In Proc. of the 2010 IEEE International Symposium on Network Coding (NetCod 2010), Toronto, Canada, June 2010.

Network Coding in Passive Optical Networks

Konstantin Miller*, Thorsten Biermann†, Hagen Woesner*, and Holger Karl†

*Berlin Institute of Technology, Berlin, Germany
†University of Paderborn, Paderborn, Germany

{konstantin.miller, hagen.woesner}@tu-berlin.de, {thorsten.biermann, holger.karl}@upb.de

***Abstract*— One of the most promising technologies for high-speed access to the Internet are Passive Optical Networks as they provide high data rates at low cost. We integrate network coding into the Ethernet Passive Optical Network architecture to increase downlink throughput by up to 50 % without changing the hardware.**

In contrast to previous work, we suggest to code packets not only between pairs of nodes but also between an arbitrary number of nodes forming a cycle. We characterize the expected gain analytically and by means of simulations and investigate the trade-off between queuing delay, traffic variability, and throughput gain. We show that in practical scenarios, a simple scheme already achieves a reasonable amount of the maximum possible coding gain.

I. Introduction

Due to their high bandwidth requirements, applications like Peer-to-Peer (P2P) file sharing, high definition television, and video on demand turned out to be a challenge for communication networks [1]. While the networks' backbone capacity has increased to meet these requirements, high costs prevented access networks from scaling up accordingly.

The currently most promising architecture for access networks is a Passive Optical Network (PON) [2]. PONs have only passive, i.e., non-powered, equipment in the field, reducing energy, deployment, and maintenance costs. This way, high data rates can be provided at lower costs compared to, e.g., wireless solutions, even in rural areas.

In this paper, we propose an extension to the standard Ethernet PON (EPON) architecture (Sec. II). This extension uses linear Network Coding (NC) [3] to increase downlink throughput by up to 50 % without changing PON hardware. The approach leverages PON-internal traffic, i.e., traffic whose source and destination are both located within the same PON. Details on the extension are provided in Sec. III. Deployment scenarios and applications where a significant amount of PON-internal traffic is present, like audio/video conferencing, P2P file dissemination, and fiber-wireless access networks with cooperating base stations are discussed in Sec. IV. For P2P traffic, a method to increase traffic locality without interfering with traffic semantics is given in Sec. V. Sec. VI presents evaluations that demonstrate the benefits of NC in EPONs.

The research leading to these results has been conducted in the 4WARD project and received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216041.

II. Ethernet Passive Optical Networks

The EPON architecture (Fig. 1) is defined in the IEEE 802.3 standard. EPONs have a physical tree topology with the root, the Optical Line Terminal (OLT), residing at the central office of the Internet Service Provider (ISP). The OLT is connected to several Optical Network Units (ONUs) in the field via an optical fiber that is split up using 1:N optical splitters/combiners. This creates a tree topology that provides a broadcast medium in the downstream (OLT $\rightarrow$ ONUs) and point-to-point links with a common collision domain in the upstream (ONUs $\rightarrow$ OLT). Since the upstream and the downstream channels use different wavelengths, their operation is completely independent.

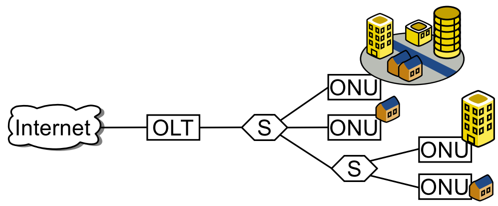


Fig. 1: Example PON tree. The OLT provides connectivity to four ONUs via two splitters (S). A single ONU can connect a single house, a company, or a whole district.

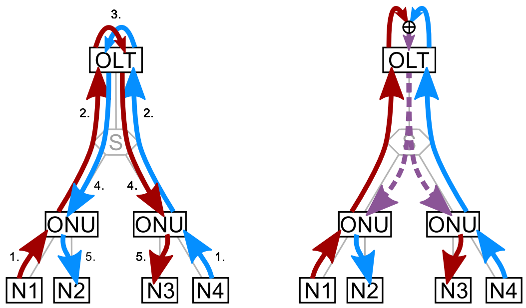
Upstream collisions are avoided by the Multi-Point Control Protocol. It is controlled by the Dynamic Bandwidth Allocation (DBA) process that resides at the OLT and applies time division multiplexing among ONUs [2]. The upstream bandwidth allocation algorithm is not part of the standard. Several alternatives were proposed in the literature. One of the most prominent is IPACT [4]. It assigns timeslots to ONUs in a round-robin manner, based on the reported queue lengths.

