

# Optimized Cooperative Streaming in Wireless Mesh Networks

Luca Baldesi, Leonardo Maccari, Renato Lo Cigno

Department of Information Engineering and Computer Science (DISI), University of Trento, Italy

{luca.baldesi, leonardo.maccari, renato.locigno}@disi.unitn.it

**Abstract**—Peer-to-peer video streaming is a valuable technique to reduce the overhead produced by centralized and unicast-based video streaming. Key to the efficiency of a peer-to-peer approach is the optimization of the logical distribution topology (the *overlay* with respect to the underlying network, the *underlay*). This work studies peer-to-peer streaming in wireless mesh networks for which the underlay is known. We propose an optimized, cross-layer approach to build the peer-to-peer distribution overlay minimizing the impact on the underlay. We design an optimal strategy, which is proven to be NP-complete, and thus not solvable with a distributed, light weight protocol. The optimal strategy is relaxed exploiting the knowledge of the betweenness centrality of the underlay nodes, obtaining two easily implementable solutions applicable to any link-state routing protocol. Simulation and emulation results (experimenting with real applications on a network emulated with the Mininet framework) support the theoretical findings, showing that the relaxed implementations are reasonably close to the optimal solution, and provide vast gains compared to the traditional overlay topology based on Erdős-Rényi models that a peer-to-peer application would build.

## I. INTRODUCTION

Video streaming is the major component of the global Internet traffic, and it is thought to be increasing for several years to come. Today streaming is normally delivered with several seconds of delay, but users are asking for more performing systems, and video calls and conferences, which are still somewhat “rare”, require a much more timely delivery of the video. Currently, the majority of the Video Service Providers (VSPs) deliver video streams using unicast traffic and leveraging centralized platforms supported by world-wide Content Delivery Networks (CDNs).

Ten years ago, the peer-to-peer (P2P) paradigm and technologies promised to offer a solution for massive content distribution, including video streaming and conferencing. The reasons P2P systems were unable to meet their potential are many, a key one was the difficulty to realize P2P overlays optimized from the point of view of the Internet Service Providers (ISPs).

An application field in which this problem can be overcome is the field of wireless mesh networks, and in particular Community Networks (CN from now on). CNs are large

This work was financed partially by the University of Trento under the grant “Wireless Community Net-works: A Novel Techno-Legal Approach” —Strategic Projects 2014, and partially by the European Commission, H2020-ICT-2015 Programme, Grant Number 688768 ‘netCommons’ (Network Infrastructure as Commons).

ISBN 978-3-901882-83-8 © 2016 IFIP

mesh networks (primarily made of wireless links) that are flourishing in many different scenarios, from the developing country where there is no other connectivity means, to the urban areas of western cities where they compete with other network providers. The steep decrease of the prices of outdoor wireless equipment makes it possible to build cooperative wireless mesh networks with links that can achieve tens of Mbit/s and support CNs made of hundreds of nodes. The most prominent example is the Guifi network ([www.guifi.net](http://www.guifi.net)) that is a collection of various networks in East Spain, for a total of about 30,000 nodes [1]. In CNs the underlay is normally known, since the routing protocols exports it to each node (as long as a link-state routing protocol is used), which removes one of the technical barriers that blocked the deployment of P2P video streaming on the Internet. We assume that the distribution engine of the video is installed in the mesh nodes, as already proven feasible in [2], while the video can be enjoyed on standard terminals, so that underlay details are easily accessible to the overlay manager.

The contribution of this paper is a cross-layer optimization scheme to perform *live* video streaming (i.e., with a strict deadline on the arrival delay) in mesh networks [3]. The optimization minimizes the impact of the streaming overlay on the underlay network exploiting information on the topology and routing of the underlay. The case of non real-time streaming can be seen as a sub-case of live streaming relaxing the constraints on delivery delay.

The optimization is based on the concept of *centrality*, which is also at the base of successful algorithms as Google PageRank [4]. Taking into account the centrality of peers in the underlay graph, the optimized overlay topology greatly improves the efficiency of the video distribution and maintains high performance. The resulting algorithm and protocol are first tested on synthetic topologies, showing that they are robust and efficient; finally they are implemented within PeerStreamer<sup>1</sup>, an existent live P2P-streaming platform, and tested using a network emulator configured with the topologies of real wireless mesh networks.

## II. MOTIVATION AND PROBLEM STATEMENT

We consider a communication system where the cooperative distribution exploits a logical topology called the *overlay* built on top of a meshed routing network called the *underlay*.

<sup>1</sup>PeerStreamer is Open Source effort supported by the DISI-ANS research group of the University of Trento. See <http://peerstreamer.org>

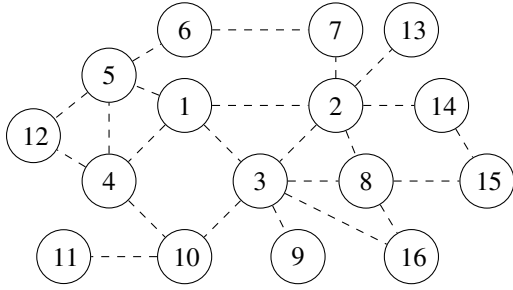


Fig. 1: Example of wireless mesh underlay graph. Hosts are numbered with an arbitrary ordering. Dashed lines represent the wireless links.

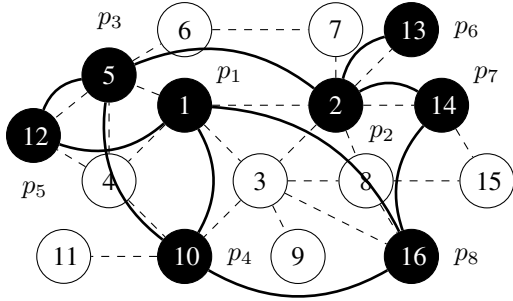


Fig. 2: Possible overlay graph (black vertexes and edges) over the underlay graph of Fig. 1.

Nodes in the *overlay*, that from now on we call peers, do have access to information concerning the *underlay*, including details on the topology and quality of its links. This is true for a wireless mesh network using link-state routing protocols, in which every wireless router needs to know the whole topology of the network to perform routing. For instance, standard implementations as the OLSRd daemon implementing the Optimized Link State Routing (OLSR) protocol [5] export the topology with a simple API. The optimization we propose regards the choices of the edges in the *overlay* so that the impact on the *underlay* is minimized and evenly distributed.

We model the *underlay* with an undirected graph  $U(H, L)$  with vertexes  $h \in H$  called hosts or nodes, and edges  $l \in L$  called links. The size of  $H$  is between a few tens up to a thousand hosts, which corresponds to the realistic size of a CN [6]. Fig. 1 shows the graph representation of a sample underlay with 16 hosts.

