Nozzilla: A Peer-to-Peer IPTV Distribution Service for an IMS-based NGN

Alex Bikfalvi*, Jaime García-Reinoso†, Iván Vidal† and Francisco Valera†
*IMDEA Networks
Avda. del Mar Mediterráneo, 22
Leganés - Madrid, Spain
alex.bikfalvi@imdea.org
†University Carlos III of Madrid
Avda. de la Universidad, 30
Leganés - Madrid, Spain
{jgr, ividal, fvalera}@it.uc3m.es

***Abstract*—During the last years next generation networks (NGNs) have gained attention, as an increasing number of network operators moved toward triple-play services: telephony, Internet and IPTV. Unfortunately, from the service provider perspective, IPTV usually comes at a high cost in terms of equipment and/or necessary bandwidth. In this paper we propose Nozzilla, a peer-to-peer like video streaming service for an NGN based on the IP Multimedia Subsystem (IMS), which can be used by third-party IPTV service providers as a cost effective alternative to IP multicast or content distribution networks. The service uses the residential gateways (NGN customer premises equipment) to create a streaming overlay using the remaining or a fixed amount of network bandwidth. Nozzilla takes advantage of the QoS and resource reservation enabled by a NGN to support multiple-description streaming that improves the perceived quality during temporary failures.**

I. Introduction

Over the past decade there has been a clear trend in the policy of telecom operators to converge their legacy networks into a unified platform. This movement comes not only for reasons of cost-efficiency, but also because - as the broadband access has been transformed into a commodity - fewer users choose to contract these classic options and instead they orient themselves to the cheaper market of Internet services. In these cases the operators can lose their infrastructure investments as the bulk of the money flow goes directly to the third party providers.

Because services such as VoIP and IPTV are gradually replacing their legacy counterparts, a lot of effort has been directed into designing and standardizing a Next Generation Network (NGN) that will allow both the network operators and third party providers to compete in the service market. The NGN emerged as an integrated broadband network that could act as a flexible platform for any type of service [1]. The main idea is to decouple the service provision from the transport network through a set of standardized interfaces. One of the most widely used NGN specifications features an IP core called IP Multimedia Subsystem (IMS) [2], developed by 3GPP.

Given the separation between the service and transport layers, resource demanding applications like IPTV require a particular attention. In legacy deployments, where the service provider is the same as the operator and has complete control over the transport network, IP multicast or content distribution networks (CDNs) are the favorite solutions. However, in an NGN the transport provider may be incapable or unwilling to enable such IPTV distribution mechanisms for third-parties. In these circumstances, maintaining the IPTV service comes with a higher cost for the service providers that have to invest into streaming equipment and necessary bandwidth.

To mitigate these shortcomings, our idea is to create a peer-to-peer (P2P) streaming overlay for an IMS-based NGN that is accessible to any third party IPTV provider. This approach is similar to the P2P video streaming solutions that existed in Internet for some time, but is particularly tailored to the structure, features and business model of an NGN. For this, we propose to use capabilities of the NGN residential gateways (RGWs), which enable service providers to install their own client application, giving an appropriate agreement with the transport provider.

Nozzilla uses application level multicast (ALM) in a tree or mesh structure with RGW nodes. Because in ALM peers use uplink bandwidth (from the user to the network) to redistribute the video packets to other peers, Nozzilla can avoid interfering with existing network traffic by using only the available uplink network capacity that can be measured by the RGW (the utilization of downlink bandwidth is mandatory for all RGWs connected to the service). Bandwidth necessary for video streaming can be reserved with different levels of QoS, which is suitable for advanced video coding mechanisms such as *multi-description coding* and *scalable video coding*.

Since Nozzilla is a content distribution service for applications like IPTV, the Nozzilla service provider will have an agreement with the transport provider and will offer this service to IPTV service providers. This approach has several advantages. First, the IPTV providers do not need any direct agreement with the transport provider. Second, because the solution is a software application, the operators do not have to enable undesired functionality such as IP multicast, nor have to invest into an expensive CDN.