Within the EPON, Ethernet frames are used to transport data. To address a certain ONU, a Logical Link ID (LLID) is prepended to each frame, replacing two octets of the preamble. This way, the broadcast medium can be exploited for "free" single-copy multicast operations, like in wireless, by introducing multicast LLIDs (up to $2^{15} - 1 = 32767$ ONU groups). We will exploit this feature later on. ONUs filter all incoming frames and pass only those frames to the Medium Access Control (MAC) that carry the own (group) LLID.

III. Network Coding extension for Ethernet Passive Optical Networks

A. Concept

In our context, NC can be introduced in a straightforward manner. Assume, any node connected to ONU_1 wants to send a packet p_1 to a node connected to ONU_2. At the same time, a packet p_2 is to be sent in the opposite direction (from a node connected to ONU_2 to a node connected to ONU_1), not necessarily between the same two nodes. Without NC, two packet transmissions are required upstream and two downstream (Fig. 2a).



(a) No NC active. (b) NC is activated.

Fig. 2: PON-internal data flow with and without NC. N1 to N4 are nodes behind an ONU, e.g., connected via a LAN. The small numbers indicate the order of the various transmissions.

We modify the OLT behavior such that instead of forwarding p_1 to ONU_2 immediately, it waits until the reception of p_2. If p_2 arrives with a maximum delay of $d_{\max}$, both packets are linearly combined to a new packet $p := p_1 \oplus p_2$, where $\oplus$ is the bit-wise exclusive or (XOR) operation. The new packet p is then sent to the multicast group consisting of ONU_1 and ONU_2. Upon reception, ONU_1 decodes p_2 by applying the XOR operation once again: $p_2 = p \oplus p_1$; ONU_2 decodes by $p_1 = p \oplus p_2$. If p_2 does not arrive at the OLT within $d_{\max}$, p_1 is forwarded downstream uncoded. The resulting data flow with activated NC is illustrated in Fig. 2b.

Since multicast in EPONs exploits the broadcast nature of the underlying medium, only one time slot is required per downstream multicast operation. Hence, downstream throughput increases by 50 % by using the time slots freed by NC. Note that a similar approach has been studied in the context of wireless networks, see, e.g., [5].

The presented mechanism can be extended to scenarios with no direct communication between any ONU pair. Instead, any cyclic traffic is sufficient. Assume that, e.g., ONU_1 sends a packet p_1 to ONU_2, which sends a packet p_2 to ONU_3. ONU_3, in turn, sends a packet p_3 to ONU_1 again. (The 3 transmissions can occur in an arbitrary order.) Here, NC can be applied in the following way. The OLT multicasts packet $p' := p_1 \oplus p_3$ to ONU_1 and ONU_2, and packet $p'' := p_2 \oplus p_3$ to ONU_2 and ONU_3. Upon reception, ONU_1 decodes by applying $p_3 = p' \oplus p_1$, ONU_2 decodes by $p_1 = p' \oplus p'' \oplus p_2$, and ONU_3 decodes by $p_2 = p'' \oplus p_3$. Instead of 3 downstream transmissions, the OLT only requires 2 transmissions, achieving still a gain of 33 %. In general, the NC gain for any cyclic traffic is $\frac{1}{n}$, where n is the number of nodes forming the cycle. (Note that bidirectional traffic is a special case of cyclic traffic with $n = 2$.)

B. Implementation at the OLT

We propose the following scheme to apply NC to traffic cycles with lengths up to $c_{\max}$. The scheme runs at the OLT and requires a buffer matrix $\{b_{ij}\}$. Each buffer b_{ij} stores packets from ONU_i to ONU_j together with their arrival times.

