

Peer-to-peer Overlay Multicast for Scalable Audiovisual Services over Converging Wired and Wireless Networks

Ahmed Mehaoua¹, Li Fang¹, George Kormentzas², Dominique Seret¹

¹ University Paris Descartes - Faculty of Mathematics and Computer Science
45 rue des Saints Pères 75006 Paris – France - Tel: +33 1 39 25 40 59, Fax: +33 1 39 25

Email: {mea, fang, seret}@math-info.univ-paris5.fr

² National Center for Scientific Research 'Demokritos', Institute of Informatics & Telecommunications,

15310 Aghia Paraskevi Attikis, POB 60228, Athens, Greece

Email: gkorm@iit.demokritos.gr

Abstract. The deployment of scalable audiovisual multicast services over heterogeneous core and access networks is a challenging problem. In this article we propose E-Cast, an efficient, source-specific, scalable, overlay multicast system based on the peer-to-peer system Kademia. This novel overlay multicast system enables IPTV multicast to be deployed as a service level infrastructure to overcome IP network multicast discontinuity in converging wired and wireless network domains. E-cast consists of a set of edge devices called E-Cast Service Node (ESN) and one managing device called E-Cast Service Manager (ESM). ESNs are distributed in the network and provide efficient video distribution services. They not only transport the media-streaming but also can cache the media files according to some given criteria. Performance evaluation using simulation and features comparison with existing proposals are also proposed.

Keywords: Overlay multicast, IPTV streaming, Peer-to-peer communication, Scalability.

1 Introduction

Although the concept of IP multicast was introduced in 1989 and it is now enabled in many routers, many network providers are still not willing to deploy multicast services today. The IP multicast model allows scalable and efficient multi-party communication, particularly for groups of large size. However, deployment of IP multicast in a multi-operator environment requires substantial infrastructure modifications and coordination. The advantage of overlay multicast is that it is able to bypass these issues and network layer complexity. However, the performance of overlay multicast might not be as good as plain IP-based multicast. The deployment of scalable audiovisual multicast services over heterogeneous and converging core and access networks is a challenging problem.

In this article, a novel overlay multicast system, named E-cast, is proposed to enable multicast to be deployed as a service level infrastructure to overcome IP

network multicast shortcoming. E-cast is setting up a virtual topology constructed on top of multiple network domains and involving a set of distributed application-level multicast service nodes called E-Cast Service Nodes. These E-Cast Services Nodes communicate with terminals and with each other using a set of unicast signalling and forwarding mechanisms. To ensure scalable audiovisual service operations, E-cast is based on the peer-to-peer system Kademia, where E-cast Service nodes act as peer proxies that forward and replicate AV data packets on behalf of the senders. The data paths among E-Cast Service Nodes within an E-Cast session form a virtual multicast tree, where each tree branch is a QoS-enabled unicast connection pre-established. The association between a terminal and its delegated E-Cast Service Node for a particular multicast service request is decided by an E-Cast Service Manager and based on pre-defined policies.

The remainder of the paper is as follows: Section 2 provides the related works and features comparison of the E-cast system. Sections 3 and 4 present the overall architecture and related protocols. Section 5 then describes the performance evaluation of the system and results analysis using three simulation scenarios. Finally, future works and conclusion are provided in Section 6 and 7 respectively.

2 Related Works

Many projects have explored implementing multicast at the application layer. Some of them offered some ideas for the design of E-Cast. We will describe them briefly and compare them at last.

Narada[1] is one of those first projects. It constructs multicast trees in a two-step process: firstly, it constructs efficient meshes among participants, and secondly, it constructs spanning trees basing on the mesh. The mesh-based approach supports well multi-source applications. Narada serves specially among end-to-end system and wants to know the global state of a system. In this case, its control is central and not scalable for serving a large number of terminals. It isn't suitable to construct the whole architecture of E-cast, but its mesh-based approach is the base of the construction of E-cast's intra-cluster.