The peers form an *overlay* that is also modeled as an undirected graph  $O(P, E)$  with vertexes  $p \in P$  called peers, and edges  $e \in E$  called virtual or logical links. Each peer resides in one host only, and it has access to information pertaining to  $U$ , including the association between peers and hosts. Fig. 2 depicts a possible overlay graph on the underlay of Fig. 1.

The goal is to find a viable (meaning that can be implemented as a distributed system with limited signaling overhead and acceptable computational overhead) methodology to select virtual links between peers to build  $O(P, E)$  given  $U(H, L)$  and  $P$  so that the load imposed by the video streaming on the *underlay* links is minimized, and links are loaded as fairly

TABLE I: Summary of the main symbols used through the paper and their meaning.

Peers, Hosts, Links and Virtual Links sets	$P, H, L,$ and $E$
Overlay and underlay graphs	$U(H, L), O(P, E)$
Fairness of $O(P, E)$ over $U(H, L)$	$\mathcal{F}$
Network load of $O(P, E)$ over $U(H, L)$	$\mathcal{L}$
$k$ th undirected overlay edge in $P \times P$	$e_k$
Cross-layer overlay edge descriptor of $e_k$	$\bar{e}_k$
$i$ th overlay peer	$p_i$
Set of the overlay edges in $O(P, E)$ linking $p_i$	$S'_i$
Family of all $S'_i, i = 1, \dots,  P $	$F'$
$O(P, E)$ as an intersection graph	$\Omega(F')$
Target function on $\Omega(F')$ addressing $\mathcal{L}$ and $\mathcal{F}$	$O_c$
Binary variable representing of whether $e_k \in E$	$z_k$
Estimation of link usage in $O(P, E)$	$b$

as possible.  $O$  is dynamically created and maintained because both  $O$  and  $U$  can change frequently, so the modification of  $O$  must be fast and efficient.

#### A. Formal Problem Definition

Tab. I reports the main notation we use in the paper. Given an edge  $e$  connecting  $p_i, p_j \in P$ , we call  $D(e)$  the Dijkstra function returning an (ordered) set of links in the underlay that form the shortest path from the host where  $p_i$  resides and the host where  $p_j$  resides. For example in Fig. 2 we have:

$$e = (p_1, p_4) \mapsto D(e) = \{(1, 3), (3, 10)\}$$

where  $(i, j)$  is the link  $e$  connecting hosts  $i$  and  $j$ . A generic weight  $w(l)$  is assigned to any link in the *underlay*, so that the load  $\mathcal{L}$  imposed by  $O$  on  $U$  is

$$\mathcal{L} = \sum_{e \in E} \sum_{l \in D(e)} w(l) \quad (1)$$

This representation perfectly fits the routing protocols that use the ETX metric [7], ETX is the expected average number of frames sent on the link to correctly deliver one frame, and it is used by OLSR and other protocols.

Every link  $l$  is loaded by a number of virtual links. To measure fairness, we use Jain's fairness index on the distribution of the number of logical links insisting on every  $l$ . Let  $\mathcal{H}(l)$  be the number of logical links loading  $l$ :

$$\mathcal{H}(l) = |\{e \in E : l \in D(e)\}| \quad (2)$$

where  $|\cdot|$  is the size of a set. The Jain's fairness is defined as

$$\mathcal{F} = \frac{(\sum_{l \in L} \mathcal{H}(l))^2}{|L| \sum_{l \in L} \mathcal{H}(l)^2} \quad (3)$$

Jain's fairness is maximal if  $\mathcal{F} = 1$  and minimal when  $\mathcal{F} = \frac{1}{|L|}$ , but we do not expect that maximal fairness can be reached, as in general there are links in the *underlay* that do not support any edge in the *overlay*.

The contribution of this paper is twofold:

- First we derive a formal framework wherein it is possible to define an optimization problem that allows finding the topology of  $O(P, E)$  given  $U(H, L)$  and  $P$  that minimizes a metric composed of  $\mathcal{L}$  and  $\mathcal{F}$ . The problem is NP-complete and we'll show that it can be reduced to a quadratic knapsack problem [8];

- Second we propose two relaxations of the optimization in decreasing complexity order and we show, with numerical solutions and with an implementation in a real P2P streaming platform, that the two relaxations are close to the global optimum and that they vastly outperform the tradition P2P *overlay* building based on selecting uniformly neighboring peers to build an Erdős-Rényi graph.

The paper focuses on building a mesh overlay and not on how video chunks are distributed on it. On mesh topologies this latter problem can be tackled with several different strategies. Even if this is not the focus of the paper, Sec. III briefly refers some relevant works on this topic, justifying our choice for the distribution strategy we use in Sec. VI and Sec. VII. The assumption we start from, which becomes a constraint of the optimization problem, is that each peer should receive exactly one copy of every chunk. We assume also that each peer will contribute to the dissemination serving chunks to other peers proportionally to its degree in  $O$ , so that the chunk dissemination strategy becomes agnostic of the volume of served chunks, simplifying it. This is one reason why the *overlay* cannot simply be a full mesh, since the traffic generated by each node is proportional to its degree. Another reason is that to maintain a set of neighbors each peer will periodically send probe messages to verify their state and possibly the network conditions (loss, delay etc.). Again the overhead needed for the maintenance of the network graph is linear with the degree of each node. The theoretic findings are validated implementing the strategies devised in Sec. V into the PeerStreamer platform [9], and comparing them with the standard overlay management strategy available in PeerStreamer and used as reference for several works like [10], [11].

### III. RELATED WORKS

Cooperative video streaming (including P2P) is an established research area. We focus on unstructured and mesh-based approaches, in which there is no specific structure (like a tree) in the topology. This approach has been shown to be particularly robust even in networks with churn (i.e., peers leaving and joining the swarm).

We do not consider here papers that perform streaming optimization on mesh networks requiring modifications to the lower layers (they cannot be applied to existing CNs) or that are not tailored for live video streaming (e.g., using large chunks that imply several seconds of buffering delay). We also do not consider techniques (e.g., like cloud-assisted or SDN based), where the role of the peers is, in one way or another, not fundamental, and we assume that security [12] and collusion [13] issue need not be solved by the application itself. The following discussion is focused on two parts: *topology management*, which is directly related to our contribution, and *chunk/information scheduling*, which justify the choice of the chunk selection strategy in Sec. VI and Sec. VII.

1) *Topology Management*: As we already mentioned, *overlay* optimization on the Internet is not feasible due to lack of information on the *underlay* details; however several efforts

have been done to adapt and improve the *overlay* topology to some measured *underlay* characteristics.