1) Whenever the OLT receives a packet p_i from ONU_i to ONU_j, it constructs a directed graph $G = (V, E)$, where each vertex $v \in V$ corresponds to an ONU and edges ij correspond to non-empty buffers b_{ij}. This graph is analyzed to find the smallest cycle of maximum length $c_{\max}$ that contains edge ij. This can be done very efficiently, e.g., by using Dijkstra's shortest path algorithm [6].
   a) If no cycle is found, the received packet is stored in buffer b_{ij} together with its arrival time. If the buffer was empty, a timer with the packet's maximum buffering time $d_{\max}$ is started.
   b) If such a cycle is found, the OLT performs the corresponding coding operations and transmits the coded packets. For each packet removed from a buffer, the corresponding timer has to be restarted with a duration equal to the remaining maximum buffering time of the oldest packet in that buffer, or canceled if there is no remaining packet.
2) Whenever a timeout occurs for buffer b_{ij}, the oldest packet in this buffer is sent downstream uncoded. The timer is restarted with a duration equal to the remaining buffering time of the now oldest packet in the buffer. If no packets remain in the buffer, the timer is canceled.

Note that the presented scheme considerably simplifies if we consider only cycles of maximum length 2, that is, if only bidirectional traffic is coded. In this case, the construction of a graph and the search for cycles reduces to one single test if buffer b_{ji} is empty. In Sec. VI-B, we will present an analysis suggesting that the additional gain achieved by considering cycles with length greater than 2 is relatively small.

Note that a simulative study of NC in PONs without additional buffering at the OLT is performed in [7]. There, coding is performed among packets that are queued at the OLT due to congestion.

C. Implementation at ONUs

To decode incoming packets, an ONU has to keep a copy of each outgoing upstream packet whose destination is located within the same PON. Thus, like the OLT, ONU_i maintains a buffer b_{ij} for each destination ONU_j with $j \neq i$. Whenever ONU_i sends a packet p_i to ONU_j, a copy is stored in b_{ij}.

Now, consider the case where coding is performed over traffic cycles of size 2 (bidirectional traffic). In this case, one of the following can happen. (1) The OLT forwards p_i to

ONU$_j$ uncoded, (2) the OLT encodes p_i with another packet p_j going from ONU$_j$ to ONU$_i$ and multicasts the resulting packet $p_i \oplus p_j$ to both ONUs, or (3) p_i is dropped, e.g., due to congestion. We handle all three cases without the need for additional packet identifiers by exploiting once again the broadcast property of the downstream. In the first case (no NC), the sending ONU$_i$ overhears the uncoded downstream transmission of its own packet p_i and discards the locally stored copy. In the second case (NC), ONU$_i$ receives the encoded packet $p_i \oplus p_j$. It is then decoded using the stored copy, which is removed from the buffer afterwards. In the third case (packet loss), it must be ensured that copies of packets that are lost are eventually removed from the buffer. To achieve this, in the first and second case, ONU$_i$ not only removes the stored copy of p_i but also copies of packets that were sent to ONU$_j$ prior to p_i. Those are exactly the lost packets.

The proposed scheme requires that a source ONU is able to determine the destination ONU of each packet it sends upstream. Therefore, each ONU maintains a list of IP address ↔ LLID mappings, similar to an ARP cache. These mappings are created/updated whenever an ONU overhears a downstream frame that contains an IP datagram.

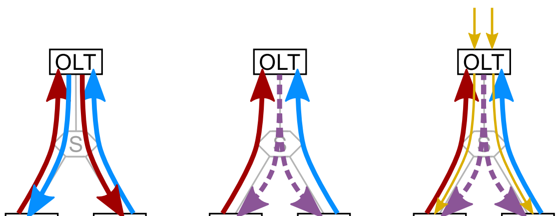
If coding shall be done for traffic cycles that contain more than 2 ONUs, additional mechanisms are required to inform the ONUs about how to decode encoded packets. An example for such a signaling mechanism together with its performance evaluation can be found in [8].

IV. Application Scenarios

This section discusses several application scenarios that benefit from the proposed NC scheme in PONs.

A. General strategies

In general, NC helps in scenarios where the PON downstream is a bottleneck. In this case, NC reduces downstream traffic whenever the traffic is cyclic and PON-internal, like, e.g., for interactive voice and video communication. This is illustrated for bidirectional traffic in Fig. 3.



(a) No NC. Up- and downlink are saturated.