OMNI[3] is a two-tier infrastructure to efficiently implement large-scale media streaming applications on the internet. Different to Narada, OMNI is a proxy-based system. Service providers deploy a set of service nodes (called MSNs) in the network and these MSNs are organised into an overlay and act as proxies to serve a lot of clients. In order to have a good use of bandwidth and distribute a fair charge among MSNs, OMNI gives a dynamic priority to different multicast service nodes based on the size of their service set respectively. Moreover, an approach which iteratively

modifies the overlay tree using localized transformations was proposed to adapt with changing distribution of MSNs, clients and network conditions. ESNs of E-Cast system are similar to MSNs of this infrastructure.

Bayeux[4] and Scribe[5] are two overlay multicast system basing on peer-to-peer architecture protocols called Tapestry[6] and Pasty[7] respectively. Bayeux utilizes a prefix-based routing scheme which it inherits from Tapestry, a wide-area location and routing architecture. On top of Tapestry, Bayeux provides a simple protocol to

organize the multicast receivers into a source-rooted distribution tree. In addition, Bayeux leverages the Tapestry infrastructure to provide simple load-balancing across replicated root nodes and reduced bandwidth consumption by clustering receivers by identifier. Finally, Bayeux provides a variety of protocols to leverage the redundant routing structure of Tapestry. Scribe is another overlay multicast architecture similar to Bayeux. It builds a multicast tree per group on top of a Pastry overlay, and relies on Pastry to optimize the routes from the root to each group member based on some metric, like latency. Compared to Bayeux, the expected amount of group membership information kept by each node in Scribe is smaller because this information is distributed over the nodes. And group join and leave requests are handled locally. Additionally, the multicast tree in Bayeux consists of the routes from the root to each destination, while in Scribe the tree is composed of the routes from each destination to root. Consequently, messages traverse less long links near the root in Scribe. These two architectures show us the benefit of distributed-hash table (DHT) to locate node and values. That spurs us to develop content advertisement protocol and content discovery protocol on kademia, a peer-to-peer information system based on the XOR metric.

The two last above architectures can find neighbour nodes (in logic) and files quickly but they don't take consider of the underlying network structure. Two nodes near in logic ID can be very far each other. As a result, that will cause long latency and degrade the quality of a media streaming service. Recently, some researchers attribute to exploit underlying network topology data to construct efficient overlays. LCC [8] is one of these projects. It consists of two phases: a locating phase and an overlay construction phase. The first one is based on an accurate and scalable global position technique. Using partial knowledge of location-information for participating nodes, the algorithm consists in locating the closest existing set of nodes (cluster) in the overlay for a newcomer. Secondly, the multicast overlay construction phase builds and manages a topology-aware clustered hierarchical overlay so as to optimize the average end-to-end delay. This scheme shares the same idea of E-Cast topology construction protocol among ESNs to avoid initially randomly-connected structures and construct a topology having less latency.

Finally, Table.1 provides a comparison of the main features of the discussed P2P systems and protocols. The criteria are chosen in order to suit live media applications.

Table 1. Features comparison of P2P systems and protocols for the support of live media streaming.

Protocols	Narada	OMNI	Bayeux	Scribe	LCC	Ecast
Underlying network topology-aware	Y	N	N	N	Y	Y
Distributed control (scalable)	N	Y	Partial	Y	Y	Y
Proxy-based	N	Y	Y	Y	Y	Y
Content advertisement protocol	N	N	Y	Y	N	Y
Content discovery protocol	N	N	Y	Y	N	Y

3 E-CAST System Architecture

In this paper we propose E-Cast, an efficient, source-specific, scalable, overlay multicast system. This system consists of a set of devices called E-Cast Service Node (ESN) and one managing device called E-Cast Service Manager (ESM). ESNs are distributed in the network and provide efficient data distribution services. They not only transport the media-streaming but also can cache the media files according to some given criteria. The ESM coexists with service provider and manages the service of those ESNs. Four actors are connected by ESNs or ESM in E-Cast: Content Consumer (CC), Content Provider (CP), Service Provider (SP) and Network Provider (NP). A CC subscribes with SP to get media-streaming transported or cached by ESNs. A CP publishes its hot and new media information at SP and distributes the information to ESNs, and sends the media packets to a large number of CCs through ESNs. An SP is in charge of the management of the services delivered from CP to CC through an ESM. An SP establishes one or several agreements previously with many CPs. Moreover, a CC can pass the information of a media file which it wants by SP to execute a searching among ESNs to find it. NPs provide physical support to CCs, CPs and SP, for example, the placement of ESNs. The E-Cast overall architecture is depicted in Figure 1.