Sec. II summarizes some previous work done in the field of P2P video streaming, not particularly used for IMS. Sec. III is an overview of IMS-based NGNs intended to explain the structure and design requirements for Nozzilla. In Sec. IV we present Nozzilla architecture, including a prototype P2P protocol based on SplitStream [3] and in Sec. V we justify the design decision through some preliminary results. Sec. VI concludes, outlining the future work.

II. P2P-based IPTV Related Work

The Internet Protocol built-in multicast mechanism was designed for point-to-multipoint application purposes such as IPTV streaming. However, many network operators are reluctant to support IP multicast in their routers despite the growth of available bandwidth and enhancement of audio-video compression. As a result, P2P solutions have emerged as a promising alternative, where the task of replicating and forwarding the video traffic is transferred to the end-users interconnected by a P2P overlay network via unicast connections.

In the best-effort Internet, the major barrier towards the design and deployment of P2P video systems is the lack of any guarantees regarding the performance and the video quality experienced by the users. Most peers rely on other peers, which can join and leave at any time and have connections between them with characteristics constrained at modest values. Nevertheless, previous research has shown that even in the conditions of the current Internet, P2P video streaming is possible on a large scale, though in some cases the uplink capacity is a limiting factor [4].

Existing proposals use a P2P protocol to create a mesh or a tree between the peers connected to the same video service. Mesh approaches (or *pull*, because peers are responsible for retrieving the content) are somewhat similar to the unstructured P2P overlays based on swarming where every peer obtains the pieces of video that it needs from a set of other peers. Tree streaming mechanisms follow the multicast principle where the peers organize themselves, often using a DHT-based algorithm, in one or more multicast trees. These are also called *push*, because tree nodes automatically forward video packets to their known children.

Existing schemes include: Prime [5] and Bullet [6] that are mesh-based; SplitStream [3] and PeerCast [7] tree-based; mTreebone [8] hybrid. A comparative study [9] showed that in the current Internet the mesh protocols exhibit a superior performance, mainly due to the high churn and to the inflexible nature of push-based solutions.

However, in this paper we argue that hybrid or pure-tree streaming architectures are a viable option for NGNs for several reasons. First, pull algorithms have a higher video streaming delay that is unacceptable in commercial IPTV services, like those expected to be deployed in an NGN. Second, we expect the NGN RGWs to have a much lower churn rate than typical peers in the Internet. Coupled with a fast tree recovery algorithm, we expect to have a better performance than the same tree in Internet environment. Finally, if we take advantage of the enabled QoS and resource reservation features we can optimize the P2P algorithm to search only for those peers that have available resources.

