular networks, HetNet, small cells, video streaming, video distribution, rate allocation, opportunistic communication, cooperative communication, 4K video.**

I. Introduction

Video streaming in wireless cellular and local area networks is re-surfacing as a significant engineering problem because of the explosive growth of high quality video on demand services [1]. The most important challenge is the provisioning of a high wireless communication data rate that is required for high quality video streaming. Even after compression, video is bandwidth-hungry and delay-sensitive. Hence, there is no fundamental way to ensure smooth playback during the delivery phase of the video through streaming, besides providing more bandwidth. To ameliorate the situation in cellular networks, low power small cell base stations (SCBS) are deployed closer to the user. The benefits of this approach are twofold: First, spatial re-use is increased because the SCBSs can share the same time/frequency resources due to the low transmission power. Second, higher data rates for each individual user can be achieved since shorter physical distances lead to high spectral efficiency. This heterogeneous cellular network (HetNet) includes several SCBS (picocells and femtocells) with overlapping coverage, while the typical macro base station (MBS) provides umbrella coverage in the complete macrocell [2]. A critical observation is that as more users join the system, an easy way for the mobile operators to keep increasing the performance of HetNets is by deploying more SCBSs.

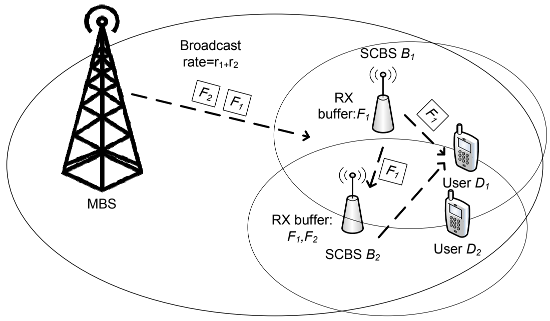


Fig. 1. Multi-flow (F_1 and F_2) and multi-user (D_1 and D_2) video streaming in a HetNet with a wireless backhaul. Each user receives the requested video flow through the SCBS that is associated. In our system model the SCBSs are allowed to cooperate opportunistically by overhearing packets from both video flows. Only the streaming of video flow F_1 to D_1 is depicted to avoid clogging the figure.

Consequently, the high density of SCBSs means that several of them may be reachable by a user (Fig. 1) besides the single SCBS that a user is typically associated. The question is whether the dense deployment of these SCBSs can offer additional benefits for video streaming, besides the increase in spatial-reuse and spectral efficiency. In this paragraph with the help of simple examples we illustrate further potential optimization opportunities. Consider first the scenario of a wireless backhaul connection between the MBS and the SCBSs as illustrated in Fig. 1. The MBS broadcasts packets F_1 and F_2 that belong to two different video flows and these transmissions can be overheard from the two SCBSs within the range of user D_1. The immediate benefit is that because of channel diversity, a packet has higher probability to be received at least in one SCBS. Assume now that SCBS B_1 has been granted access to the channel, and transmits packet F_1 to the destination. Due to the proximity of the two SCBSs, SCBS B_2 cannot simultaneously transmit with B_1. However, this does not prevent it from *overhearing* the transmission of packet F_1 from its neighbor and remove it from its local buffer when user D_1 acknowledges it. The same situation will occur if more SCBSs are deployed close to B_1, i.e. they can also overhear. When the SCBSs connect to the core network through a wireline backhaul, the communication model is similar only in this case B_2 can overhear packet transmissions for video flow F_1 only from B_1 that user D_1 is associated. In the two previous scenarios we see that the two fundamental communication topologies that emerge, and are illustrated in Fig. 2, are effectively *cooperative*.