The first approach to mention is the use of “network coordinates” as a means to compute distances between hosts in a certain space. Several algorithms were proposed [14], [15], [16], that are designed to work in the heterogeneous environment of Internet. In all of them the goal is clearly to find a method to infer details on the *underlay* (the Internet), a problem that we do not have, as we take advantage of the available information on the network topology provided by routing protocols in Wireless Mesh Networks (WMNs). We believe that our solution can be adapted, albeit not straightforwardly, to situations where the *underlay* is not known but approximated with network coordinates.

A second line of research has been concerned with the adaptation of the *overlay* based on bandwidth [17] or delay (normally the round trip time between peers) [18] measures, but also on a mix of the two [19]. The solutions found in these works are, once again, tailored to the Internet, where delays can be large (CNs spans a few tens of hundreds of km at most), and bandwidth asymmetry at the edges impose hard limits to the capacity of peers to contribute to dissemination.

Extremely interesting and promising for topology management is the adoption of centrality metrics as means to better understand the topology characteristics of a network graph as it emerges from the routing protocol. Centrality metrics in graphs have been used in social science since the 70s to identify the most influential elements in social networks. Quite surprisingly, they were not applied to multi-hop networks up to recent times [4]. Centrality metrics can be used to enhance network monitoring and routing [20], intrusion detection and firewalling [21], [22], and topology control [23]. There are several metrics based on different centrality “concepts”. In this paper we use the betweenness centrality (see [22] for a definition tailored to our problem) to relax the optimization problem as it is strictly related to shortest path routing.

2) *Chunk Scheduling*: In a mesh-based *overlay* the problem of chunk scheduling is the selection of the neighbors to send/receive information to/from while contextually choosing the right information (chunk) to send/retrieve. This problem has been extensively studied [24], [25], [26], and in some specific contexts with restrictive assumptions, the existence of an optimal scheduling strategy has been proven [27].

Those works show that an efficient and robust chunk scheduling technique that works well in most environments consists in selecting a neighbor with a random (possibly weighted) strategy, and push the most recent chunk that is still missing at the receiving peer (Latest Useful Chunk). This is the standard methodology used in PeerStreamer and that we use to benchmark our proposal.

### IV. OVERLAY MODEL

The links in  $U$  are bidirectional and assumed quasi-symmetric: the most common routing protocols take care of excluding unidirectional or highly asymmetric links. Thus both  $O$  and  $U$  are undirected, and the maximum number of edges they can have is  $m_O = \frac{|P|^2 - |P|}{2}$  and  $m_U = \frac{|H|^2 - |H|}{2}$  respectively.

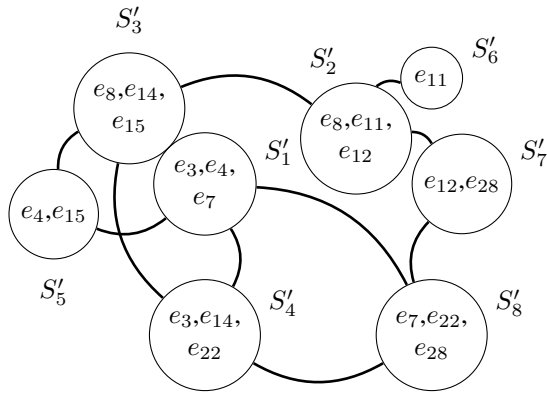


Fig. 3: Example of overlay intersection graph. Elements  $e_k$  are the cross-layer overlay edge descriptors between the nodes.

Let  $r \in 1 \dots m_U$  be an arbitrary ordering on the links;  $r$  is also a mapping from the two hosts  $h_i$  and  $h_j$  that are the endpoints of the link:  $r(h_i, h_j)$ . Similarly we define an ordering  $k \in 1 \dots m_O$  on the edges of  $O$ , and  $k$  is a mapping  $k(p_i, p_j)$ .

Every link  $l_r$  is represented by a binary array of size  $m_U$  with the  $r$ th element set to one and all other elements set to zero (we use the bar sign to refer to the array representation of a link):

$$\bar{l}_r = (0, \dots, 0, 1, 0, \dots, 0)$$

Equivalently each virtual link  $e_k$  is represented by the sum of the link arrays in the corresponding shortest path:

$$\bar{e}_k = \sum_{l_r \in \mathbf{D}(e_k)} \bar{l}_r$$

We call  $\bar{e}_k$  the cross-layer overlay edge descriptor, since it connects the overlay with the underlay. Link weights  $w(l)$  in  $U$  can be easily taken into account: let  $W \in \mathbb{R}^{m_U \times m_U}$  be a diagonal matrix such that  $W_{r,r} = w(l_r)$ , then  $\bar{e}_k W$  is the weighted representation of the overlay edge.

#### A. Overlay re-definition as an intersection graph

Given all  $\bar{e}_k$  for a set of peers  $P$ , we can take advantage from a transformation into the intersection graph space (see [28] for the complete definitions and properties of intersection graphs) in order to formulate our optimization problem.

Let  $S$  be a set and  $F = \{S_1, \dots, S_p\}$  a nonempty family of distinct nonempty subsets of  $S$  whose union is  $S$ . The intersection graph of  $F$  is denoted  $\Omega(F)$ , with  $S_i$  and  $S_j$  adjacent whenever  $i \neq j$  and  $S_i \cap S_j \neq \emptyset$ . It is easily shown that if  $O(P, E)$  is a full mesh then it is isomorphic with the intersection graph space  $\Omega(F)$  where

$$S_i = \{\bar{e}_{k(p_i, p_j)}, \forall p_j \in P\} \quad \forall p_i \in P; \quad S = \cup_{i=1}^{|P|} S_i \quad (4)$$

and each  $S_i$  is the set of all the possible virtual links built on shortest paths from peer  $p_i$  to all other peers. As a consequence, given an underlay  $U(H, L)$  and a set of peers  $P$ , any overlay  $O(P, E)$  over  $U(H, L)$  can be defined as the intersection graph  $\Omega(F')$  with  $F' = \{S'_1, \dots, S'_{|P|}\}$  where

$S'_i \subseteq S_i \quad \forall S_i \in F$ . If each peer chooses only a subset  $S'_i \subseteq S_i$  and activates only a subset of possible edges, then the resulting overlay is isomorphic with some  $\Omega(F')$ . The overlay depicted in Fig. 2 is isomorphic with the intersection graph shown in Fig. 3: every  $S'_i$  in Fig. 3 corresponds to a peer  $p_i$  in Fig. 2.