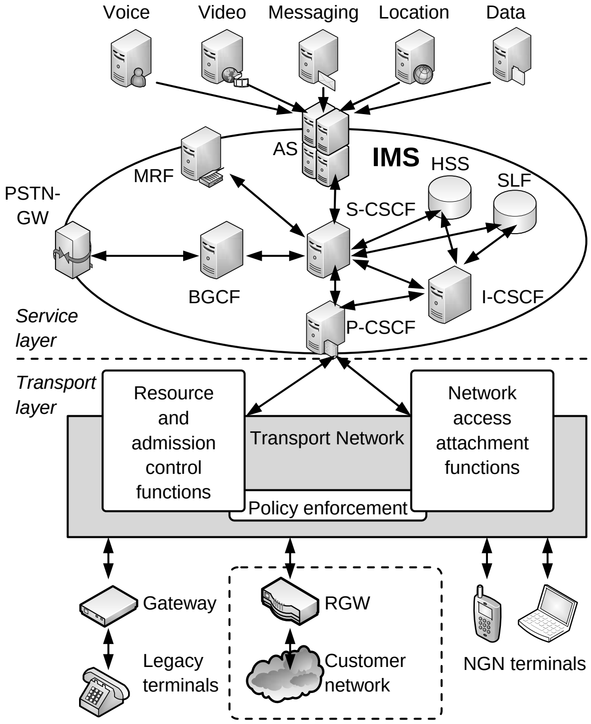


Fig. 1. NGN architecture overview

III. NGNs and IMS

NGNs emerged from the growing need of operators to converge their infrastructure and service offer into a new integrated network. The work done by ITU-T was one among the first initiatives into this direction and materialized in several standards published in 2004: Rec. Y.2001 and Y.2011. In parallel, the Third Generation Partnership Project (3GPP) was working on a similar principle but focused on meeting the immediate market needs. The result was the IP Multimedia Subsystem (IMS), a packet switched environment using Internet protocols that merged the mobile telephony and Internet worlds together into a single packet switched architecture. Following the positive feedback given by the industry and the subsequent standardization effort by ITU-T, IMS is widely seen as a key component of an NGN.

Although evolved from 3GPP second and third generation mobile architectures, IMS features a completely new structure. The design goal was to specify IMS components' functions that interconnect through a set of standardized interfaces. Fig. 1 illustrates the simplified architecture of the IMS (only the most important functions are shown).

Among the most important characteristics of the (IMS-based) NGN is the separation between the service provisioning and transport network and between different control functions: bearer capabilities, call/session and application/service [1]. The NGN requirements include support for IP multimedia sessions, QoS, interworking with the Internet, roaming and service control through operator-imposed policies (global or individual).

Some of the IMS components relevant for us are the CSCFs (Call/Session Control Functions) - SIP servers that process SIP signaling in the IMS. There are three types of CSCF: S-CSCF (Serving-CSCF), P-CSCF (Proxy-CSCF) and I-CSCF (Interrogating-CSCF). The ASs (Applications Servers) are SIP servers hosting services provided to the subscribers. In our current context, any IPTV service will be enabled through a dedicated AS.

In the NGN business model customers benefit from the wider range of services. In general they can maintain a contractual agreement with a *packager* entity, which in turn manages the agreements with the service providers. Service providers establish relationships with the transport provider as needed (although this is not necessary when the packager is the transport provider).

IV. Nozzilla

Nozzilla is a P2P-like distribution service for IPTV in an NGN. Its role is to replace classic video streaming techniques (e.g. IP multicast, CDN) that are not favored by the transport providers. Although it is possible to use Nozzilla with any type of NGN-enabled terminal or legacy gateways, we focus in particular on RGWs because these have the capability of supporting any third party service. Giving an appropriate agreement between the transport provider and a service provider, the latter can install its own software in the RGW in order to support the customer-end of the service.

A. Design Overview and NGN Integration

Nozzilla is a protocol used to create an application-level-multicast (ALM) distribution mechanism between the RGWs in an NGN, and it is integrated into the NGN architecture and business model (see Fig. 2). Usually, the Nozzilla service provider has an agreement with the transport provider to use its infrastructure, while third party IPTV providers will establish agreements with the Nozzilla service provider. From their perspective, Nozzilla represents a virtual CDN: not a physical one but one created by the RGWs interconnected through the overlay.

Nozzilla maintains a *P2P overlay* in order to create a tree or mesh *streaming overlay* between RGWs. This is then used to forward or retrieve (depending on whether the distribution is push or pull) the video stream from the IPTV provider's media servers to all connected RGWs. Nozzilla does not define the streaming protocol that should be used but it enables each RGW connected to the service to locate other nodes (RGWs or IPTV provider servers) from where to retrieve or where to send the live content.

It uses the NGN resource reservation mechanism to determine whether an RGW can download content (i.e. connect to the service) and upload content (i.e. forward the stream to other peers). Resources (bandwidth, delay, jitter, etc.) are reserved whenever a new downstream or upstream flow is created. All RGWs that connect to the IPTV service must meet the minimum requirements for downstream resources. Nozzilla employs a flexible mechanism to compute the available upstream resources, usually depending on the agreement between the Nozzilla provider and the transport provider. Our suggestion is the transport provider assigns only the available resources to Nozzilla that are physically available but are not contracted by the customer. The RGWs of the subscribers that contract a higher percentage of uplink line capacity can use less for video streaming. However, other mechanisms (e.g. assigning a fixed bandwidth amount for each QoS level) are also possible and transport providers can analyze their customer agreements and usage patterns in deciding which resource allocation mechanism to use.