The previous discussion indicates that the adoption of a cooperative communication model in dense HetNets can lead to potential benefits for wireless video streaming. Nevertheless, network cooperation for wireless video distribution has been investigated considerably in the literature [3], [4], [5], [6], [7]. More specifically, physical layer cooperative communications for wireless video has been investigated in [3], [4], [5]. However, as it is evident from our discussion in the last paragraph, in this work we are interested in cooperative techniques at the higher layers of the protocol stack, also known as packet-level cooperation. In this category, one of the most well-known techniques is wireless network coding (NC). The primary benefit of NC is higher throughput, but it was also shown that it can improve the quality of wireless video multicasting applications [6], [7]. In the previous two works the authors employ linear network coding of video packets before they are broadcasted to a multicast group of clients. However, with the combination of NC and video, there is the requirement that coded packets are acknowledged by all the participating intermediate nodes. This is necessary in order to improve the coding decisions of the time-sensitive video packets at the source. Furthermore, the previous works target generic ad-hoc network topologies that are different from HetNets. For the particular case of HetNets it has been recently shown that wireless video streaming can be improved through caching [8], [9]. The femtocaching idea that was proposed in [8], suggests that the small cells should cache video files in order to serve them repeatedly to several users. More recently it has also been shown that there is a tradeoff between caching and resource use decisions when the backhaul cost is considered [9]. Nevertheless, caching is an offline optimization approach that can lead to delay and power/cost minimization, aspects that are complimentary to our work that focuses on video quality.

From our brief overview of the related work, we notice that none of the previous works considered the problem of multi-flow multi-user video streaming in dense HetNets where opportunistic communication is allowed. *Therefore, in this paper we design a video streaming system for a category of HetNets that allow the SCBSs to opportunistically cooperate while the backhaul can be either wireless or wired*. For this class of HetNets first we propose a lightweight cooperative protocol that operates between the SCBSs. The protocol allows the SCBSs to make decisions for packet forwarding or discarding in a completely distributed fashion by only collecting information passively. In our previous work [10], we have also considered the impact of opportunistic packet overhearing in wireless networks in order to optimize randomized network coding decisions. The claim in [10] was that there is a throughput benefit in a wireless network if nodes have a more accurate knowledge of the information available in their immediate neighborhood. In this work we are interested to improve video quality. In the second part of this work, and after we design the HetNet with opportunistic communication, we design the streaming system that allocates optimally the end-to-end communication rate across multiple users. Our second contribution is that we employ network utility maximization (NUM) for improving the quality of video streaming in a dense opportunistically cooperating HetNet.

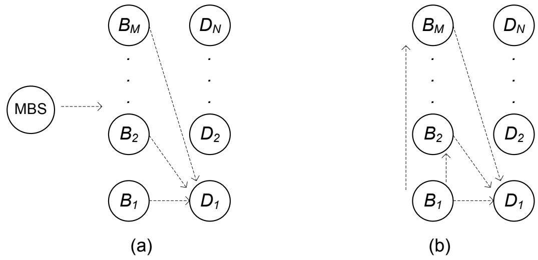


Fig. 2. The fundamental communication model in dense HetNets with wireless backhauling is illustrated in (a), and wireline backhaul in (b).

The benefits of our complete system design are: 1) Opportunistic communication in dense HetNet with our overhearing algorithm leads to higher communication rate from the source to the users. 2) The rate allocation algorithm of the streaming system is decoupled from the opportunistically cooperating HetNet. 3) SCBSs only overhear packets without exchanging any type of out-of-band information. 4) The users do not overhear packets and so power consumption is minimized.

II. System Model and Assumptions

Heterogeneous Small-Cell Network Model: In this paper we consider the unicast streaming of a set $\mathcal{F} = \{F_1, ..., F_N\}$ of N pre-compressed and packetized video flows from a single video streaming server located at the MBS, to a set $\mathcal{D} = \{D_1, ..., D_N\}$ of N users that each one is interested in a specific video flow. The set of users $\mathcal{D}$ is reachable by a set $\mathcal{B} = \{B_1, ..., B_M\}$ of M SCBSs that are not the consumers of the video but their task is to aid by forwarding traffic to the users as seen in the macrocell system model in Fig. 1, and the communication model in Fig. 2. Hence, we focus our attention in a single neighborhood of the HetNet that consists of these two sets of nodes, $\mathcal{D}$ and $\mathcal{B}$. In the first flavor of our network model the backhaul link from the MBS to the SCBSs is a wireless link [11] and the N videos are stored at the MBS. We also consider a wireline backhaul model. In this case wireless transmissions occur only from the SCBSs that may also have cached locally the video [8], [9]. Thus, SCBSs can overhear only neighboring SCBSs.