#### B. Performance Measures

We can now redefine in terms of intersection graphs also the performance metrics Eq. (1) and Eq. (3) defined in Sec. II:

$$\mathcal{L} = O_I(\Omega(F')) = \bar{\mathbf{1}} \cdot L(F'); \quad L(F') = \sum_{\bar{e}_k \in S'} \bar{e}_k W \quad (5)$$

where  $L(F')$  is the array that associates the traffic potentially produced by the overlay to each link in the underlay,  $\bar{\mathbf{1}}$  is the array of size  $m_U$  made of all ones, and  $\cdot$  is the dot product. Eq. (5) redefines Eq. (1) through operations done in the intersection graph space. Similarly we re-define the Jain's fairness index as

$$\mathcal{F} = O_f(\Omega(F')) = \frac{(\sum_{k=1}^{m_U} L(F')_i)^2}{m_U (\sum_{k=1}^{m_U} L(F')_i^2)} \quad (6)$$

#### V. OVERLAY OPTIMIZATION

For the sake of simplicity, but without loss of generality we take as weighting matrix  $W$  the identity matrix. Let's say we want to build an overlay  $O$  determined by a choice of  $F' = \{S'_1, \dots, S'_{|P|}\}$  that minimizes the load on the underlying edges and guarantees a fairness as close as possible to 1. We have to choose the sets  $S'_i \subseteq S_i$  with  $S' = \cup_{i=1}^{|P|} S'_i$  such that both  $O_I$  and  $O_f$  are minimal, which is a multi-objective combinatorial optimization problem. The problem allows the definition of a combined metric  $O_c$  that expresses the *cost* of the overlay. Since each array  $\bar{e}_k \in S'$  corresponds to a set of links the cost of the overlay is defined as:

$$O_c(\Omega(F')) = \left\| \sum_{k=1}^{|S'|} \bar{e}_k \right\|_2$$

The creation of an efficient overlay  $\Omega(F')$  can now be formulated as a minimization problem as follows. Select  $F'$  in order to minimize the expression:

$$\arg \min_z \left\| \sum_{k=1}^{|S|} z_k \bar{e}_k \right\|_2 \quad (7)$$

where

$$z_k = \begin{cases} 1 & \text{if } \bar{e}_k \in S' \\ 0 & \text{otherwise} \end{cases}$$

In order to avoid a trivial solution we impose on each peer a minimum node degree  $d > \log_2(|P|)$ , which also guarantees that the resulting overlay is connected with high probability<sup>2</sup>.

This problem can be rephrased as: find the overlay graph  $\Omega(F')$  with minimum degree  $d$  defined by  $F' = \{S'_1, \dots, S'_{|P|}\}$

<sup>2</sup>Using simple P2P techniques the probability actually converges to 1 [19].

so that the norm of  $L(F')$  is minimized subject to the following constraint

$$\left( \sum_{\bar{e}_k \in S_i} z_k \right) \geq d; \quad \forall i = 1 \dots |E| \quad (8)$$

In Eq. (8) we did a small abuse of notation to improve the readability: the sum spans all the edges  $e_k \in S_i$ , but it is indeed a sum over  $k(p_i, p_j)$  to correctly identify the indication function  $z_k$ . We will use this notation also in several other equations.

If, instead of the norm, in Eq. (7) we use the sum of the elements, this would simply minimize  $O_I$ . The norm instead prefers solutions that are close to the minimum  $O_I$  and, among two potential solutions with the same  $O_I$ , it prefers the one in which the weights of  $L(F')$  are more fairly distributed.

This problem is a zero-one quadratic programming problem [29], similar to a quadratic knapsack problem [8]. In this kind of problems one wants to minimize the value of an expression  $c^T x + x^T Q x$  where  $x = \{0, 1\}^n$  is an array of binary variables,  $c \in R^n$  and  $Q$  is a symmetric matrix of size  $n \times n$ . The minimization is subject to a constraint of the kind  $h^T x + x^T G x > g$  where  $h$  is an array of size  $n$ ,  $G$  a symmetric matrix of size  $n \times n$  and  $g$  some real value. If we call  $\bar{A}$  the matrix made of columns corresponding to the arrays  $\bar{e}_k$  and  $z$  the array made of  $z_k$  elements then:

$$\arg \min_z \left\| \sum_{k=1}^{|S|} z_k \bar{e}_k \right\|_2 = \arg \min_z z^T \bar{A}^T \bar{A} z$$

Our problem is thus a zero-one quadratic problem with  $Q = \bar{A}^T \bar{A}$ ,  $G$  and  $c$  made of all zeros,  $h = \bar{1}$ . This family of problems is known to be NP-hard, but there are algorithms in literature that make them tractable up to a certain size using branch-and-bound techniques. Still, when  $|S|$  grows beyond a few hundreds the problem can not be solved on commodity hardware.  $|S|$  corresponds to the number of possible edges in the overlay  $m_O$ , so it scales quadratically with the number of peers, which quickly makes the problem intractable.

#### A. Betweenness Centrality-based Relaxation

We need to find a relaxation in which each  $p_j$  can solve a portion of the problem, making some assumptions on the behaviour of the other peers. This corresponds to a scenario in which every peer is aware of the other peers, independently selects its own neighbors, and it communicates them its choice. This is the way P2P streaming protocols based on peer sampling typically work (including PeerStreamer).

Let us first separate the contribution to the overall cost of the edges chosen by  $p_j$  and all the other peers in Eq. (7)

$$\left\| \sum_{k=1}^{|S|} z_k \bar{e}_k \right\|_2 = \frac{1}{2} \left\| \sum_{S_i \in F, S_i \neq S_j} \sum_{\bar{e}_k \in S_i} z_k \bar{e}_k + \sum_{\bar{e}_k \in S_j} z_k \bar{e}_k \right\|_2 \quad (9)$$

The value  $\frac{1}{2}$  comes from the observation that when we separately count each link, every link is counted twice in the

sum. Let's call  $b_j$  the vector representing the choices of the peers in  $P \setminus \{p_j\}$ :

$$\left\| \sum_{k=1}^{|S|} z_k \bar{e}_k \right\|_2 = \frac{1}{2} \left\| b_j + \sum_{\bar{e}_k \in S_j} z_k \bar{e}_k \right\|_2 \quad (10)$$

and we can say that:

$$\arg \min_z \left\| \sum_{k=1}^{|S|} z_k \bar{e}_k \right\|_2 = \arg \min_z \left\| b_j + \sum_{\bar{e}_k \in S_j} z_k \bar{e}_k \right\|_2 \quad (11)$$

Our goal is now to find a relaxation of the problem in which every peer chooses its own neighborhood making some assumptions about  $b_j$ , which represents the choice that the other peers  $p_i \neq p_j$  do. We need to find a reasonable approximation  $\bar{b} \simeq b_j$  that node  $p_j$  can use in (11).

Let us now introduce a notion that helps us in this task. In graph theory, the notion of *betweenness centrality* is a property of the edges (or nodes) of a graph defined as the fraction of the total number of shortest paths that passes through that edge (or node). It is a metric used to identify the edges (or nodes) that are more involved in multi-hop interactions between the vertexes of a graph, so for some applications they can be considered more important than the others.