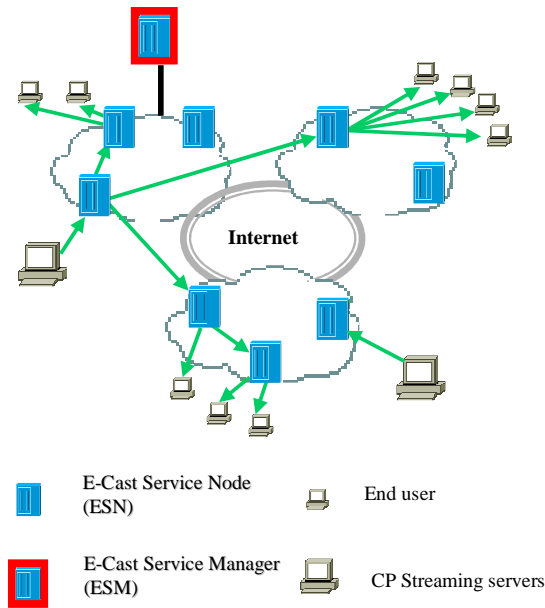


Fig. 1 : E-Cast system architecture

4 E-CAST OVERLAY PROTOCOLS

Four protocols cooperate in E-Cast system to realise a resilient, efficient, source-specific and scalable media overlay multicast. *E-Cast topology construction protocol* is in charge of helping a new ESN to find a proper ESN cluster to bootstrap E-Cast service and clustering the ESNs which are close to each other on considering underlying network condition. *E-Cast content advertisement protocol* is carried on by the content providers. They publish the information not only on the service provider but also on ESNs. However, the information on ESNs is different from that on the list of service provider. The core of this information is a hash value of the real information, and this value serves for a rapid, effective searching of content based on the kademlia [9] protocol. This searching procedure is carried on *E-Cast content discovery protocol*. This protocol takes effects when one content consumer joins E-Cast system. The CC can choose a media file information from its subscribed SP or find a media file information distributed on ESNs by the searching procedure. This search can even be applied on the E-Cast deployment of another SP, which has already established the cooperation relationship previously with the subscribed SP. On the basis of the information published by content advertisement, content discovery protocol locates the media file either on the source (CP streaming server) or on one of the ESN which has cached it to provide a high quality and rapid transporting. Finally, *E-Cast overlay multicast routing protocol* is in charge of constructing and maintaining a reliable, fault-tolerant, efficient multicast tree for a content session facing a large number of dynamic content consumers. Moreover, it is responsible to construct a tunnel between the content consumer and the ESN of another E-Cast system.

4.1 E-Cast topology construction protocol

E-Cast topology construction protocol takes the underlying network condition in considering, clusters the ESNs which are close to each other and classifies them into two levels. This way can reduce the latency from the source to ESNs and balancing the consumption of network bandwidth.

In E-Cast system, we adopt a similar approach of LCC protocol [8] to cluster ESNs so as to provide a resilient and low-latency topology to bootstrap the E-Cast system. There is a previously given active node called Rendezvous Point (RP) for a new ESN, say node A . RP can be the ESM or the ESN cluster leader recently or often connected by the new ESN. In general, the information of the two nodes is both known for A , A contacts firstly with the known ESN cluster leader. If the ESN cluster leader is still alive and can still accept new ESNs, A joins to its cluster. If the ESN cluster leader has already left or is already saturated, A contacts to the ESM to obtain the identity of a randomly selected boot cluster leader, B . A measures the distance (delay) from itself to B , $d(A,B)$, and then compares it with the given limited delay, R_{max} . If $d(A,B) < R_{max}$, A joins B 's cluster. Otherwise, B queries its all neighbour cluster representative nodes to locate closest clusters for A and then it sends a candidate list to A . A chooses the closest to measure the delay and compares, all like to B . This procedure is repeated until A finds a proper cluster to join or a give searching time T is reached, A creates its own cluster. Then A will get a cluster ID (CID) which is unique in one E-