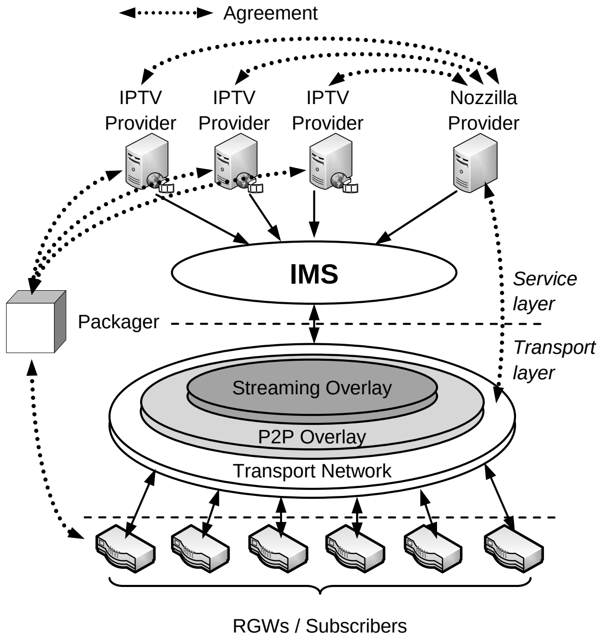


Fig. 2. Nozzilla service in a NGN

Because Nozzilla is QoS-aware, an RGW is selected as source of the video stream, only if a flow with a particular QoS can be reserved. The P2P protocol takes this into account when deciding which RGWs are advertised as possible stream sources, in order to minimize the number of requests sent to peers that either do not have the bandwidth or cannot meet the QoS requirements. This feature enables the use of advanced video coding techniques such as *multi-description coding* and *scalable video coding*, in which the video stream is divided into several sub-streams (called *descriptions* or *stripes*). The descriptions that encode the essential information to play the video can have more stringent QoS requirements compared to descriptions that only enhance the image quality.

B. Service Integration and Session Establishment

The service integration into the NGN involves two phases. First, the Nozzilla and the IPTV providers install their software in the RGWs through the agreement with the transport

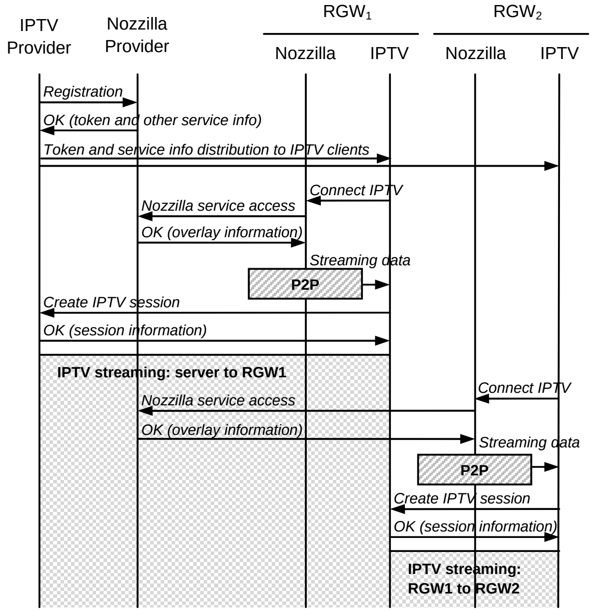


Fig. 3. Service integration and session establishment in a successful scenario

provider. The interaction between the software components is done through common agreed software APIs. Second, the IPTV provider registers its service information (media servers' addresses, necessary bandwidth and QoS, number of descriptions, etc.) with the Nozzilla provider (see Fig. 3). This information is used to create the appropriate P2P and streaming overlays. If the registration is successful, the IPTV provider receives an *access token* used by its software in RGWs to access the *streaming data*. Other exchanged information may include the addresses of the first tier peers (for push streaming), statistics, etc. Although not illustrated in the figure, this signaling is usually done through IMS.