Video Packet Transmission: The source transmits video flow n at rate r_n bits/sec by allocating optimally the rate of the end-to-end channel. To accomplish that, the average throughput measured at each user is periodically collected at the source (e.g. through RTP messages). Because rate allocation is executed at the source based on information for the complete end-to-end channel, all the packets that are transmitted from the source should reach the user and not be dropped. Therefore, the SCBSs know that they must transmit all the packets in a FIFO order.

Channel Model and PHY Modulation: Network nodes inside the cell access the channel independently through a mechanism that ensures orthogonal access. In particular here we assume the most general case of a fully distributed scheme like CSMA/CA. All the channels are considered to be narrowband block-fading Rayleigh. The channel coefficients are quasi-stationary, that is they remain constant for the coherence time of the channel (slow fading). Additive white Gaussian noise (AWGN) with zero mean and unit variance is assumed at the SCBSs and the users. At the PHY we assume that a single-carrier (SC) Phase Shift Keying (PSK) modulation is used. The transmitters employ ARQ that is typical in cellular and wireless LAN standards. Also we only assume channel state information at the receiver (CSIR).

III. SCBS Overhearing Protocol

In this paper we fully exploit the broadcast property of the wireless channel around the SCBSs. The reason is that many SCBS may receive the same packet. At the core of the proposed system is the overhearing algorithm. Depending on the backhaul link setup, and because of the randomness of the wireless channel different broadcasted packets will be received at different SCBSs. When a SCBSs forwards a packet to the user, the remaining SCBSs also overhear this transmission and its acknowledgment.

The pseudo-code for the small cell video packet overhearing (VPO) algorithm is presented in Fig. 3. The notation we adopt follows the IEEE 802.11 frame format since this protocol is practically possible to be used for a testbed implementation. In our implementation the SCBSs employ a type of pseudo-broadcast by transmitting with unicast to their respective next hop and allowing overhearing. The SCBS at the application layer uses the process *tx_pkt_app*() in order to enqueue to the MAC layer packets that are overheard (buffer **OHR**), and also packets that are directly arriving at the SCBS from the wireless or wireline backhaul (buffer **DIR**). When the SCBS is granted access to the channel the MAC transmits without any further interaction with our algorithm the head-of-line (HOL) packet. An important part of the functionality occurs in the procedure *rcv_pkt_app*() that receives packets from the operating system (MAC and network layers). In this case we require that the MAC layer passes all the information from the MAC protocol data unit (MPDU) to the application.[1] The VPO algorithm checks if the MPDU corresponds to a data packet and in this case it places it in the **OHR** buffer. It also stores the MPDU sequence number in order to keep track the transmission of this specific MPDU (loss or success) as we describe next. When the MPDU that is delivered to the application is an ACK, then the algorithm checks if this MPDU is stored in the **OHR** buffer. If this is the case it removes it

```
tx_pkt_app()
if OHR!=NULL then
    p=OHR→ HOL
    tx_pkt_mac(p,p→ dst)
else
    p=DIR→ HOL
    tx_pkt_mac(p,p→ dst)
end if
rcv_pkt_app()
p=rcv_pkt_mac() //Waiting for pkt from MAC
if p→ MPDU→ type==DATA then
    if p→ MPDU→ dst==D then
        add(p,OHR,p→ MPDU→ SeqNo)
    end if
else if p→ MPDU→ type==ACK then
    if p→ MPDU → Addr1 ==D then
        remove(p → MPDU→ SeqNo,DIR) OR
        remove(p → MPDU→ SeqNo,OHR)
    end if
end if
```

Fig. 3. Pseudo-code for overhearing algorithm at a SCBS.

since the interested user has received it. By following this approach, the neighboring SCBSs know that in the case they also have the packet in their buffer, there is no need to transmit it when they obtain the channel. Thus, in case a packet transmission fails this is detected by the lack of an ACK and all the SCBSs retain the lost packet in the **OHR** buffer since any one of them may be transmitted in the next opportunity. Therefore, the overhearing algorithm allows the SCBSs to forward uniquely each packet to a user. In other words the algorithm ensures that the SCBSs know which SCBS transmitted a specific packet and what was the outcome of this transmission, but the precise packets available at each node are not known.