Cast and a local ID (EID) which is only unique in its cluster. If *A* has joined to a cluster, the CID is given by the cluster leader. If *A* creates its cluster, the CID will be generated randomly. The combination of CID and EID of an ESN is unique in an E-Cast system and will be used until it leaves the system. To be resilient, *A* will conserve also the information of the second closest ESN cluster leaders and the edge node which is in several clusters' scope. The multicast tree construction on the clusters' leaders and interior of one cluster will be organised and managed by *overlay multicast routing protocol*. The figure 2 depicts the cluster structure after running E-cast topology construction protocol.

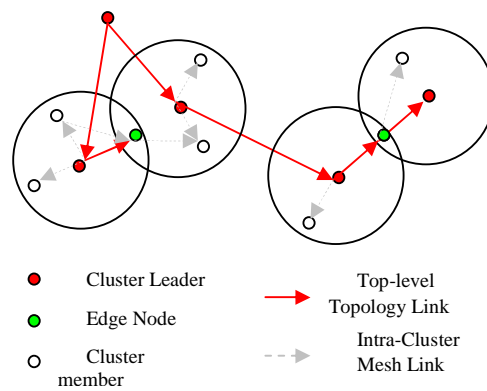


Fig. 2 : E-Cast topology construction involving ESNs

4.2 E-Cast content advertisement protocol

The procedure of content advertisement consists of two parts. Shortly, Content provider publishes its media information to service provider and distributes the information of contents to ESNs to support the searching procedure of the *E-Cast content discovery protocol*.

The first part is between CPs and SP. CP updates regularly the information of its media contents database and sends it to the SP so that content consumers can look for and then choose what they want in the available contents at a moment. The depiction of each media file is complete: name, quality of media, published date, etc. At the bootstraps step, CP sends a complete description of the whole media content database. After that, CP sends regularly just the modification of its database: new media files, out of date files and etc. It is to avoid bandwidth waste.

Moreover, CP distributes the information of contents to ESNs. That's the second part of advertisement protocol. The principle of this part is similar to the STORE procedure of protocol Kademia [9]. This step guarantees that SP can find a media file efficiently even if this file isn't in SP's local list. In this step, the depiction of a media file is a set of $\langle sID, sValue, SP@, CP@, QoC, n \rangle$, called *key* of a media content.

“sValue” can be the name or a short description of a media file. “sID” is a 160bits hash value of sValue, uniquely identifying a media file in a E-Cast system. SP@ is the address of SP (or the address of ESM coexisting with SP) which has the right to manager this file. Since one CP can probably have contact with more than one SP. This value is used to distinguish different SPs and will be useful for the cooperation between SPs. The “CP@” is the address of a content provider server. “QoC” is the quality of content, a description of quality of this media file in terms of resolution. This description can be just a degree number, like (1 to 10), the higher resolution a media file has, the higher degree it should be set. The norm which evaluates a content quality should be unified for all the content providers previously. A media file quality is evaluated automatically on basis of the norm and then a QoC degree corresponding is set in the *key*. “n” is a caching start switch counter for registering the amount of downloading of this file. This value is very important for start ESN caching function. It is initialized to 0 and augmented by 1 if and only if when this file is downloaded and also the ESN is in the multicast tree of this file downloading. When “n” reaches to a given value, which means that this file is enough “hot” for this ESN, this ESN begins to cache this media. The caching start value can be set manually. For the ESNs which have great storage capacity, the caching start value can be set very small. “n” starts to decrease if the content doesn’t pass any more for a given time. When a maximum limited time reached, the content will be removed from the ESN for save the space for other new hot media files. One *key* is stored on ESNs with EIDs “close” to the sID in their exclusive or value.

4.3 E-Cast content discovery protocol

A CC can acquire the media information by two ways. The simpler case is to choose the content which has already had the information in a list of the SP. In this case, a client will be redirected to a proper ESN. In another case, CC can search the content for the reason either the current version of this media is not satisfied or the content it needs is not yet known for the SP. In this case, a searching procedure will be carried on firstly among ESNs which conserve the information of contents in a distributed way. And then, if the searching result is positive, a list of information of contents will be returned to client to choose. CC can choose a proper version based on its hardware ability. For example, it can choose a high resolution video if it has enough bandwidth resource.