(b) NC active. Downlink utilization is reduced by 50 %.

(c) Free downstream capacity is used for additional traffic.

Fig. 3: Exploiting NC gain when traffic is PON-internal and bidirectional.

Additionally, some applications allow to influence the traffic patterns to generate cyclic and PON-internal traffic when the source of data does not matter, like for P2P file transfers or information-centric networking [9]. Instead of downloading a data chunk from an external peer, it is fetched from a peer within the same PON. After replacing external with internal traffic, the situation is identical to Fig. 3a. NC then allows to free downstream capacity that can be used, e.g., for additional data chunk downloads from peers that are located outside the PON.

B. Video and voice communication

Traffic generated by video communication is expected to have the fourth highest annual growth rate of 58 % from 2008 to 2013 [1] and further growth of this traffic type is expected beyond 2015 [10]. These numbers do not include video streaming from servers to clients.

In video and voice communication, there are always at least two communicating partners. Hence, there is a bidirectional traffic pattern by nature which leads to large benefits in NC-enabled PONs transporting such flows.

C. Peer-to-Peer

Today, P2P traffic constitutes the largest fraction of Internet traffic with approximately 50 % [1]. This fraction is expected to decrease to one third until 2013 (due to the high growth of video traffic) but will still be the second largest part of the overall traffic.

Some of today's P2P protocols, like BitTorrent, have built-in mechanisms that generate bidirectional traffic by nature, e.g., "tit-for-tat". Furthermore, in order to reduce inter-provider costs and to increase performance, many research activities focus on controlling P2P transfers to exploit locality on provider level [11], [12]. We adopt and extend this idea to create PON-internal and bidirectional P2P traffic. Details will be given in Sec. V.

D. Future cellular networks

Future 4G cellular networks, like Long Term Evolution (LTE) Advanced [13], target data rates of 1 Gbit/s downstream and 500 Mbit/s upstream. These high rates cannot be achieved with the backhaul infrastructure of today's radio access network. Hence, cheap, optical techniques, like PONs, are a popular candidate for replacing existing infrastructure [14].

Exploiting cooperative diversity plays an important role in 4G networks. For this, base stations have to be coordinated and a lot of traffic has to be exchanged between base stations for combining packets [15]. Both cause bidirectional traffic.

V. Peer-to-peer oracle

A lot of research has been done to optimize P2P overlays by exploiting knowledge about the underlying network infrastructure [11], [12], [16], [17]. By preferably downloading from peers that are connected to the same ISP, downloading performance increases and inter-provider traffic decreases. Both the user and the ISP benefit. It has been shown that up to 40 % of P2P traffic within a campus network can be made local [18], i.e., the downloads can be carried out within the local network.

The cooperation between peers and ISPs can be implemented such that whenever a peer intends to download a piece of data, called chunk, it contacts an *oracle* server [12]. This oracle is operated by the ISP (and hence knows the network infrastructure) and tells the peer from which other peer it should download the data.

We use this idea to not only keep P2P transfers within the same ISP network but also within the same PON tree whenever possible. This permits to increase the amount of bidirectional traffic between ONUs and thus to increase throughput by using NC.

Obviously, the strategy persued by the oracle when selecting download partners is critical for the performance of the whole system. Since the focus of our study, however, is not on oracle strategies, we use a simple heuristic approach for the performance evaluation in the next section.

The strategy works as follows. Each time a peer is ready to download a new chunk, it sends a request to the oracle. This request contains a list of remaining chunks that are required by the peer. Based on this list, the oracle now decides which chunk is downloaded next from which other peer. The decision is returned to the peer, which starts the download accordingly.

In order to implement this strategy, the oracle maintains a data base that stores which chunks are available at which peers within the PON. Additionally, it keeps track of currently ongoing downloads. Given that, the oracle is able to create a list of peers, located within the PON, that are downloading *from* the requesting peer and that can provide a chunk that is useful *to* the requesting peer. If the list is not empty, a random peer is chosen from that list and a required chunk is downloaded from that peer. Otherwise, a chunk is downloaded from an arbitrary peer within the PON to permit bidirectional traffic for future chunk downloads. In case that no peer within the PON is able to provide a chunk required by the requesting peer, a chunk is downloaded from an external peer.

