

Multicast Cloud with Integrated Multicast and Unicast Content Distribution Routing¹

Dan Li, Arun Desai, Zheng Yang, Kenneth Mueller,
Stephen Morris, Dmitry Stavisky
Cisco Systems, Inc.

Abstract

In this paper, we describe the concept and design of "application-layer multicast cloud", the first overlay network design that provides integrated content distribution routing between IP multicast and unicast via a user-configured group of multicast senders and receivers for content distribution across different layer-3 IP multicast groups, content distribution channels, and unicast distribution relay trees, that for the first time allows for application-layer (as opposed to layer-3 or layer-4) control of multicast security, failover, QoS, and bandwidth utilization. None of these were possible before and yet the need for such capabilities is great in Content Distribution Network (CDN) deployments.

Note that here "multicast cloud" refers to an "application-layer" user configured entity with (1) a number of properties that govern the IP multicast flow and the interaction between multicast and unicast content distribution, and (2) a set of hosts that can participate in multicast content replication that happens on multiple layer-3 IP multicast groups and across multiple layer-3 IP multicast islands.

1 Introduction

The foundation of scalable content distribution is efficient one-to-many data transport, i.e., multicast². Many content distribution network designs have centered around "application-layer multicast" built on top of IP unicast, for its effective transport, incremental deployment, asynchronous delivery, application-aware routing, and versatility [1].

While some have embraced application-layer multicast and denounced the usefulness of IP Multicast, we think both have strengths and weaknesses. An optimal "multicast" system would take advantage of both of them. In particular, IP multicast helps application-layer multicast utilize

physical broadcast media, build efficient distribution, and improve scalability [2].

In this paper, we present the architecture and design of a content distribution network (CDN) that effectively integrates IP unicast and IP multicast into a single application-multicast architecture, and offers the CDN operator superior application-layer control of the distribution traffic.

Here are some terminologies we use throughout this paper:

- The CDN consists of many devices deployed throughout the network, that we call the "Content Engines" or "CEs". Collectively, they perform the function of pre-positioning content from "Origin Servers", often located at the corporate data center or scattered on the Internet, to the edge of the networks where the end-users reside, e.g., the retail stores or field offices.
- To manage the CDN devices and activities, the CDN administrator uses a central management station we call the "Content Distribution Manager" or "CDM". CDM communicates with every CE in the CDN, sending them configuration updates and collecting status information.
- To specify what content goes to which CEs, CDN administrator creates a "Channel" on the CDM. The channel defines a set of subscriber CEs as well as a head among them that we call the "Root CE" for the channel. Content enters into the CDN via the Root CE, which contacts the origin server to download the content.
- In channel routing, a CE that stores and forwards content for other CEs is called a "Forwarder CE". A CE that receives content from a forwarder CE is called a "Receiver CE". Note that a forwarder CE is often a receiver CE itself as well because the forwarder CE in turn needs to receive the content from its own forwarder to begin with.

This paper, then, is mostly concerned about how the content flows from the Root CE to all the other subscriber CEs in the most efficient manner, given the availability of unicast and multicast network connectivity among the subscriber CEs. Deciding the distribution flow for each channel is a process we call "Channel Routing".

In the rest of the paper, Section 2 describes a multi-tier overlay topology as a means for the CDN operator to specify the network topology in terms of unicast connectivity and adjacency. Section 3 presents the concept of "multicast cloud" as a means to specify the IP multicast topology. Section 4 details the channel routing process that synergistically combines IP unicast and IP multicast into coherent application-layer multicast. Section 5 describes the added benefit of our design in terms of superior application-layer

¹ Please see the online version at <http://www-cs.stanford.edu/~dli/mcast-cloud-paper.pdf> and contact <lidan@cisco.com> for matters related to this paper.

² "multicast" here refers to the 1-to-many service model, not necessarily the specific instance of IP multicast.

traffic control. Section 6 is the related work and Section 7 concludes the paper.