We call  $b = \sum_{\bar{e}_k \in S} \bar{e}_k$  the summation of all the cross-layer overlay edge descriptors of the complete overlay. Recall that  $\bar{e}_k$  corresponds to a shortest path in  $U$  between two hosts on which a peer resides, thus, each element of  $b$  corresponds to a link in  $U$  and expresses the number of shortest paths in the set  $S$  that insist on that link.

Consider the limit case in which every host in the underlay contains a peer ( $|P| = |H|$ ) and let  $b^*$  be the value of  $b$  normalized to the total number of shortest paths:

$$b^* = b \frac{2}{|P|^2 - |P|}$$

$b^*$  is exactly the vector corresponding to the betweenness centrality of each link in  $L$ .

If  $|H| > |P|$ ,  $b^*$  is an approximation of the real array of centralities. This is a known fact that is used to approximate centrality in large networks: if the number of nodes is too large to compute all the shortest paths, centrality can be estimated using a subset of the paths chosen from a random set of nodes [30]. A key fact is that the convergence to a solution close to the real one is pretty fast in power-law graphs, that are extremely frequent in real communication networks, and also in some large CNs [1]. Thus, even if  $p_j$  ignores  $b_j$  a reasonable assumption is that whatever the choice of each other peer is, the elements of  $b_j$  (that represent the sum of all choices) have a shape similar to the centrality expressed by the normalized value  $b^*$ . This, on power-law underlays is true even for values of  $|P|$  one order of magnitude smaller than  $|H|$ .

Given that we impose each peer to have (at least)  $d$  neighbors, the number of edges in  $E$  will be approximately  $\frac{d|P|}{2}$  and finally the best approximation for  $b_j$  turns to be

$$\bar{b} = b^* (|P| - 1) \frac{d}{2} = b \frac{d(|P| - 1)}{|P|^2 - |P|} = b \frac{d}{|P|} = \frac{d \sum_{k=1}^{|S|} \bar{e}_k}{|P|} \quad (12)$$

The complexity of Eq. (12) is polynomial with  $P$  [30], that allows the computations of its solution for overlays of hundreds of peers using commodity hardware. Moreover in communication networks, that are sparse graph, betweenness can be computed quickly using heuristics [31], which explains why we introduced this formulation based on centrality. Thus, replacing  $b_j$  with  $\bar{b}$  in (11) each peer  $p_j$  resolves the following optimization problem:

$$\arg \min_z \left\| \bar{b} + \sum_{\bar{e}_k \in S_j} z_k \bar{e}_k \right\|_2 \quad (13)$$

$$\text{conditioned to: } \sum_{\bar{e}_k \in S_j} z_k \geq d \quad (14)$$

The formulation of (13) is another zero-one quadratic minimization problem, but the dimension of the problem is now bounded by  $|P| < |S|$  (the maximum number of neighbors for  $p_j$ ), and can be effectively solved up to hundreds of peers. In the rest of the paper we will use the branch-and-bound solver given by the YAMMLIP library [32] which solves the problem (13) for a network of 100 nodes in few seconds.

If still the dimension of the problem or the available hardware do not allow the solution of the optimization, we can apply a greedy search algorithm, ranking each possible  $\bar{e}_k$  for its weight and choosing the ones that minimize the sum. This corresponds to relax (13) to:

$$\arg \min_z \sum_{\bar{e}_k \in S_j} z_k \left\| \bar{b} + \bar{e}_k \right\|_2 \quad (15)$$

$$\text{conditioned to: } \sum_{\bar{e}_k \in S_j} z_k \geq d \quad (16)$$

which of course captures only a part of the original problem but greatly simplifies the computation. Our results show that the solutions generated by (7), (13) and (15) in the case of realistic network topologies are reasonably close one another.

It is worth nothing that if we set  $b = \bar{0}$  in Eq. (15) then the l-2 norm respects the order of a l-1 norm. In practice each  $p_j$  chooses the neighbors that are close in terms of hops in the underlay. The  $b$  terms instead introduces a bias in the choice towards the neighbors connected through links that are less overloaded and introduces a higher fairness. In the comparison we include also a strategy in which  $b = 0$  because this strategy can be used also in absence of full information from the underlay topology, since the distance from another peer can be measured with probing tool (like the `traceroute` application) or can be inferred by the time-to-live field in IP packets. Results show that the performance of this simple ranking function is sensibly lower compared to the proposed strategies.

Tab. II summarises the different optimization strategies and labels them with names used in the rest of the papers.

## VI. EXPERIMENTAL SET-UP

We evaluated the proposed strategies with two different approaches, first, we implemented them in a simulator that computes the best overlay according to each strategy on

Name	Symbol	Formula
Global Optimization	$G_o$	$\arg \min_z \left\  \sum_{k=1}^{ S } z_k \bar{e}_k \right\ _2$
Local Optimization	$L_o$	$\arg \min_z \left\  \bar{b} + \sum_{k=1}^{ S_j } z_k \bar{e}_k \right\ _2$
Local Equalized Ranking	$E_r$	$\arg \min_z \sum_{k=1}^{ S_j } z_k \left\  \bar{b} + \bar{e}_k \right\ _2$
Local Ranking	$L_r$	$\arg \min_z \sum_{k=1}^{ S_j } z_k \left\  \bar{e}_k \right\ _2$

TABLE II: A summary of the optimization functions

synthetic network topologies, then we run a modified version of PeerStreamer in an emulated network topology derived from previous studies of real mesh networks.

### A. Simulations

Simulations have been performed using the Networkx library, a powerful library for the generation and analysis of graphs realized in the Python language. Given an underlay topology we have implemented the proposed strategies and for each one we compared the measures of load and fairness of the generated overlay graph. The global and the local optimization problems are solved with the YAMMLIP library, while the others have been implemented directly in Python.

### B. Emulations

All the code used for emulations is open-source and is freely available on-line<sup>3</sup>. For our emulation experiments we take advantage from real-life WCN topologies, taken from the Ninux and the FFWien<sup>4</sup> networks [33] respectively made of 131 and 236 nodes.

Emulations are based on a modified version of Mininet, a lightweight emulator for arbitrary network topologies [34]. We use the ETX information from the topology dataset to evaluate the link loss during experiments. We consider a link delay uniformly distributed in  $[30, 1000] \mu\text{s}$  and a constant link bandwidth of 10 Mbit/s. This is a reasonable assumption since WCN links can typically provide even more bandwidth, both in uplink and downlink. The sample video we use in our experiments is a re-encoding at bit rate of  $300 \text{ kbit/s}^{-1}$  (including both audio and video) of Big Buck Bunny<sup>5</sup>. Currently, hardware constraints limit the number of overlay nodes we can emulate, in our experiments we setup overlays of 30 peers with a minimum node degree of 10.