The IPTV provider distributes to RGW clients the information about its service (e.g. available channels, current program, etc.) and Nozzilla access information (e.g. the access token). The Nozzilla client software that joins the P2P overlay when the RGW is powered on. Later, when the subscriber connects to the IPTV channel, it will use overlay specific operations to determine the source(s) for the video stream. Ideally, the P2P overlay should select other peers having enough uplink resources to support the streaming. When the transport provider chooses to allow access to its network topology, Nozzilla can use a topology-aware P2P protocol to make a better decision by selecting nearby RGWs, whenever possible.

After selecting the streaming sources, Nozzilla will give this information as *streaming data* to the IPTV client software which will create end-to-end IPTV sessions to the upstream peers. In Fig. 3, where a single description stream is used, RGW_1 chooses the IPTV streaming server as streaming source. In turn, RGW_2 selects RGW_1 as streaming source, assuming RGW_1 has enough resources to support the upstream video connection.

C. Prototype of a Peer-to-Peer Protocol

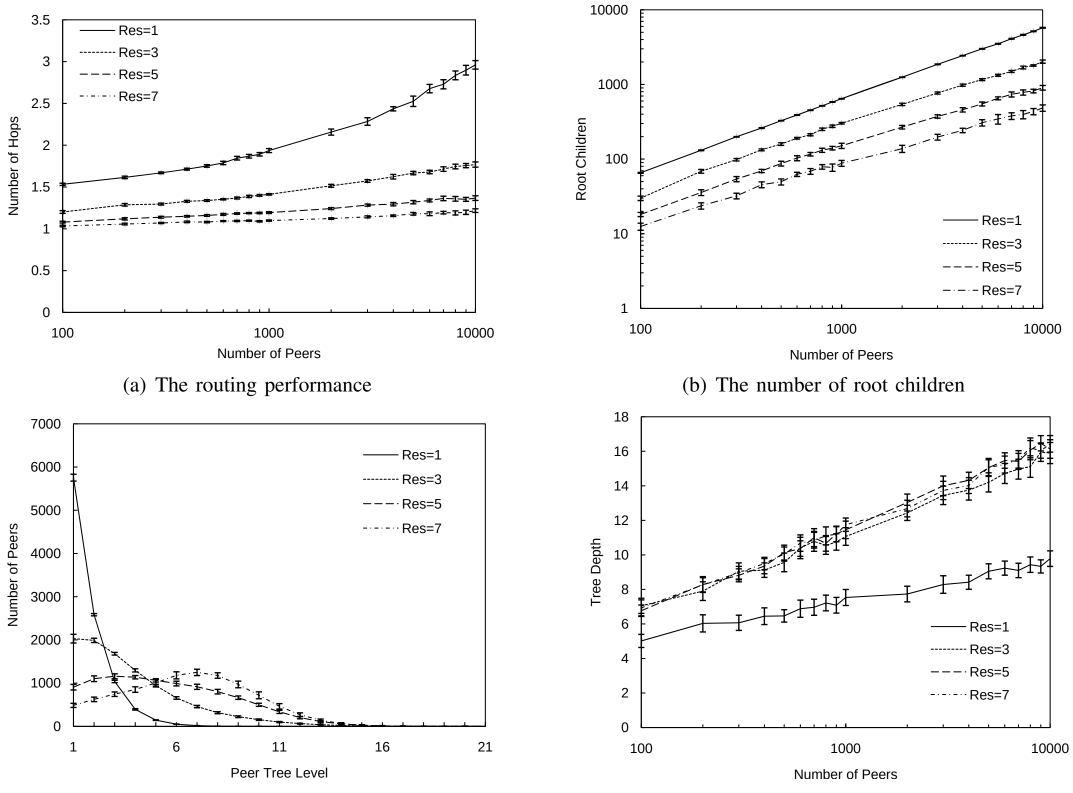
In this section we describe an overview of a prototype P2P protocol for Nozzilla (we omit the details for simplicity). Its objective is to achieve a multi-description video delivery mechanism where an RGW (or *peer*) can use the available uplink resources measured with respect to the contracted bandwidth. For simplicity purposes, we assume that Nozzilla uses three streaming descriptions: one with high priority (HP), one with medium priority (MP) and one with low priority (LP).