2 Overlay Topology

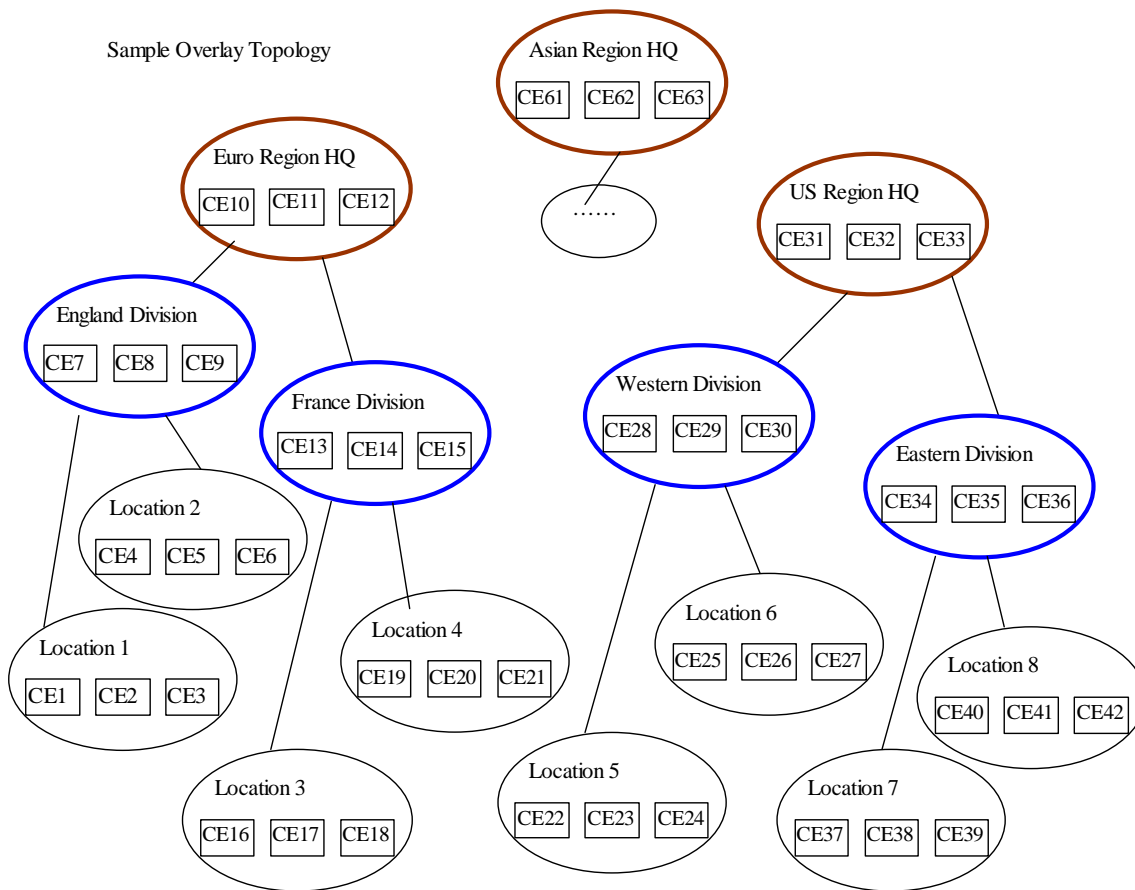
The purpose of an “overlay topology” is to guide the construction of the application-layer multicast tree for content replication. The CDN administrator configures it on the CDM for the entire CDN, based on CEs’ physical locations in the IP network. Then, for any particular channel, the CDN automatically forms a “channel distribution tree” among all CEs assigned to the channel, following the overlay topology as much as possible, --- a process called “channel routing”.

In this architecture, the overlay topology has multiple tiers, with “tier-1” being the top tier (and normally closest to the IP backbone), and then “tier-2”, “tier-3” and so on, toward the edge IP networks. On each tier, there are multiple locations, corresponding to data centers or POPs or topological vicinities in the IP network. Each location may have multiple CEs in it. Every location has a “parent” location on the higher tier, except tier-1 locations.

The CDN administrator configures the overlay topology at

the CDM by first grouping CEs into locations based on their geographic adjacency and then specifying child-parent relationships between locations. A location can have a single parent. Locations with no parents are placed at tier-1. Locations whose parents are at tier 1 are placed at tier 2, and so on. Hence, this topology can be thought of as a forest of trees rooted at tier 1. Note that such an overlay topology will never have loops. See the sample graph below on a 3-tier overlay topology.