## VII. RESULTS

Figs. 4 and 5 report the comparison of all the described strategies in a small (20 peers) scenario, increasing the size of the underlay. With 20 peers we are able to solve all the optimization problems, so this is a good benchmark to outline the differences between each strategy. Figs. 4a and 4b report the differences in the strategies measured using an Erdős-Rényi (ER) underlay of increasing size, while Figs. 5a and 5b report the same values computed on a Barabási-Albert (BA) graph of the same size. In all the cases, the optimized results

<sup>3</sup><https://ans.disi.unitn.it/redmine/projects/peerstreamer>

<sup>4</sup>See <http://ninux.org> and <http://www.funkfeuer.at/>

<sup>5</sup><http://www.bigbuckbunny.org>

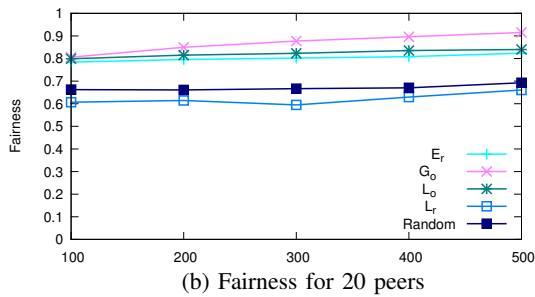
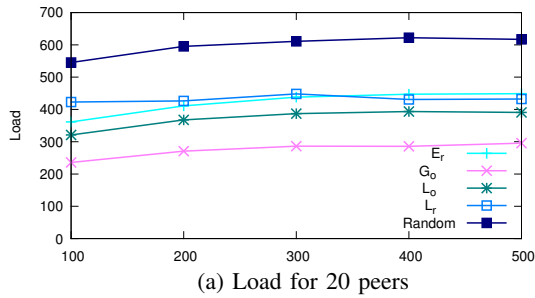


Fig. 4: Load and fairness for an ER underlay from 100 to 500 hosts and 20 peers

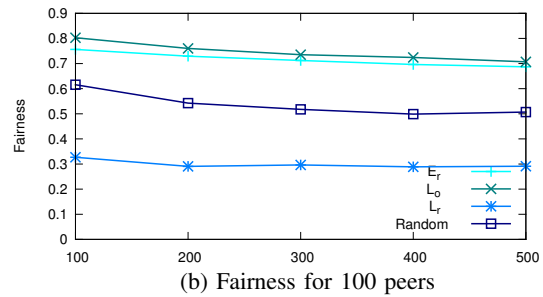
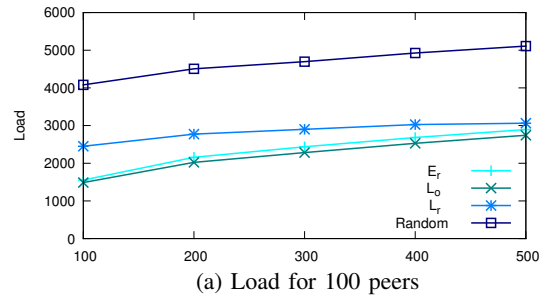


Fig. 6: Load and fairness for a BA underlay from 100 to 500 hosts and 100 peers

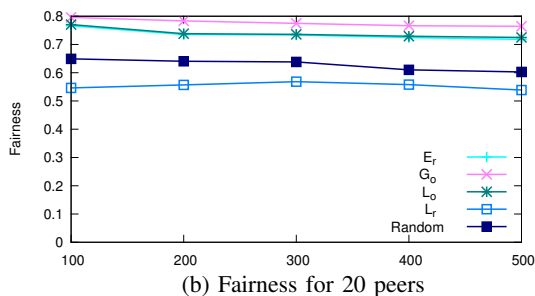
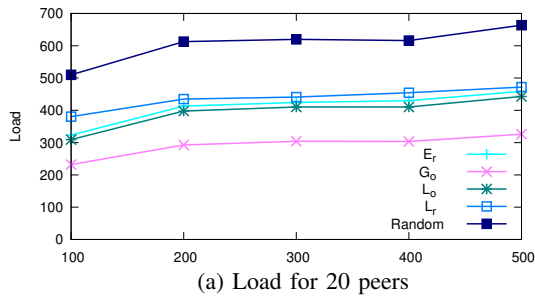


Fig. 5: Load and fairness for a BA underlay from 100 to 500 hosts and 20 peers

vastly outperform the random strategy, which is not surprising since they use available information on the underlay. What is most significant is that even the strategies that are less costly to compute, namely  $L_o$  and  $E_r$ , achieve results that are very close to the optimal strategy  $G_o$  especially in terms of fairness. This is true for both the topology types chosen, and more evident for the BA graphs.

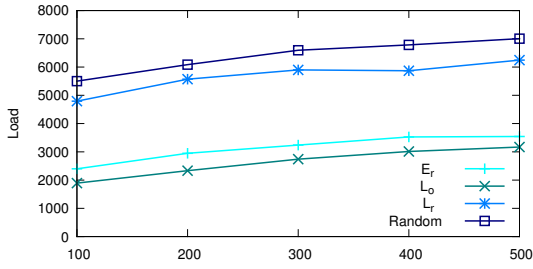
This means that even if the quadratic optimization remains NP, in the analysed graphs (especially for BA graph) the

number of available disjoint paths between two peers is low and the space of the solutions of the optimization problem is small enough for all the optimization strategies to be very close. This is a key observation since many works in literature show that real communication networks present a scale-free topology as the BA algorithm produce.

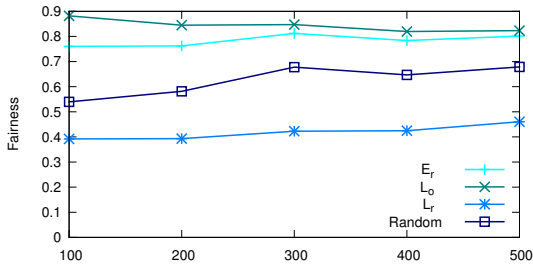
In the next set of results we evaluate the strategies using two different topology generator, the already mentioned BA and the generator proposed by Cerdá-Alabern (CE) in [1], derived from the analysis of a number of real mesh networks. The CE algorithm uses a preferential attachment algorithm to create a core of interconnected hosts, and then adds leaf hosts using a Gamma distribution. We performed experiments with overlay size up to 100 peers, since there is no qualitative difference in the results we report only the results for 100 peers. With that size we are not able to use the  $G_o$  strategy, so the comparison is done only with the remaining strategies.

Figures 6 and 7 report the results on a 100 peers overlay and show that in both the considered topologies the  $L_o$  and the  $E_r$  strategies are very close and achieve a substantial improvement compared to the random strategy.