VI. EVALUATION

This section evaluates the gain of NC in different scenarious. It contains both analytical and simulative results.

A. Expected gain for bidirectional traffic

To get a first idea about possible NC gains and requirements that have to be fulfilled in order to gain, we evaluate the scenario for bidirectional traffic analytically. For this, we assume that each traffic flow's rate R_{ij} per coding period d_{max} from ONU$_i$ to ONU$_j$ is a random variable with a normal distribution, $N(\mu, \sigma^2)$; all R_{ij} are identically distributed and independent both among ONU pairs and across successive timeslots.[1] To normalize the variability of these flows and to ease the presentation of results later on, we will not use the standard deviation σ but the coefficient of variation $c_v = \frac{\sigma}{\mu}$, hence $R_{ij} \sim N\left(\mu, (\mu c_v)^2\right)$.

The amount of codeable traffic between ONU$_i$ and ONU$_j$ is the minimum of R_{ij} and R_{ji}. This is also the absolute gain G_{abs} (two packets from OLT to ONUs replaced by one network-coded packet); its expected value can be calculated according to Eq. (1). The expected value of the minimum of two independent, normally distributed random variables is solved in [19].

$$G_{\text{abs}} = E\left[\min(R_{ij}, R_{ji})\right] = \mu - \frac{\mu c_v}{\sqrt{\pi}} = = \mu\left(1 - \frac{c_v}{\sqrt{\pi}}\right) \quad (1)$$

Dividing G_{abs} by the total traffic produced by two ONUs whose traffic is encoded (2μ) gives the relative gain G_{rel}. It only depends on c_v and is derived in Eq. (2).

$$G_{\text{rel}} = \frac{G_{\text{abs}}}{2\mu} = \frac{1}{2}\left(1 - \frac{c_v}{\sqrt{\pi}}\right) \quad (2)$$

The gain becomes larger the less variable the traffic flows are. But traffic can be smoothed by buffering it for a longer time. Hence, we consider time as additional dimension to relate the maximum buffering time at the OLT d_{max} to c_v. The effect of doubling d_{max} can be seen as combining the traffic of two adjacent time slots of length d_{max}. Under the assumption that traffic in subsequent time slots of length d_{max} are i.i.d. and normal-distributed as well, the random variable $\tilde{R}_{ij}$, which represents the traffic during both time slots, is the sum of $R_{ij} + R_{ij} \sim N\left(2\mu, 2\sigma^2\right)$. Eq. (3) shows the traffic's new variance coefficient $\tilde{c}_v$.

$$\tilde{c}_v = \frac{\sqrt{2\sigma^2}}{2\mu} = \frac{\sqrt{2}\sigma}{2\mu} = \frac{1}{\sqrt{2}}\frac{\sigma}{\mu} = \frac{1}{\sqrt{2}}c_v \quad (3)$$

If for a given d_{max} the resulting traffic has a variance of c_v, buffering for $2^l d_{\text{max}}$ leads to a gain of $G_{\text{rel}}(l)$ (Eq. (4)).

$$G_{\text{rel}}(l) = \frac{1}{2}\left(1 - \frac{\tilde{c}_v(l)}{\sqrt{\pi}}\right) = \frac{1}{2}\left(1 - \frac{1}{\sqrt{2}^l}\frac{c_v}{\sqrt{\pi}}\right) \quad (4)$$

This equation lets us calculate the required d_{max} (in terms of the factor exponent l) to achieve the desired NC gain for a certain traffic with variance coefficient c_v. Fig. 4 shows the relation between c_v, l and the resulting G_{rel}. To achieve a large gain, the traffic has to be smooth (small c_v) or larger buffers and long delays are required (large l).

B. Expected gain for arbitrary traffic cycles

As mentioned in Sec. III, NC can be applied for any cyclic PON-internal traffic. The gain, however, decreases as the cycle size increases, and detecting larger traffic cycles is more difficult. This leads to the question whether spending effort for detecting large traffic cycles is beneficial at all.