We chose to model the overlay network as a forest instead of a mesh for several reasons. First, a tiered overlay design matches well with the reality because it is our experience that most corporations and ISPs have a multi-tier IP network instead of a full mesh network. Second, the tiered design reduces the overall system complexity in terms of configuration and channel routing because the topology does not have loops. Third, this design is highly flexible in that it can degenerate into either extremes of the spectrum: either into a full mesh if the CDN administrator pools all CEs into a single location or into a strict tree if the CDN administrator makes each CE a separate location of its own and string them together via parent-child relationships. Where in the spectrum the CDN is will be entirely up to the CDN user to configure. Hence this design is highly customizable, enabling us to build one architecture for a



wide range of deployment scenarios.

It's important to note that the parent-child relationships between locations guide but does **not** dictate whether content replication traffic physically flows from the parent location to child location. For instance, if there is no CE in the parent location assigned to the channel or if the Root CE is in a child location to begin with, the content won't flow from the parent location to the child location. In short, the overlay topology provides only the "map" or "search path" for finding channel forwarders, while the CDN provides the rest of the intelligence. If there's no eligible forwarder in a parent or grandparent location, that location is completely bypassed in the application-layer multicast tree constructed for the channel. See also the "integrated channel routing" section.

3 Multicast Cloud

Separate from the overlay topology configuration, which governs how unicast content distribution flows, the CDN administrator also creates a "multicast cloud" on the CDM to govern how the multicast distribution flows and interacts with unicast distribution.

The multicast cloud specifies which CEs will be the multicast senders, which will be multicast receivers within the cloud, as well as properties of the multicast such as the multicast IP address for session advertisements, the range of multicast IP addresses for data transmissions, the IP TTL, whether the multicast medium is satellite or terrestrial, whether and how much FEC (forward error correction) to use, bandwidth and QoS settings, etc. We impose that a CE cannot be a sender in multiple clouds or a receiver in multiple clouds.

Then, the CDN administrator can assign multicast clouds to channels. One channel can have multiple clouds. Likewise, one cloud can be used in multiple channels. A channel can have both multicast receivers and unicast receivers. A receiver CE considers the channel a "multicast channel" if and only if the CE belongs to a multicast cloud as a receiver and the cloud is assigned to the channel. It's possible for such a channel to have some receiver CEs not belonging to any multicast cloud and receive the content via unicast.

On the sending side, a CE sends out content of a channel via multicast if the CE belongs to a multicast cloud as a sender and the multicast cloud is also assigned to the channel. On the receiving side, a CE tunes in to the multicast session advertisement address if the CE belongs to a multicast cloud. If the CE hears any session advertisement for content of a multicast channel it is subscribed to, the CE will tune into the channel multicast IP address to receive the content.

While the concept of multicast cloud and multicast channel is straightforward, the key challenge is in guaranteeing the scalability, reliability and eventual delivery of content. For example, if the receiver is down at the time of the multicast session, the multicast connectivity is broken, or the multicast sender crashed, the receiver CEs still must replicate the content timely. Furthermore, when a multicast cloud consists of thousands of CEs, the multicast loss repair becomes a scalability bottleneck, which often results in poor reliability and calls for a parallel unicast hierarchy for more efficient NAK aggregation and loss repair [3]. Addressing these challenges leads us to integrated channel routing for solutions.

4 Integrated Channel Routing

Next, we detail the flow of content distribution and the construction of distribution trees.

□ Store and Forward via a combination of unicast and multicast⁴

For CDN scalability and reliability, we use "store and forward" to distribute content from the Root CE to all the receiver CEs in the channel, meaning a receiver CE does not just go directly to the Root CE for content. Rather, it finds out who is its "forwarder CE" and goes to the forwarder either for unicast content replication in the case of unicast channels, or for multicast failover and repair service in the case of multicast channels. In turn, the forwarder CE downloads content from its own forwarder (which may ultimately be the Root CE), store the content on disk, and forward the content on to edge CEs that request content from it. If the forwarder CE is a multicast sender per the CDN configuration, it also proactively pushes out the content via multicast. In this case, any receiver in the multicast cloud will get the multicast content, regardless the sender is the receiver's direct forwarder or not.