It is interesting to note that the random strategy generates a distribution of the traffic in the underlay in which the links with a higher centrality have a higher load. In practice, the value of fairness computed on the random overlay mirrors the fairness computed on the  $b^*$  array for the underlay. Only the  $L_r$  strategy produces a lower fairness compared to the random strategy, since  $L_r$  simply chooses the neighbors for peer  $p_j$  among the closest ones. It thus reduces the overall load (since the average distance is decreased) but it prefers links that are highly central, and peers that are highly central, thus decreasing the measure of fairness. This observation is important to understand that to achieve a fair distribution of



(a) Load for 100 peers



(b) Fairness for 100 peers

Fig. 7: Load and fairness for a CE underlay from 100 to 500 hosts and 100 peers

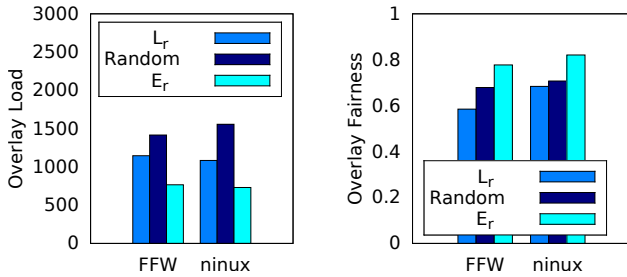


Fig. 8: Load and fairness of the overlay graph computed on FFWien and ninux topologies

the resources it is necessary to have at least some information on the underlay, or else the local-only optimization performed by  $L_r$  leads to a global choice that is largely sub-optimal.

### A. Results on Real Topologies

To further corroborate our results, first we tested the proposed strategies on two topologies that have been extracted from real mesh networks, then, on the same topologies we run a modified instance of PeerStreamer that implements the  $E_r$  and  $L_r$  strategies using the emulation environment. The goal of this subsection is to show that the real implementation has even better performance compared to the simulated algorithm, that the  $E_r$  strategy can be easily implemented and that it guarantees a timely delivery of chunks in a realistic wireless mesh network scenario.

Fig. 8 shows the load and fairness on the overlay of the ninux and FFWien networks, obtained with simulations. The results are perfectly compatible with the ones we have described so far. Moreover, they are perfectly compatible with Fig. 9 that reports the corresponding values measured with the

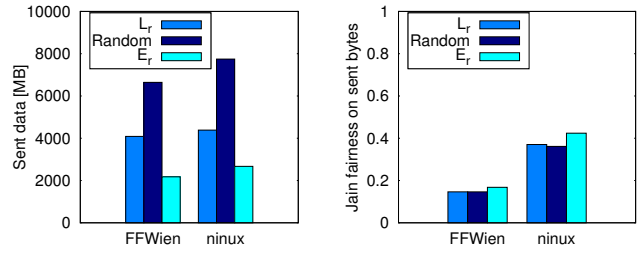


Fig. 9: Sent data and data fairness of the overlay graph measured with PeerStreamer on real topologies

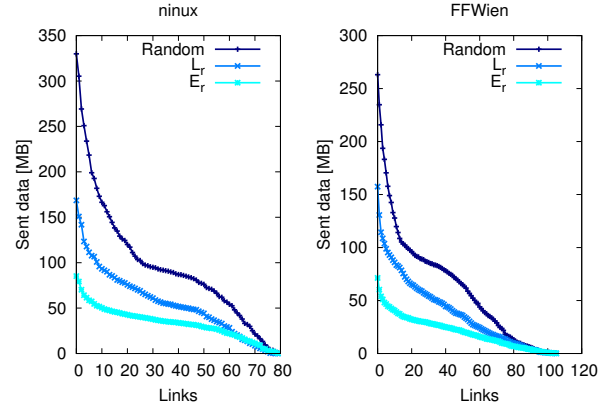


Fig. 10: Load measured on each link on real topologies

PeerStreamer emulation. In this case the occupation of each link has been measured in terms of number of video bytes transmitted on each underlay link. PeerStreamer includes a number of mechanisms that limit the traffic per link, while in the simulations we considered that each link would have carried an unit of traffic. For this reason the absolute values are not comparable, indeed the relative values show an extremely similar behaviour, especially in the load measure. The distribution process operated by PeerStreamer is extremely dynamic and driven both by its algorithms and by random choices. This very dynamic behaviour leads to an hardly predictable link resource usage and it may reduce the absolute value of the fairness.

The graph in Figure 10 show the number of bytes sent on each link of the underlay measured on all the emulation runs. Links are ordered on the  $x$  axis for the measured load, reported on the  $y$  axis. The graph confirms the decreased total traffic, it shows that  $L_r$  and  $E_r$  have a small deviation from their average value (which confirms the fairness measure) but it also shows that the peak of the measured traffic is strongly reduced. This is a fundamental feature that shows that a fair distribution of resources can effectively prevent bottlenecks and saturation on the most central links.

Finally, Tab. III reports the packet loss in the emulated network. We do consider successfully delivered only chunks that arrive before 1 second after their generation, chunks arrived with a higher delay would not be useful in a real-time live streaming. It is evident that the proposed strategies do not significantly impact the delivery of packets, which is



Network	Strategy	Loss (%)
FFWien	$L_r$	4.8
FFWien	$E_r$	4.9
FFWien	Random	3.8
ninux	$L_r$	1.5
ninux	$E_r$	1.6
ninux	Random	1.6

TABLE III: Packet loss measured via emulation

in both networks higher than 95%. Ninux has lower losses since it has a higher average link quality[33]. Note that with a lower delivery rate it would be necessary to assess the received video with specific video-quality metrics, but with less than 5% or even 2% loss, we do not consider necessary to apply such metrics, and we can focus only on networking issues.

### VIII. CONCLUSIONS

WMNs are an integral part of the present and future Internet access, as they provide large capacity with flexible allocation and mobility. Thus, video streaming in WMNs with a small footprint on the precious wireless resources is of paramount importance. Starting from this observation we designed a novel strategy for cooperative (live) video streaming on distributed networks which can be successfully deployed on a WMN, or any other multi-hop network that provides to the network nodes a complete view of the topology.

Exploiting a novel mapping of the problem onto intersection graphs, we formulated an optimization problem to build an overlay that not only reduces the total load on the underlay, but also increases the fairness in the distribution of the load on the underlay links. This problem in its general formulation is NP, so we proposed two relaxations based on betweenness centrality.

We applied the proposed technique to video streaming on the PeerStreamer platform, but we believe it can be applied to any similar problem in which an overlay must be optimized to efficiently use the resources of the underlay network. Moreover, recently introduced heuristics make centrality metrics fast to be computed even on thousands of nodes so our proposal can be extended to networks and applications with a larger number of peers than what we analysed in the paper.