To answer this question, we simulated the abstract traffic model from the previous section in a PON consisting of 8 ONUs. We analyzed the traffic to detect cyclic flows with up to c_{max} involved ONUs and finally calculated the mean gain when applying NC for the detected cycles. The maximum

[1] The assumptions normality and independence are of course both arguable. Replacing the normal distribution by another distribution does not change the exposition substantially; not assuming independence, however, requires a considerably more complex analysis.

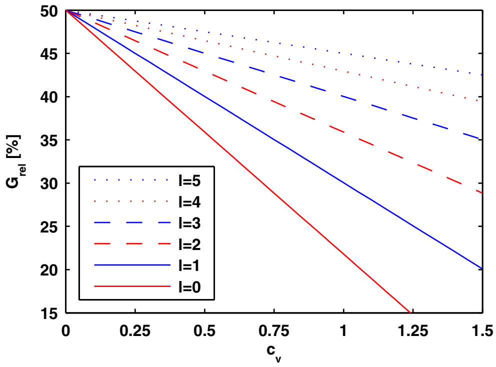


Fig. 4: Relative NC gain G_{rel} depending on c_v for different factor exponents l.

cycle size $c_{\max}$ is varied from 2 (bidirectional flows) to 5. Coding opportunities are detected in a greedy way to reduce complexity as described in Sec. III-B: Smaller cycles (higher gains) are searched for first. Thereafter, remaining traffic is analyzed for larger cycles.

Note that for the goal of reducing the overall downstream load it does not matter whether small or large traffic cycles are searched first. It can be shown that for two overlapping traffic cycles the same NC gain is achieved independently of whether the smaller or larger one is encoded first. The resulting overall load for two circular flows of size n and $n+1$ having k links in common and rates r_n and r_{n+1} is always $l_{n,n+1}$ (Eq. (5)).

$$l_{n,n+1} = n(r_n + r_{n+1}) - kr_n \tag{5}$$

According to the traffic model, each ONU pair ONU$_i \rightarrow$ ONU$_j$ is assigned a certain data rate r_{ij}, with $R_{ij} \sim N(6, \sigma)$. Different application/traffic types with more or less symmetry properties are simulated by varying σ. The resulting gains and analytic values for $c_{\max} = 2$ are shown in Fig. 5.

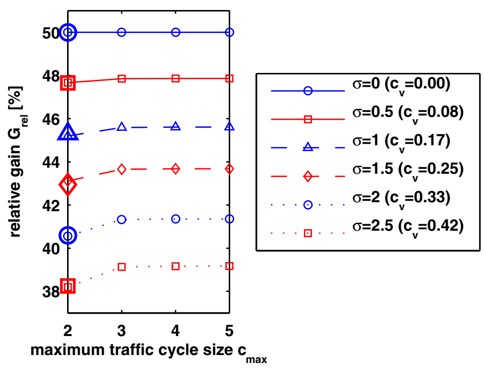


Fig. 5: Expected NC gain depending on $c_{\max}$ and σ. Besides the simulation results, analytic values according to Eq. (2) are plotted (large markers) for $c_{\max} = 2$ to validate the model.

The plot shows the expected behavior that smaller values for σ (i.e., more balanced/symmetric traffic) lead to higher gains. Furthermore, increasing $c_{\max}$ only shows additional gains for high σ values. These gains, however, are only in the order of a few percents. Hence, in the following evaluations, we only consider bidirectional traffic for NC.

C. Performance for real world traffic

We implemented the proposed NC extension for bidirectional traffic in the OMNeT++ simulator.

In the evaluated setting, clients are connected to ONUs via intermediate routers, while the OLT is connected to the Internet via a gateway router. ONU routers forward all upstream traffic to the gateway router, which either forwards it to the Internet or, in case of internal traffic, to the router of the destination ONU. As a result, the EPON operates as a distributed switch, i.e., it is transparent to the clients. All links (optical and non-optical) operate at a data rate of 1 Gbit/s.

For the scheduling of upstream transmissions, we used the *IPACT* [4] Dynamic Bandwidth Allocation (DBA) algorithm, which assigns timeslots to ONUs in a round-robin manner.

In order to avoid joint encoding of very large and very small packets, e.g., data and acknowledgment packets, as this would lead to a low NC gain, we only feed packets into the encoder that exceed a size of 500 byte. This procedure has the advantage that it is very easy to implement. On the other hand, it clearly wastes some coding opportunities. To overcome this drawback, flow synchronization techniques [8] can be applied before feeding traffic into the encoder.