Our protocol is an extended version of SplitStream [3] - a structured peer-to-peer protocol for video streaming multicast. In consequence, we use a tree multicast topology with a DHT-based P2P protocol that meets the low delay requirements (although the current design can be easy extended to a mesh structure for increased reliability). We implemented several extensions to SplitStream that allow the QoS-related resource management and shapes the multicast tree topology in order to reduce the load on the service provider. The main idea is that once the uplink available resources are determined, each peer can decide to become an interior node in any of the multicast trees. If no resources are available, the node will be a leaf in any multicast tree it wants to join.

For this, peers will use a separate *node identifier* to advertise their presence in the P2P overlay for each tree. Eventually, when a peer has no available uplink resources it should advertise itself using only a corresponding leaf identifier, indicating that although it is part of the network, it cannot accept other children. The decision on which identifiers are advertised is taken solely by the peer depending on the available resources. Using this P2P mechanism, an RGW can easily discover potential parents for each description because it can filter out neighbors that do not have resources. The root of each tree is the IPTV streaming server, which joins as the first peer in the overlay when the IPTV provider registers to the Nozzilla service. However, unlike SplitStream where the node resources are not quantified, we want to reduce the probability of a peer connects directly to the streaming server. For this we use a random parent selection algorithm with the exclusion of the root, resulting in a uniform load distribution among the peers with resources while excluding the service provider whenever possible.

V. Prototype Performance Evaluation

In this section, we present some experimental results with the goal to assess the feasibility of the P2P protocol using the PeerfactSim.KOM [10] simulator, without going into the evaluation of the video path. The evaluation scenarios considered the limitations of the simulator and consisted in a small-to-medium network with 100 to 10000 RGWs. We used a static model where the peers inter-arrival time was large enough that the system converged into the new state after every join. Because we did not use a dynamic model (currently there is no study examining the behavior of RGWs) our results only illustrate whether the system is functional under limited resource assumptions. The RGWs were equally divided in eight classes: first class has res resources for all priorities,



(a) The routing performance

(b) The number of root children

(c) Peer distribution with tree level

(d) The multicast tree depth

Fig. 4. Evaluation results

second only for HP and MP, third for HP and LP and so on, with the eighth class having zero resources. The average number of resources per RGW per priority is $res/2$. There are four scenarios: $res \in \{1, 3, 5, 7\}$. For each input data set we performed minimum ten simulations and the results obtained show the 99% confidence intervals.

A. *Routing Performance*

One of the objectives of the experiment was to evaluate the performance of joining the multicast tree considering the dynamic nature of the overlay. Because peers send *negative advertisement* whenever local resources are exhausted, the routing distance increases causing peers to be rejected. Figure 4(a) illustrates the average number of hops versus the overlay network size for the four selected resource values. These results indicate that a new node can join a multicast tree in a reasonable number of hops and that negative advertisements decrease the number of hops necessary to join a multicast tree, because they reduce the number of advertised identifiers. The routing performance is improved when increasing the value of res, although it is less significant when $res \geq 3$.

B. *Peer Distribution in the Multicast Tree*

We analyzed the distribution of peers inside the multicast tree to determine whether the P2P overlay plays an important role in distributing the video traffic. The analysis focused on the number of peers using the IPTV server, the multicast tree depth and the distribution of peers versus the tree level.

The number of peers using the IPTV server (i.e. tree root children) is shown in Fig. 4(b). Our target is to minimize this number by avoiding the selection of the root as the first hop whenever possible. When $res = 1$ the percentage of root children is large because a peer has in average 0.5 uplink resources per description. Thus, in average 50% of the RGWs cannot be tree interior nodes, while the remaining 50% will form a highly unstable peer group from which a peer leaves whenever it accepts a new child. This tendency is also observed in Fig. 4(c) illustrating the distribution of the peers with the tree level for an overlay of 10000 peers.