### REFERENCES

- [1] L. Cerda-Alabern, "On the topology characterization of Guifi.net," in *IEEE 8th Int. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2012, pp. 389–396.
- [2] L. Maccari et al., "Live Video Streaming for Community Networks, Experimenting with PeerStreamer on the Ninux Community," in *ACM DIYNetworking '15*, Florence, IT, May 22, 2015, pp. 1–6.
- [3] L. Baldesi, L. Maccari, and R. Lo Cigno, "Improving P2P streaming in Wireless Community Networks," *Computer Networks*, vol. 93, Part 2, pp. 389–403, 2015.
- [4] D. Katsaros, N. Dimokas, and L. Tassioulas, "Social network analysis concepts in the design of wireless ad hoc network protocols," *IEEE Network*, vol. 24, no. 6, pp. 23–29, Dec. 2010.
- [5] T. Clausen and P. Jaquet, "Optimized link state routing protocol (olsr)," Internet Requests for Comments, RFC 3626, Oct. 2003.
- [6] B. Braem et al., "A Case for Research with and on Community Networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, pp. 68–73, Jul. 2013.
- [7] M. Campista et al., "Routing metrics and protocols for wireless mesh networks," *IEEE Network*, vol. 22, no. 1, pp. 6–12, 2008.

- [8] D. Pisinger, "The quadratic knapsack problem—a survey," *Discrete Applied Mathematics, Elsevier*, vol. 155, no. 5, pp. 623–648, Mar. 2007.
- [9] R. Birke et al., "Architecture of a Network-Aware P2P-TV Application: The NAPA-WINE Approach," *IEEE Communications Magazine*, vol. 49, pp. 154–163, June 2011.
- [10] S. Traverso et al., "Experimental comparison of neighborhood filtering strategies in unstructured P2P-TV systems," in *12th IEEE Int. Conf. on Peer-to-Peer Computing (P2P-12)*, Tarragona, Spain, Sept. 2012, pp. 12–24.
- [11] L. Abeni, C. Kiraly, and R. Lo Cigno, "Robust Scheduling of Video Streams in Network-Aware P2P Applications," in *IEEE ICC 2010*, Cape Town, ZA, May 2010, pp. 1–5.
- [12] G. Gheorghie, A. Montresor, and R. Lo Cigno, "Security and Privacy Issues in P2P Streaming Systems: A Survey," *Springer Peer-to-Peer Networking and Applications*, vol. 4, pp. 75–91.
- [13] G. Ciccarelli and R. Lo Cigno, "Collusion in Peer-to-Peer Systems," *Elsevier Comput. Netw.*, vol. 55, no. 15, pp. 3517–3532, Oct. 2011.
- [14] T. E. Ng and H. Zhang, "Predicting internet network distance with coordinates-based approaches," in *IEEE INFOCOM 2002*, New York, NY, USA, June 2002, pp. 170–179.
- [15] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," in *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, 2004, pp. 15–26.
- [16] Y. Chen, X. Wang, C. Shi, E. K. Lua, X. Fu, B. Deng, and X. Li, "Phoenix: A weight-based network coordinate system using matrix factorization," *Network and Service Management, IEEE Trans. on*, vol. 8, no. 4, pp. 334–347, 2011.
- [17] R. Fortuna et al., "QoE in Pull Based P2P-TV Systems: Overlay Topology Design Tradeoffs," in *10th IEEE Int. Conf. on Peer-to-Peer Computing (P2P-10)*, Delft, NL, Aug. 2010, pp. 1–10.
- [18] A. Russo and R. Lo Cigno, "Delay-Aware Push/Pull Protocols for Live Video Streaming in P2P Systems," in *IEEE ICC 2010*, Cape Town, ZA, May 2010, pp. 1–5.
- [19] S. Traverso et al., "Neighborhood Filtering Strategies for Overlay Construction in P2P-TV Systems: Design and Experimental Comparison," *IEEE/ACM Trans. on Networking*, vol. 23, no. 3, pp. 741–754, June 2015.
- [20] S. Dolev, Y. Elovici, and R. Puzis, "Routing betweenness centrality," *J. ACM*, vol. 57, no. 4, pp. 25:1–25:27, May 2010.
- [21] L. Maccari and R. Lo Cigno, "Waterwall: a cooperative, distributed firewall for wireless mesh networks," *EURASIP Jou. on Wireless Communications and Networking*, vol. 2013, no. 1, pp. 1–12.
- [22] L. Maccari and R. Lo Cigno, "Betweenness estimation in OLSR-based multi-hop networks for distributed filtering," *Elsevier Jou. of Computer and System Sciences*, vol. 80, no. 3, pp. 670–685, May, 2014.
- [23] A. Vazquez-Rodas and L. J. de la Cruz Llopis, "A centrality-based topology control protocol for wireless mesh networks," *Ad Hoc Networks*.
- [24] Y. Sakata et al., "A Chunk Scheduling Based on Chunk Diffusion Ratio on P2P Live Streaming," in *IEEE NBiS 2012*, Sept. 2012, pp. 74–81.
- [25] K.-L. Hua et al., "An efficient scheduling algorithm for scalable video streaming over P2P networks," *Elsevier, Computer Networks*, vol. 57, no. 14, pp. 2856–2868, Oct. 2013.
- [26] J. Zhang et al., "Modeling and performance analysis of pull-based live streaming schemes in Peer-to-Peer network," *Elsevier Computer Communications*, vol. 40, pp. 22–32, Mar. 2014.
- [27] L. Abeni, C. Kiraly, and R. Lo Cigno, "On the Optimal Scheduling of Streaming Applications in Unstructured Meshes," in *IFIP Networking*, Aachen, DE, May 2009.
- [28] F. Harary, "Graph theory," 1969.
- [29] H. D. Sherali and J. C. Smith, "An improved linearization strategy for zero-one quadratic programming problems," *Optimization Letters*, vol. 1, no. 1, pp. 33–47, 2007.
- [30] U. Brandes and C. Pich, "Centrality estimation in large networks," 2007.
- [31] R. Puzis, P. Zilberman, Y. Elovici, S. Dolev, and U. Brandes, "Heuristics for Speeding Up Betweenness Centrality Computation," in *Privacy, Security, Risk and Trust (PASSAT), 2012 Int. Conf. on and 2012 Int. Conf. on Social Computing (SocialCom)*, Sep. 2012, pp. 302–311.
- [32] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Computer Aided Control Systems Design, 2004 IEEE Int. Symposium on*. IEEE, 2004, pp. 284–289.
- [33] L. Maccari and R. Lo Cigno, "A week in the life of three large Wireless Community Networks," *Ad Hoc Networks, Elsevier*, vol. 24, Part B, pp. 175–190, Jan. 2015.
- [34] L. Baldesi and L. Maccari, "NePA TesT: Network Protocol and Application Testing Toolchain for Community Networks," in *12th Conf. on Wireless On-demand Network Systems and Services (WONS)*, Cortina d'Ampezzo, IT, Jan. 2016.