E-Cast content discovery protocol takes effects on both of the two cases. It works similarly to the FIND-VALUE procedure of protocol Kademia [9]. We present the second case the first. In the second case, the content information entered by CC is hashed to a 160bits value, called SearchValue (SV). A searching procedure starts at the same time in all of the clusters to look for an ESN which has EID closest to SV. And then, each cluster returns the information of its ESN which has the closest file information and the related file information registered on this ESN to SP.

4.4 E-Cast overlay multicast routing protocol

An ESM coexists with one service provider. It is responsible for computing the multicast tree for one content session whose source is on a content provider or on an ESN which has cached the content. Specially, it contacts regularly with ESN cluster leaders to get the fresh information of ESN which has cached “hot” media to updates its local information. In the part of an ESN, it knows its neighbour ESNs and all its local content consumers. Moreover, the ESN reports regularly its information about “hot” media to its cluster leader.

5 Performance Evaluation

We simulated the E-Cast topology construction protocol on the PeerSim simulator. Since the importance of this protocol is its two-level cluster structure which takes consider of the underlying network condition. We especially evaluated its efficacy in reducing latency and in balancing link charge. The underlying network topologies were given randomly to insure the reliability of evaluation.

5.1 PeerSim simulator

There are many simulators for evaluating a new protocol, like Narses, 3LS, NS2, PeerSim, etc. Each of them has its advantages and disadvantages. PeerSim simulator is simple to simulate a scalable network topology, which can reach to 10^6 nodes, and has a graph factory to produce varied network topologies. Those characters are suitable to evaluate the efficacy of E-Cast topology construction protocol. PeerSim has been developed with extreme scalability and support for dynamicity in mind. Our simulation is to evaluate the efficacy of the two-level cluster topology construction on the scalable network, and besides, we need to compare the longest propagation time from the source to terminals for unicast and two overlay multicast architectures (E-Cast and Narada), so the underlying network condition should be kept the same. For all these reasons, the cycle-based engine was chosen. Benefiting on the graph generator of PeerSim, after some modifications, one underlying network topology can be produced randomly each time and our experiments were carried on these random topologies to get a result more accurate. The figure 3 is a network topology example with 100 terminals and 5 ESNs.

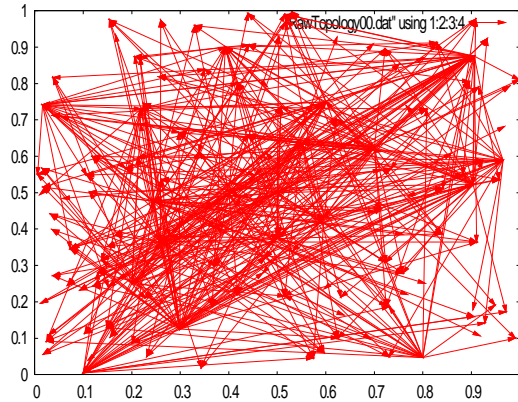


Fig.3. An example of E-cast network topology

In this graph, the point aimed by one or several arrows represents a terminal (content consumer). The point reached by arrows and also from which starts one or several arrows represents a proxy node (a router for the unicast or an ESN for E-Cast and Narada). The points has just arrows setting out from it represents an original media source.

5.2 Simulation model

The experiments were carried on the random network topologies which has one original media source. For each topology, we simulated three transport architectures: (1) unicast, (2) E-Cast and (3) Narada. To insure the reliability of evaluation, we tested 10 times for each architecture and the average of results of 10 times was taken as the final result.

Model assumptions are the followings:

- 1) The Euclidean distance between two directly connected nodes was considered as the IP packet transfer delay on this link. For example, if terminal node 25 connects directly with proxy node 3, and the Euclidean distance between proxy 3 and terminal 25 is 12 units of distance, the delay from proxy 3 to terminal 25 is 12 ms. And if node 25 connects with proxy 5 passing by proxy 3, and the distance between proxy 5 and proxy 3 is 30 units of distance, the delay between proxy 5 to terminal 25 is 42 ms.
- 2) Since packet loss is not evaluated in the E-Cast topology construction protocol, it was assumed that all the terminals can always receive one duplicate of packet by one or a series of proxies even if there is a limit serving amount for one proxy. And there is no congestion in the system.
- 3) The amount of terminals served by a proxy is limited to avoid all or most of terminals concentrated to one or several proxies. But as mentioned in 2), it doesn't cause an isolated terminal.