With this design, content flows through a channel-specific distribution tree via a combination of unicast and multicast, enabling the CDN to reach across unicast IP networks to bridge isolated multicast IP islands. For example, in the sample graph in Section 2, all the country divisions may belong to a global satellite multicast network and receive content via multicast. Then the division receiver CEs turn around and act as unicast forwarders for the children locations within each country, or as multicast forwarders / senders if terrestrial multicast is available in that country. Similarly, a location may receive content via unicast from

⁴ Our system does carry live video traffic over the same overlay network for delivery to the end-users, which does not require store-and-forward. Such live traffic is channel-routed in a similar fashion as described in this paper, but with additional care for live stream performance and reliability, that we won't detail in this paper.

its forwarder and then internally replicate via multicast wherever IP multicast routing is turned on.

□ **A Distributed and Dynamic Channel Routing Process**

While the channel distribution tree is global, the channel routing process is actually distributed, as a *per-CE* function that answers the local question “who is my forwarder for this particular channel”. On each receiver CE, the channel routing algorithm is run periodically to dynamically pick a forwarder for each channel.

The channel routing process considers both the overlay topology configuration and the dynamic availability of eligible forwarders. However, it incurs no “routing probes”, unlike SODA (self-organizing distribution architecture) [4], which is a major improvement (see also the “related work” section). Instead, it uses the byproduct from the content distribution process as the feedback to the channel forwarder selection. Every time a forwarder is chosen, the CE contacts the forwarder for content downloads. If the CE cannot reach the forwarder or experience poor service (e.g., frequent disconnections, indicative of a busy forwarder), the channel routing module remembers this fact about that particular forwarder and tries to pick another (better) forwarder next time around.

□ **Location Leader**

For any location where more than one CE have been subscribed to the channel, one and only one CE in the location will act as the “location leader” for the channel, i.e., to replicate content from outside of the location (via either unicast or multicast), while other CEs in the location will replicate content only from the location leader in the case of unicast. This ensures that only one copy of the content will cross the link connecting any two locations. In the case of multicast, i.e., some CEs in the location are part of a multicast cloud, these CEs will still receive multicast transfers even if the multicast sender is outside of the location. For them, such multicast replication will preempt the need of unicast replication from the location leader. However, if the multicast fails or has losses, the CEs recover using unicast within the location so there is never any extraneous traffic on the inter-location links.

For failover purposes, the other CEs in the location act as “backup location leader” in case the location leader is down. The channel routing algorithm deterministically (through consistent hashing), though distributedly, picks who is the location leader and generates an order list of the other CEs that are 1st backup, 2nd backup, and so on. In the Root location (i.e., the location where the Root CE belongs), the location leader is always the Root CE that the CDN administrator designated, while the “backup location leaders” are actually “backup Root CEs”, which are al-

lowed to go directly to the origin server to download content if the Root CE is down.

Note that a “location leader” is always a per-channel and per-location concept, a “forwarder” is always a per-channel and per-CE concept.

□ **Forwarder Selection**

A receiver CE finds its forwarder by examining the series of locations on the overlay topology “toward” the Root location, following the parent-child relationship.

- First, find a forwarder within the CE’s own location. The location leader should be the forward. If the location leader is down or too busy, use the backup location leader as the forwarder. If found, exit the algorithm.
- If none found or the CE thinks it is the location leader itself, look for a forwarder in the next location “toward” the Root Location⁵. If still none found (e.g., because the CDN administrator assigned no CE of that location to the channel or because all the potential ones are unreachable or too busy), then look further in the yet next location “toward” the Root location, and so on. The recursion ends if a suitable forwarder is found or the algorithm reaches the Root CE’s location.
- *Multicast Forwarder*: for a multicast channel, the channel routing module will run the above search algorithm first trying to find a “multicast forwarder”. Only when it failed to find any suitable multicast forwarder, will it run the algorithm again, this time, looking for “unicast forwarders”. Note that any multicast forwarder is capable of unicast as well.
 - A multicast forwarder for a CE must satisfy all of the following rules, while a unicast forwarder satisfies only the first two rules.
 - #1 The forwarder is on the topological location path toward the Root CE.
 - #2 The forwarder belongs to the same channel.
 - #3 The forwarder belongs to the same multicast cloud as a multicast receiver or sender.
 - #4 The multicast cloud they belong to is assigned to the channel.
- If the search reached the Root location still without success, as a last resort, the receiver CE may decide to contact the origin server directly for content, after proper retries and timeouts during the search.

⁵ The next location “toward” the Root Location may be below, instead of above, this location if the Root Location is below this location on the overlay topology graph.