IV. Rate Allocation

In this section we answer the question of how the source calculates the optimal streaming rate for a number of N video flows targeting the set of users $\mathcal{D}$. We formulate our optimization problem as a NUM. Different utility functions can be employed. In our case, the utility function for a specific video flow, is defined as the reduction of the reconstruction distortion of the video flow n when r_n bits/sec are allocated to it:

$$U_n(r_n) = \sum_i \Delta D(i) \quad \text{with} \quad \sum_i \Delta R(i) \leq r_n. \qquad (1)$$

In the above i enumerates the packets of video flow n, and $\Delta R(i)$ is the size of packet i in bits. Also $\Delta D(i)$ is the value of the MSE distortion that includes both the distortion that is added when packet i is lost and also the packets that have a decoding dependency with i.[2] In this way the utility

[1]This is possible in Linux through the socket buffer (`sk_buff`) data structure.

[2]For example the ΔD for an I frame includes the ΔD of the P and B frames that depend on it.

formulation considers also the possible drift that might occur due to the loss of particular packets/video frames. Now, in order to compute the utility in (1) we previously label the media packets comprising the video in terms of importance using the procedure from [12]. Therefore, the index i in the summations in (1) enumerates the most important media packets in the presentation up to a data rate of r. In other words, $U_n(r_n)$ corresponds to the cumulative utility of the most important packets up to the rate point r_n.

Using the notation introduced previously we can write the optimization problem as

$$\max_{r_n} \sum_{n=1}^{N} U_n(r_n) \tag{2}$$

$$\text{s.t.} \quad \sum_{n=1}^{N} r_n \leq T,$$
$$r_n \geq 0, \quad n = 1, ..., N$$
$$U_n(r_n) \in \mathcal{U}_n, \quad n = 1, ..., N$$

In the above T is the measured end-to-end throughput. The last constraint ensures than the utility value for each flow belongs to a valid utility point that belongs in the discrete set $\mathcal{U}_n$. We proceed here to solve the optimization problem in (2). For this problem, we can apply Lagrange duality to the first constraint in (2) to produce the following partial Lagrangian

$$L = \sum_{n=1}^{N} U_n(r_n) - \lambda(\sum_{n=1}^{N} r_n - T), \tag{3}$$

where $\lambda > 0$ is the Lagrange multiplier. Similarly, r_n is current instantaneous rate allocation for flow n.

Now, (2) represents a concave optimization problem with linear constraints for the rate region. The Lagrange multiplier expresses the price of each selected rate allocation for flow n. It is known that if λ^* is the optimal solution for the dual problem, then the corresponding $r^*(\lambda^*)$ is the solution to the primal problem defined in (2). It can be shown that the following two equations represent a solution for the primal-dual optimization problems. First, the source computes the currently optimal rate allocation for flow n as

$$r_n(t) = \arg\max_{r_n} \Big\{ U_n(r_n(t)) - \lambda(t) r_n(t) \Big\}, \quad n = 1, ..., N. \tag{4}$$

Then, given $r_n(t)$ we employ a sub-gradient method [13] to update the value of $\lambda(t)$ as follows

$$\lambda(t+1) = \max\big\{0, \lambda(t) + \beta\Big(\sum_{n=1}^{N} r_n(t) - T\Big)\big\}. \tag{5}$$

In the above equation β is a small constant that is appropriately selected to ensure convergence. Sub-gradient adaptation methods such as (5) are typically used in discrete optimization problems involving Lagrange relaxation. Lastly, (4) and (5) are consecutively applied by the source until the algorithm converges.

A. *Stream Adaptation at the Source*

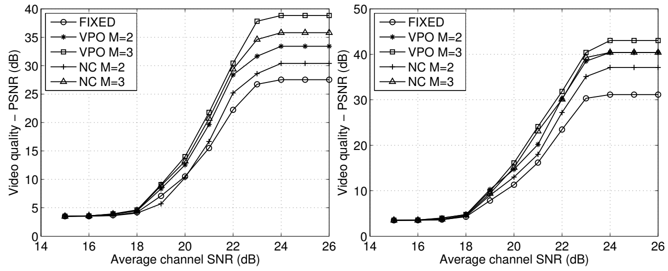
As we explained in Section II, in the proposed system each user periodically forwards the average throughput to the source. With this information, the source estimates the aggregate throughput, executes the rate allocation for the complete end-to-end system, and broadcasts the optimal packets according to the algorithms explained so far. Now, as shown in [12] the optimal rates r_n^* can be efficiently enforced using the rate-distortion (RD) characterization of the media packets comprising a flow. In particular, if $\Delta D(i)/\Delta R(i)$ is the utility gradient of packet i from a specific flow, in order to achieve the optimal rate point the source node should transmit the video packets if $\Delta D(i)/\Delta R(i) > \lambda^*$.