Another condition to guarantee that we can compare the results of three architectures is that the algorithm to find the shortest path is always *greedy* algorithm. In unicast, we used it to find the shortest path from the source to each node. In Narada, we used it to find the nearest proxy for terminals and nearest proxy neighbour for proxies. In E-Cast, it was used to find the nearest ESN for terminals, the nearest cluster leader for ESN and the shortest path to the source for a cluster leader.

Two parameters were tested:

- 1) **MAX_DELAY**: The longest delay from the source to the terminal in the shortest path so that all the terminals receive a duplicate of a packet. This parameter is used to evaluate the effect on reducing the latency of E-Cast.
- 2) **MAX_NUMBER_DUPLICATE**: The maximum number of duplicate of one packet passed on one link in order that all the terminals get one. This parameter is to evaluate the capability in balancing network charge of E-Cast.

We evaluated two scenarios:

- 1) **Scenario A**: the number of terminals was set on 100, 200, 500, 1000,. The number of proxy was fixed on 50. And for E-Cast, the number of cluster was fixed on 5;
- 2) **Scenario B**: the number of proxy began from 5, augmented by 5 for each experiment, until 50. The amount of terminals was fixed on 1000. For E-Cast, the cluster number is fixed to 5;

5.3 Results analysis for scenario A

In this scenario, the number of proxies was fixed to 50, which means there are 50 routers in unicast, 50 ESN in E-Cast and 50 ESN in Narada. The 50 ESNs of E-Cast were grouped into 5 clusters.

The figure 4(a) depicts the **MAX_DELAY** change with the augment of number of terminals. Since the amount of terminals served by a proxy is limited, the shortest path for the farthest terminal is logically extended with the augment of number of terminals. That's the reason for which the **MAX_DELAY** in tree architectures are increased. As we know, in unicast, there is always only one source to get media packet: the original source. As a result, one duplicate packet begins always from original source and arrives to one terminal finally. However, in E-Cast and Narada, the ESNs serve also as a media source, the packet can be acquired directly. Consequently, the **MAX_DELAY** in unicast is much longer than that of E-Cast and Narada. Between E-Cast and Narada, E-Cast can be still better because the ESNs have been clustered according to their locations in underlying network.