The multicast forwarders form a parallel CE hierarchy next to the multicast cloud, providing scalability, reliability, and failover to the multicast data transfers. E.g., in the case of multicast loss, instead of all going to the multicast sender for repairs, the receiver CEs go to their respective multicast forwarder for any multicast loss repairs. In case the multicast sender is down or multicast connectivity is broken, the receiver CEs failover to unicast replication from the forwarder instead of from the multicast sender or Root CE.

5 Application-layer Traffic Control

Along with the overlay topology, multicast cloud, and channel configuration, the CDN administrator can set parameters that control many aspects of the distribution traffic, beyond the topologic directions of the traffic flow (which we have detailed in Sections 2 through 4). The many aspects of traffic control include distribution priority, bandwidth control, quality of service, and security.

Much of such traffic control is typically enforced on layer 3 and layer 4, and traditionally configured on layer 3 and 4 as well. That has been problematic in CDN deployment because it is too complex for the (layer 7) CDN administrator to understand and operate. Conversely, in our system, the configuration entities in the CDM become a natural vehicle for the CDN administrator to specify parameters for the overall CDN traffic. These parameters have direct semantics associated with the CDN applications and are configured along with the CDN applications, hence are easy to understand and use for the CDN administrators.

For example, the CDN administrator can configure four bandwidth limits on each CE to control the content distribution traffic.⁶

- Incoming content acquisition traffic from origin servers.
- Incoming unicast content distribution traffic from forwarder CEs
- Outgoing unicast content distribution traffic to receiver CEs.
- Multicast content distribution traffic within a multicast cloud. Due to the nature of multicast, there is a common value for all CEs within the cloud. It's both the outgoing rate from the multicast sender and the incoming rate to the multicast receivers.

A CE may have all these limits defined because all four kinds of traffic can flow through the CE simultaneously. For each bandwidth limit, we also support “time of day”, where the CDN administrator can configure different limits for different time segments that form a weeklong cycle. For example, the admin can say “incoming unicast traffic should not exceed 100kbps from 8am to 8pm Monday

⁶ Other bandwidth limits, e.g., for media streaming and HTTP/FTP browsing, also exist in the system.

through Friday, but it can run as high as 10Mbps from 8pm throughout the night to 8am, Monday through Friday plus whole days Saturday and Sunday.”

Other parameters such as multicast security and key management are also controllable from the CDN application layer. We will detail their designs in separate publications.

6 Related Work

□ SODA

Compared to our last-generation architecture: SODA (self organizing distribution architecture) [4], this design is a major improvement in that (1) the SODA routing probes generate extra traffic overload (sometimes quite heavy) on the network, (2) the routing results are often inaccurate or undesirable because the SODA heuristics may or may not reflect the true needs of every specific CDN deployment, and (3) the CDN administrator has no direct way of influencing the routing results.

In comparison, this design does not incur routing probes but rather monitor the regular distribution traffic to collect route information as a byproduct. This design also gives the CDN operator a great deal of flexibility in a large spectrum from the most automation to the most user control, simply based on the size and number of “locations” the CDN administrator defines. We find that most customers define a 4-tier overlay hierarchy with as many leaf locations as they have branch offices. Corporations that are more global may define up to 6 tiers.

□ FastForward

FastForward [5] is another commercial CDN design. It is primarily an application-layer broadcast network based on “publish and subscribe”. In FastForward, the viewers tune in to receive the broadcast and the broadcast is delivered live to the network edge primarily via hierarchical unicast and marginally via multicast relay much like Mbone [6], via layer-3 and layer-4 tunneling.

Compared to our design, FastForward does not maintain any explicit notion of user-configured CDN multicast group for the purpose of content routing, central management, multicast security, and network monitoring. Secondly, FastForward concerns primarily the live media delivery to the end-user, while our system is a stored-and-forward network for preposition content ahead of time to facilitate video-on-demand (VOD) from the edge. The delivery to the end-user is separate and can be either live or VOD. In the case of live, our system can route the live stream through the overlay system the same way as described in Section 4, hence leveraging both IP multicast and IP unicast still. Lastly, FastForward pieces together IP multicast islands via layer-3 and layer-4 relays, while our

system explicitly manages IP multicast from application layer instead of the network layer.

□ Overlay network

Overlay network research has been focused on providing a one-to-many content delivery service via the construction of an overlay network where each hop in the overlay is a unicast TCP link between two nodes, and leads to the network edge. See publications [7] [8].