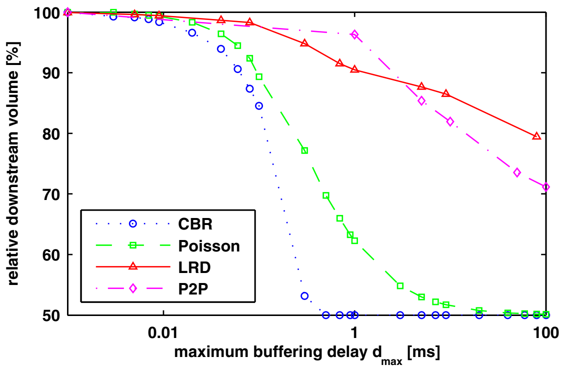
Obviously, $d_{\max}$ is a critical parameter for the performance. If $d_{\max}$ is too short, fewer coded packets are generated and the NC gain decreases. If $d_{\max}$ is too high, real-time applications suffer from delay and buffers need to be larger. Further, a high value of $d_{\max}$ increases the Round Trip Time (RTT) variance and thus decreases TCP throughput.

We simulated two different traffic scenarios. In the first one, two ONUs exchange traffic with a rate of 5 Mbit/s and a packet size of 200 byte (resulting in a mean inter-arrival time of 320 μs). With these parameters, we simulated three different types of streams: Constant Bit Rate (CBR) and Poisson streams, and streams with Long-Range Dependent (LRD) inter-arrival times.

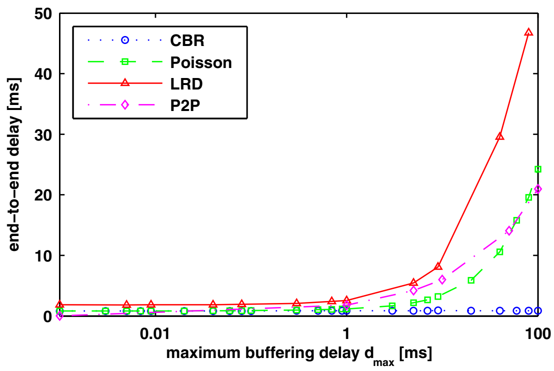
In the second traffic scenario, users participate in a 760 Mbyte BitTorrent-like P2P download that is subdivided into 152 chunks of 5120 kbyte; each user is allowed to retrieve up to 10 chunks in parallel. Each time a peer intends to start a new chunk download, the oracle is contacted to provide a source peer. The oracle uses the strategy described in Sec. V.

Results of both traffic scenarios are shown in Fig. 6. Fig. 6a illustrates the total amount of data sent downstream by the OLT, normalized to the amount of data that is sent when NC is disabled. Fig. 6b shows the average end-to-end delay experienced by PON-internal packets.

Simulations show that NC reduces the mean downstream utilization by up to 50 %, or, in other words, increases downstream throughput by 50 %, if enough coding opportunities are present. In the case of two CBR streams, a value of $d_{\max} = 1$ ms is enough to achieve the maximum possible gain



(a) Relative downstream volume



(b) End-to-end delay

Fig. 6: Impact of NC on the relative downstream volume and end-to-end delay for CBR, Poisson, and LRD streams, as well as for TCP-based P2P downloads.

without sacrificing delay. For the P2P download, the mean downstream utilization is reduced by up to 30 %. The smaller gain compared to the CBR scenario is due to the fact that a certain fraction of data has to be retrieved from outside of the PON, where NC cannot be applied. Further, due to the burstiness of TCP traffic, packet inter-arrival times can temporarily exceed $d_{\max}$, even if the latter is set to a high value. Also note that our oracle uses a simple heuristic to generate bidirectional traffic. Future work on oracle strategies will further improve efficiency.

We also evaluated the time-to-completion that peers experience for a network-coded P2P download. This time slightly increases compared to the non-coded transfer as TCP suffers from the increased end-to-end delay variance. This problem, however, including many solution approaches, is well-known for other scenarios, like 3G networks, as well [20].