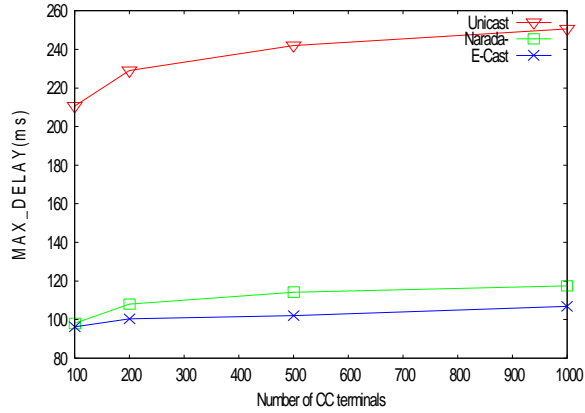


Fig.4(a) Maximum transmission delays VS numbers of end-terminals

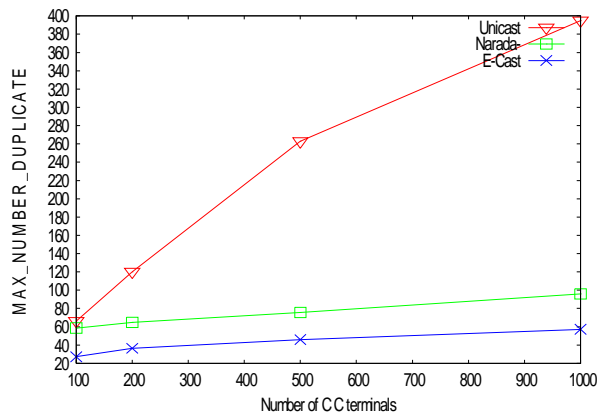


Fig.4(b) Media Packets duplication VS numbers of end-terminals

The same character is proved on the effect of MAX_NUMBER_DUPLICAT in the figure 4(b). With the augment of terminals, more and more duplicates of packet were transferred on the shortest path of each terminal. In unicast, the duplicates starts always from the original source, the routers near to the source is heavily charged. They have to transfer the packets for their own serving terminals, and at the same time have to transfer those just passing by them. In E-Cast and Narada, the links become also busier but much more softly since ESNs can produce the duplicates. In fact, their augmentation of MAX_NUMBER_DUPLICATE is most caused by the augmentation of local terminals. Between E-Cast and Narada, the same reason in 4(a), since the shortest path is improved by cluster the nearby ESNs, the repeating link is consequently reduced.

5.4 Results analysis for scenario B

In this scenario, the terminal amount is fixed to 1000, and the cluster amount for E-Cast is fixed to 5, we change the number of proxies to evaluate.

The figure 5(a) depicts the different influence on the MAX_DELAY. The augmentation of proxies has great significance for unicast to reduce MAX_DELAY. More routers exist in the system, one terminal can find a closest router to attach with more possibilities. As a result, the curve for unicast decline greatly with the augment of amount of routers. However, since the packet starts always from the original source, the MAX_DELAY is always much greater than those of E-Cast and Narada. Like in the Scenario A, the curve of MAX_DELAY of E-Cast and Narada change very softly, in this scenario, they are still not sensible to the change of amount of ESNs. It is because each ESN can be considered as a source, and terminals always look for the nearest source, E-Cast and Narada are very adaptable to the amount of terminal change.

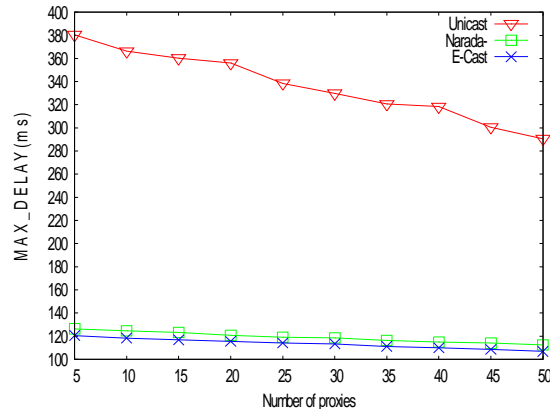


Fig.5(a) Maximum transmission delays VS numbers of ESN proxy nodes

The figure 5(b) depicts the different effects on MAX_NUMBER_DUPLICAT. The tree architectures are all sensitive to the augment of proxy because the more often a link is chosen to be the shortest path or part of the shortest path, it is more charged. As a result, when the number of proxy is augmented, the charge of the busiest is lightened and shared to the others. In E-Cast and Narada, ESNs share the charge with the original source, so the MAX_NUMBER_DUPLICATE is already much smaller than that of unicast. Between E-Cast and Narada, the cluster leaders of E-Cast can redistribute the charge even better, therefore, E-Cast has the best result. Nevertheless, since terminals connect always with the closest proxy, the rate of improvement is decreased and the advantage of augmenting proxies will be lost in the end.