V. Performance Evaluation

In this section, we present a comprehensive evaluation of the proposed algorithms comprising our framework through simulations. The number of SCBSs M is kept small since the simulator operates at the PHY symbol level (not packet-level) requiring thus significant amount of execution time. Regarding the lower layer parameters we assume a channel bandwidth of W=20 MHz, while the fading model is frequency-flat Rayleigh and remains invariant per transmitted PHY frame. The maximum PHY communication rate was equal to IEEE 802.11a, i.e. 54 MBps. The average channel SNR depicted in the horizontal axis of all the figures was assumed to be the same for all the links but it varied independently during each channel realization. We also enabled the ARQ mechanism for the evaluation of the above systems. These ARQ feedback messages are assumed without error. Note that in all figures we present the utility divided by the number of required time slots for delivering a video flow. This means the impact of ARQ delay is also considered.

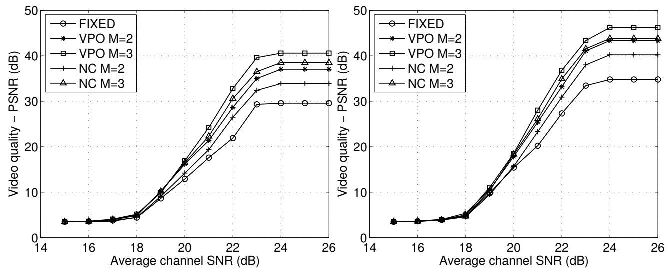
The rate allocation at the source was exercised for the duration of 10 GOPs. The video content used in the experiments consists of the 4K (4096 x 1744) sequence Tears of Steel that was compressed using the H.264 codec at an average bitrate of 8 Mbps, 24 fps, and an average quality of 46.5 dB [14]. The reason we selected this high quality video is that the simulated 54Mbps PHY allows for very fast data transmission. Each video frame corresponds to one slice and each slice was packetized to a single packet. For the experiments with two video flows different parts of the sequence were used. Also, the startup/playback delay of the video presentation at every node is denoted as d_s. In all the figures, the results correspond to the average peak signal-to-noise ratio (PSNR) enjoyed by all the destinations for the duration of 300 seconds. Finally we must mention that we did not use any form of error concealment in order to demonstrate clearly the impact of our specific metric/optimization scheme.

Simulation Results for Wireless Backhaul. Results for streaming one video flow to one user are shown for the NoOpt system that does not use NUM in Fig. 4(a), while results for the Opt system that uses the NUM framework are shown in Fig. 4(b). Generally, the increased number of SCBSs results in higher capacity and eventually higher video quality because



(a) NoOpt (b) Opt

Fig. 4. Average PNSR vs. the channel transmit SNR for system without ARQ. $d_s = 5$ seconds.



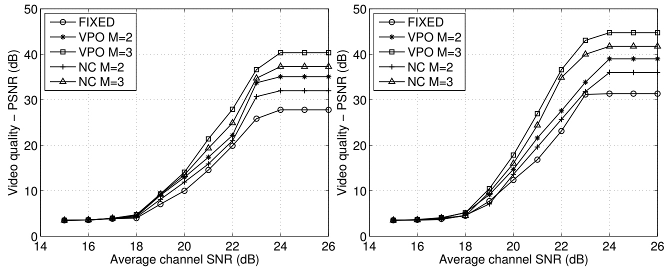
(a) NoOpt (b) Opt

Fig. 5. Average PNSR vs. the channel transmit SNR for system with ARQ. $d_s = 10$ seconds.