Our architecture differs from the typical overlay networking research in that we are trying to make native IP multicast work in the overlay network while the typical overlay network research uses peer-to-peer or hierarchical unicast to provide a one-to-many delivery service, which is often also referred to as multicast or application-layer multicast.

In essence, our architecture improves upon existing overlay network to take advantage of both IP unicast and IP multicast in providing the one-to-many delivery service. The architecture also solves the integrated content routing problem between unicast and multicast, as well as application-layer traffic control.

7 Conclusion

In this paper we described an overlay network design that provides integrated content distribution routing between IP multicast and unicast via user-configured “multicast clouds”. This paper embodies the principles in application-layer multicast as detailed in [2] and the major improvements based on our experience with SODA [4].

CDN designers are often faced with the decisions between fully automated and fully user controlled content routing, between IP unicast and IP multicast. While previous work takes advantage of one or the other approach and suffers the drawbacks of either approach, our design takes the best of both worlds and let them mitigate each other’s drawbacks. Wherever available, IP multicast helps the CDN utilize physical broadcast media, build efficient distribution, and improve overall CDN scalability, while IP unicast bridges together IP multicast islands and provides failover for multicast.

In this design, the CDN administrator configures the overlay topology and multicast clouds, providing as much or as little topology guidance as the administrator desires. With the guidance from the user configuration, the channel routing algorithm dynamically and distributedly computes the best distribution traffic path through the network, taking into account the changing network conditions. Hence, our system gives the CDN operator a lot of control over the form of the distribution tree, and yet still automate most of the route selection and failover process. Especially, the design has the flexibility to function in either extremes of the spectrum as well as in between the extremes, from fully

user configured to fully automated, depending on the number and the size of the overlay locations the CDN operator specifies.

Furthermore, the CDN administrator not only influence the topologic directions of the traffic flow but also control many other aspects of the distribution traffic, including distribution priority, bandwidth limits, quality of service and multicast security. Our overlay design makes it possible to abstract the layer 3 and layer 4 traffic control parameters and translate into meaningful application-layer controls, making it much easier to use.

Our CDN builds a channel-specific distribution tree via a combination of IP unicast and IP multicast so as to reach across unicast IP networks to bridge isolated multicast IP islands. This is not a mere addition of unicast and multicast. More importantly, such integrated routing provided synergy between unicast and multicast content distribution. The multicast forwarders form a parallel unicast hierarchy next to the multicast cloud, providing scalability, reliability, and failover to the multicast data transfers, in the event of multicast loss repair and multicast failure.

This is a proven architecture, with efficiency, intelligence, and ease of use. We have successfully implemented this system and deployed in networks ranging from a few hundred to a couple thousand CEs. We are continuing to improve the system for even larger deployments.

References

- [1] Paul Francis , "Yoid: Extending the Internet Multicast Architecture", <http://www.aciri.org/yoid/docs/index.html>
- [2] D. Li and J. Jannotti, “Application-layer Multicast and Enhancement with IP Multicast”, <http://www-cs.stanford.edu/~dli/app-mcast-paper.pdf>
- [3] Li, D.; Cheriton, D. R.; "OTERS (On-Tree Efficient Recovery using Subcasting): a Reliable Multicast Protocol" 6th IEEE Intl Conference on Network Protocols (ICNP'98). Oct. 1998.
- [4] J. Jannotti, D. K. Gifford, K. L. Johnson, M. Frans Kaashoek, J. O'Toole Jr.: Overcast: Reliable Multicasting with an Overlay Network. OSDI 2000:
- [5] FastForward, <http://www.cs.ucsb.edu/ngc2000/program/invited-francis.ppt>
- [6] Mbone, <http://www-itg.lbl.gov/mbone/>
- [7] Overlay network designs: <http://citeseer.nj.nec.com/jannotti00overcast.html> <http://www.cs.berkeley.edu/~boonloo/classes/cs268/cs268.PDF> http://www.eurecom.fr/~btroup/RESEARCH/TOPICS/w_s_erkam.htm <http://nms.lcs.mit.edu/ron/> <http://www.arl.wustl.edu/~sherlia/amcast.html> <http://www.cs.virginia.edu/~hw6h/cs793/readlist.htm>
- [8] "S. Shi and J. Turner. “Routing in Overlay Multicast Networks”. IEEE INFOCOM 2002.