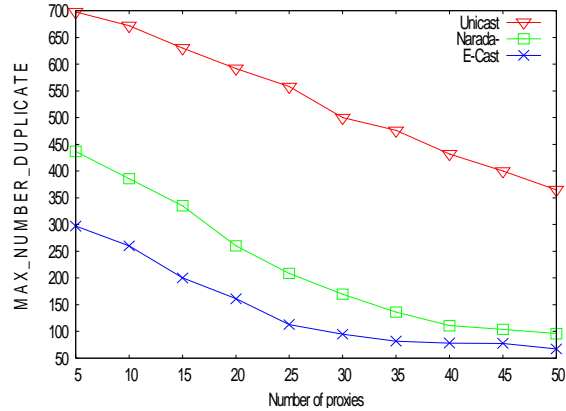


Fig.5(b) Media Packets duplication VS numbers of ESN proxy nodes

6 Future Work

The E-Cast system is currently adapted for Video-on-Demand (VoD) service. The multicast transport takes its advantage when ESNs starts to cache media packets. However, in fact, it is simple to be adjusted for live streaming, like IPTV, IP radio etc. The only modification is some of parameters of the *key* in E-Cast content advertisement protocol. We can replace the information of the content by that of a session in the “sValue”. This value can be the combination of a short description of a media file and its showing time or schedule. In fact, the disadvantage of some software of IPTV, like PPStream [11], TVCool [12] has shown such a function is very necessary to provide a flexible and satisfactory service. On the two systems, the content is repeated for some days and then is renewed. But a content consumer is always passive to get their media streaming and even cannot know the rate of advance. It merely doesn’t matter for the people who pass all their time watching it. The adaptive E-Cast will be able to let a content consumer to find out all of relative sessions thank to the E-Cast content discovery protocol (called session discovery protocol will more accurate in this case) and choose the most suitable one according to his spare time.

In addition, since ESNs are able to cache the media. This character provides us a possibility to break the single source limit and develop a multi-source model to improve video transport quality and fault tolerant ability on basis of some pre-research works, like [13].

7 Conclusion

A novel overlay multicast service management system called E-Cast is designed and evaluated in order to efficiently distribute video streams over multiple and converging wired and wireless network domains. The proposed system automatically locates and clusters newly deployed overlay service proxy nodes named ESNs. The associated overlay topology construction protocol is presented and evaluated. Compared to the well known NARADA system, E-cast is able to considering the underlying network topology to get low latency and balancing the packets transmission charge for the support of scalable and efficient audiovisual services (IPTV, VoD). With VoD services, ESNs are able to cache “hot” media files and share the downloading burden of the original source. The E-Cast content advertisement protocol and discovery protocol based on Kademia protocol ensures that available media files can be quickly shared and located by end-users. Performance evaluations are carried out using simulation (PeerSim) including comparison with unicast transmission and the P2P Narada system. The evaluation results prove that E-Cast has advantage in reducing the latency and balancing the network traffic load.

REFERENCES

1. Chu, Sanjay Y., Rao G., Seshan S. and Zhang H.: A case for end system multicast. In ACM SIGMETRICS (2000)
2. Shi S.: Design of Overlay Networks for Internet Multicast. Ph.D Dissertation, Washington University in St. Louis, August (2002)
3. Banerjee S., Kommareddy C., et al.: Construction of an Efficient Overlay Multicast Infrastructure for Real-time Applications. In IEEE INFOCOM'03 (2003)
4. Shelley Q., Zhuang, et al.: Bayeux: An Architecture for Scalable and Fault tolerant Wide area Data Dissemination. in 11th ACM international workshop on Network and operating systems support for digital audio and video NOSSDAV, New York, (2001).
5. Castro M., Druschel P., et al. : SCRIBE: A large-scale and decentralized application-level multicast infrastructure. in IEEE Journal on Selected Areas in Communications (JSAC) (Special issue on Network Support for Multicast Communications). (2002)
6. Zhao, B. Y., Kubiawicz, et al.: Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Tech. Rep., Univ. of California at Berkeley, (2001).
7. Rowstron A. and Druschel P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. in Proc. of IFIP/ACM Middleware 2001, Heidelberg, Germany (2001).
8. Kaafar M.A., Turetti T. and Dabbous W.: A Locating-First Approach for Scalable Overlay Multicast. In IEEE IWQoS 2006, New Haven, CT, USA (2006)
9. Maymounkov P. and Mazières D.: Kademia.: A peer-to-peer information system based on the XOR metric. In First International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA (2002)
10. PeerSim P2P simulator: <http://peersim.sourceforge.net/>
11. PPStream: <http://www.ppstream.com/>
12. TVCool: <http://www.coolstreaming.us.tv/>
13. Mushtaq A. and Ahmed T.: Adaptive Packet Video Streaming Over P2P Networks Using Active Measurements. In Proceedings of the 11th IEEE Symposium on Computers and Communications, Cagliari, Italy (2006)