a single packet that is broadcasted from the MBS has higher probability to be received at a group of SCBSs instead of just one. This is true even for the NoOpt system that does not apply NUM in Fig. 4(a). However, more gains can be achieved when both components of our system are combined in Fig. 4(b). Another interesting behavior, that we can see in all the figures, is that as the channel SNR is increased, the quality for all schemes becomes a flat line. This behavior is because of the use of the highest possible PHY transmission rate [15]. We also evaluated a system where we enabled the ARQ mechanism that typically exists at the link layer (e.g. IEEE 802.11, LTE). In this case we also set d_s=10 seconds to accommodate for the impact of the ARQ mechanism. The related results can be seen in Fig. 5. We see now that the startup delay is increased, and the retransmissions are also increased, this has considerable impact on the quality of both the Opt and NoOpt systems.

Simulation results for the wireless backhaul case and two video flows transmitted to two users, can be seen in Fig. 6. We also configured a slightly higher startup delay due to bandwidth splitting across the two users. As it can be seen also in Fig. 6, the results have the same form as the results for one flow. As the channel quality is improved and the number of SCBSs is increased, the Opt system presents higher gains when compared to the NoOpt system.

Comparison with Network Coding. NC offers benefits for relatively high SNR. However, in this case the majority of wireless packets are received by all the SCBSs and so there



(a) NoOpt (b) Opt

Fig. 6. Average PNSR vs. the channel transmit SNR for two flows and a system without ARQ $d_s = 15$ seconds.

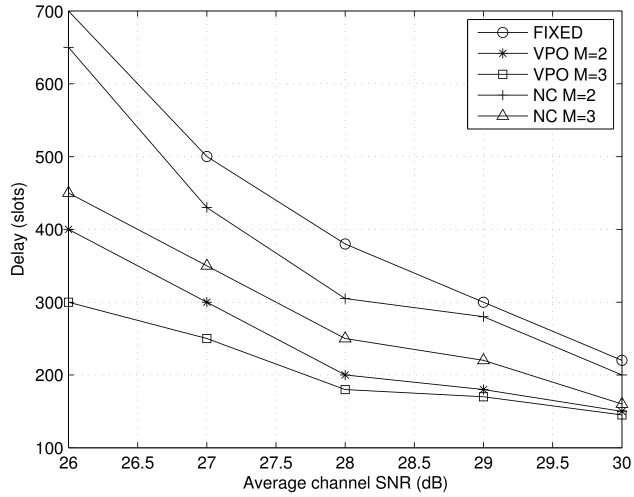
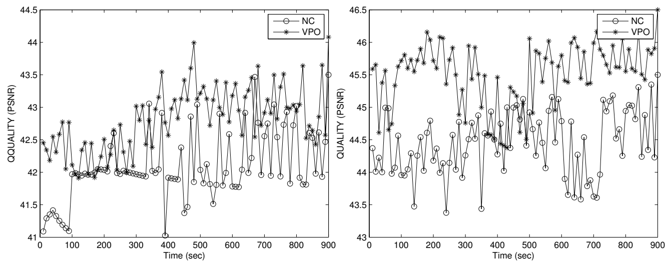


Fig. 7. Total delay vs. the channel transmit SNR for a system without optimization.

is no NC benefit when compared to our proposed scheme. The disadvantage of NC is because there is extra overhead in the transmission of buffer maps that consume channel bandwidth. On the other hand, in the low SNR regime the benefit of NC is similarly limited with VPO because the packet coding opportunities are less due to the high number of packet losses. This situation is true also for all the Opt systems as it can be seen for example in Fig. 6(b) while in this case the performance differences of VPO over NC are even bigger. Thus, the important conclusion in that overhearing implemented through a simple protocol like VPO is enough for making utility-optimal streaming decisions in this scenario where multiple SCBSs exist and the flows are unicast.

Delay Results with Wired Backhaul. Results for the delivery delay of a complete video file for the system without Opt and ARQ can be seen in Fig. 7. When the backhaul is wired, the capacity cannot be increased since it is limited by the point-to-point backhaul connection. Our measurements in this case show that the benefit of VPO with more than one SCBS is because of the lower packet error rate for individual packets and the reduced need for re-transmissions. When a



(a) M=2 (b) M=3

Fig. 8. Real-time PSNR results for the streaming of the two flows.

packet is lost from the primary SCBS this incurs an additional delay for its delivery. With VPO this delay is minimized because the lost packet from a SCBS is retransmitted from its neighbor and there it resides at the head of the queue. This is a key benefit of our proposed system in HetNets that deploy a wired backhaul.