The behavior is improved when the average resources are increased, in part due to the parent selection process. However, the drawback is the higher end-to-end delay of the video traffic, which is proportional with the average peer level. When the average number of resources is higher, most parents are regular peers leading to a greater average tree level and a higher tree depth. This aspect can be observed in Fig. 4(c), where for $res \in \{5, 7\}$ most of peers have a tree level greater than one. Similarly, in Fig. 4(d) the average depth of the multicast tree is proportional with the number of resources (although the difference when $res \in \{3, 5, 7\}$ is smaller than the measurement error).

VI. Conclusions and Future Work

In this paper we presented Nozzilla, a P2P-based IPTV streaming service for an IMS-based NGN. Its goal is to reduce streaming costs for IPTV providers, whenever the transport provider does not have the necessary infrastructure. Nozzilla maintains a P2P overlay between NGN RGWs that is used to create streaming overlays for the IPTV service. In order to validate our proposal, we introduced a QoS-aware P2P prototype based on SplitStream that reduces the selection time of streaming sources and the load on the IPTV provider, albeit with a higher but acceptable end-to-end delay. Our future work will consist in refining the protocol design, by taking into account the impact of the video traffic, and a more accurate model of the NGN user elements. We plan to include support for various types of terminals and mobility between transport providers, and to define the necessary IMS signaling considering its impact.

ACKNOWLEDGMENTS

This article has been partially supported by the Spanish Ministry of Science and Innovation through the CONPARTE project (TEC2007-67966-C03-03/TCM) and by the Madrid Community through the BIOGRIDNET project (S-0505/TIC-0101).

REFERENCES

[1] C. Lee and D. Knight, “Realization of the next-generation network,” *Communications Magazine, IEEE*, vol. 43, no. 10, pp. 34–41, 2005.

[2] 3GPP, “IP Multimedia Subsystem (IMS); Stage 2,” 3rd Generation Partnership Project (3GPP), TS 23.228, Sept. 2008. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/23228.htm

[3] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, “SplitStream: high-bandwidth multicast in cooperative environments,” in *Proceedings of the nineteenth ACM symposium on Operating systems principles*. ACM New York, NY, USA, 2003, pp. 298–313.

[4] K. Sripanidkulchai, A. Ganjam, B. Maggs, and H. Zhang, “The feasibility of supporting large-scale live streaming applications with dynamic application end-points,” *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 107–120, 2004.

[5] N. Magharei and R. Rejaie, “PRIME: Peer-to-Peer Receiver-drIven MEsh-based Streaming,” in *Proceedings of IEEE INFOCOM*, 2007.

[6] D. Kostic, A. Rodriguez, J. Albrecht, and A. Vahdat, “Bullet: High bandwidth data dissemination using an overlay mesh,” *Operating systems review*, vol. 37, no. 5, pp. 282–297, 2003.

[7] J. Zhang, L. Liu, C. Pu, and M. Ammar, “Reliable peer-to-peer end system multicasting through replication,” in *4 thInternational Conference on Peer-to-Peer Computing (P2P 2004), August*, 2004.

[8] F. Wang, Y. Xiong, J. Liu, B. Burnaby, and C. Beijing, “mTreebone: A Hybrid Tree/Mesh Overlay for Application-Layer Live Video Multicast,” in *IEEE ICDCS*, 2007.

[9] N. Magharei, R. Rejaie, and Y. Guo, “Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches,” in *Proceedings of IEEE INFOCOM*, 2007.

[10] A. Kovacevic, S. Kaune, H. Heckel, A. Mink, K. Graffi, O. Heckmann, and R. Steinmetz, “PeerfactSim. KOM-A Simulator for Large-Scale Peer-to-Peer Networks,” *Technische Universitat Darmstadt, Germany, Tech. Rep. Tr-2006-06*, pp. 2006–06, 2006.