VII. Conclusion

We showed that our NC extension for EPONs increases downstream throughput by up to 50 %. Thus, PON capacity can be significantly increased without expensive hardware upgrades. Asymmetries or random variations in the traffic patterns diminish these gains. We have shown the benefits of buffering for a longer time at the OLT (trading off capacity against delay and buffer space) and using more complex coding patterns over increasingly long traffic cycles (trading off capacity against computational overhead at the OLT) to combat such imbalances. These results are contrasted with simulations for real-world traffic (in particular, P2P), showing that even with simple oracle strategies and acceptable buffering delay good gains are achievable.

We leave it for future work to explore additional gains through NC-optimized DBAs, heterogeneous $d_{\max}$ values for different ONU pairs, dynamic adaption of $d_{\max}$ according to traffic characteristics, and optimized oracle strategies. Furthermore, we intend to generalize our analytic framework to cover more complex traffic types as well.

References

[1] Cisco Systems, Inc., "Cisco visual networking index – forecast and methodology, 2008-2013," June 2009, white paper.

[2] M. P. McGarry, M. Reisslein, and M. Maier, "Ethernet passive optical network architectures and dynamic bandwidth allocation algorithms," *IEEE Communications Surveys and Tutorials*, vol. 10, pp. 46–60, 2008.

[3] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, pp. 371–381, 2003.

[4] G. Kramer, B. Mukherjee, and G. Pesavento, "IPACT: A dynamic protocol for an Ethernet PON (EPON)," *IEEE Comm. Mag.*, vol. 40, pp. 74–80, 2002.

[5] S. Katti, H. Rahul, H. Wenjun, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," *IEEE/ACM Trans. on Networking*, vol. 16, pp. 497–510, 2008.

[6] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications* . Prentice Hall, 1993.

[7] M. Belzner and H. Haunstein, "Network Coding in Passive Optical Networks," in *Proc. ECOC*, 2009.

[8] T. Biermann, M. Dräxler, and H. Karl, "Flow synchronization for network coding," *Journal of Communications*, vol. 4, no. 11, pp. 873–884, 2009.

[9] B. Ahlgren, M. D'Ambrosio, M. Marchisio, I. Marsh, C. Dannewitz, B. Ohlman, K. Pentikousis, O. Strandberg, R. Rembarz, and V. Vercellone, "Design Considerations for a Network of Information," in *Proc. ACM CoNEXT*, 2008.

[10] Cisco Systems, Inc., "Approaching the zettabyte era," June 2008, white paper.

[11] J. Seedorf and E. Burger, "Application-layer traffic optimization (ALTO) problem statement," IETF Draft (Informational), Apr. 2009.

[12] V. Aggarwal and A. Feldmann, "Locality-aware P2P query search with ISP collaboration," *Netw. and Heterog. Media*, vol. 3, no. 2, pp. 251–265, 2008.

[13] E. Seidel, "Progress on "LTE Advanced" – the new 4G standard," July 2008, white paper.

[14] S. Sarkar, S. Dixit, and B. Mukherjee, "Hybrid wireless-optical broadband-access network (WOBAN): A review of relevant challenges," *Journal of Lightwave Technology*, vol. 25, pp. 3329–3340, 2007.

[15] C. Hoymann, L. Falconetti, and R. Gupta, "Distributed uplink signal processing of cooperating base stations based on IQ sample exchange," in *Proc. IEEE ICC*, 2009.

[16] J. P. F.-P. Gimenez, M. A. C. Rodriguez, H. Hasan, T. Hoßfeld, D. Staehle, Z. Despotovic, W. Kellerer, K. Pussep, I. Papafili, G. D. Stamoulis, and B. Stiller, "A new approach for managing traffic of overlay applications of the SmoothIT project," in *Proc. AIMS*, 2008.

[17] H. Xie, A. Krishnamurthy, A. Silberschatz, and Y. R. Yang, "P4P: Explicit communications for cooperative control between P2P and network providers," May 2007, white paper.

[18] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should internet service providers fear peer-assisted content distribution?" in *Proc. ACM IMC*, 2005.

[19] H. A. David, *Order statistics*. New York: Wiley, 1970.

[20] M. C. Chan and R. Ramjee, "Improving TCP/IP Performance over Third-Generation Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 4, pp. 430–443, Apr. 2008.