Real Testbed Results. We implemented the basic concepts of this paper in a real testbed. We used the widely popular Cisco-Linksys WRT54GL Wireless 802.11g Access Point flashed with a custom OpenWRT firmware. As a streaming server we used the VLC software for transmitting the same sequences used in the previous experiments. We configured the 20Mhz channel 6 in the 2.4GHz band. Newer WiFi standards are the target of our future research but they are also not relevant to the protocol we evaluate here. In this case we measured the real-time value of PSNR at the user for the case of two and three SCBSs. In Fig. 8 we see that indeed our scheme for VPO can achieve the best real-time video quality when compared with the NC implementation. Since here we considered a wireline bakhaul, the performance improvement comes from the lower delay that leads to fewer buffer underrun events in the VLC media player of the client. A typical packet loss in IEEE 802.11a leads to significant backoff delays something that is avoided by our VPO algorithm.

VI. Conclusions

In this paper we presented a framework for video streaming in dense small-cell wireless networks. Our first contribution is the design of an opportunistic packet overhearing algorithm that exploits the natural diversity that this emerging network paradigm offers. Our second contribution is a rate allocation framework that operates in conjunction with the overhearing algorithm. The performance results showed the significant performance benefits of the proposed scheme for high quality 4K video sequences. For this system we demonstrated that it is enough to allow overhearing of video packets and employ our utility-optimized streaming approach instead of employing more sophisticated coding techniques. We also showed that depending on the small cell network backhaul setup, our proposed system can lead to either capacity increase or delay reduction. Our key insights were also validated with a real testbed implementation.

Acknowledgment

This work was funded by the research program CROWN, through the Operational Program Education and Lifelong Learning 2007-2013 of NSRF, which has been co-financed by EU and Greek national funds.

References

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, February 2012.

[2] J. Andrews, "Seven ways that hetnets are a cellular paradigm shift," *Communications Magazine, IEEE*, vol. 51, no. 3, pp. 136–144, 2013.

[3] O. Alay, P. Liu, Y. Wang, E. Erkip, and S. Panwar, "Cooperative layered video multicast using randomized distributed space time codes," *IEEE Transactions on Multimedia*, vol. 13, no. 5, pp. 1127 – 1140, October 2011.

[4] N. Mastronarde, F. Verde, D. Darsena, A. Scaglione, and M. van der Schaar, "Transmitting important bits and sailing high radio waves: A decentralized cross-layer approach to cooperative video transmission," *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 9, pp. 1597–1604, 2012.

[5] A. Argyriou, "Compressed video streaming in cooperative wireless networks with interfering transmissions," *IEEE Transactions on Communications*, vol. 60, no. 11, pp. 3407–3416, November 2012.

[6] H. Seferoglu and A. Markopoulou, "Video-aware opportunistic network coding over wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 27, no. 5, pp. 713–728, 2009.

[7] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou, "Microcast: cooperative video streaming on smartphones," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, ser. ACM MobiSys, 2012, pp. 57–70.

[8] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *IEEE INFOCOM*, 2012.

[9] K. Poularakis, G. Iosifidis, A. Argyriou, and L. Tassiulas, "Video delivery over heterogeneous cellular networks: Optimizing cost and performance," in *IEEE Infocom*, 2014.

[10] A. Argyriou, "Wireless network coding with improved opportunistic listening," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, pp. 2014–2023, April 2009.

[11] "Small cell forum, backhaul technologies for small cells: Use cases, requirements and solutions," 2013.

[12] J. Chakareski and P. Frossard, "Distributed collaboration for enhanced sender-driven video streaming," *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 858–870, August 2008.

[13] D. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Athena Scientific, 1998.

[14] Video Trace Library: http://trace.eas.asu.edu/.

[15] A. Argyriou, "Error-resilient video encoding and transmission in multirate wireless LANs," *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 691–700, August 2008.