Overview of Overlay Multicast Protocols

Dennis M. Moen
C3I Center
George Mason University
dmoen@gmu.edu

Introduction

Multicasting remains a critical element in the deployment of scalable networked virtual simulation environments. Multicast provides an efficient mechanism for a source of information to reach many recipients. Traditional multicast protocols such as those defined by rfc 1075 [1] and rfc 2362 [2], provide mechanisms to support single source to a large number of destinations typically associated with streaming media or distribution of large volume data information distribution. In real-time collaborative and virtual simulation environments, the requirement is for many senders to send to the same destination group(s) simultaneously. This is commonly referred to as many-to-many multicast [3].

Even though IP multicasting was introduced more than 20 years ago, it still is not widely available as an open Internet service even for one-to-many multicast [4]. The most widely used multicast capability is the MBone [5]. The MBone provides a circuit overlay inter-network that connects IP multicast capable islands by using unicast tunnel connections and is commonly used in university and research environments. Only recently have public carriers started to introduce multicast services, but then only as a private network offering where all interested parties obtain service from the same carrier [6]. These new services are based on traditional multicast services providing one-to-many multicast.

Because open multicast services generally have not been available, there has been a shift to the idea of an end-host service to provide similar capabilities. By organizing end hosts into an overlay to act as relay agents, multicast can be achieved through message forwarding among the members of the overlay using unicast across the underlying network or Internet. Two general approaches have been proposed to accomplish this. One is peer-to-peer networks that were originally designed for information sharing and messaging such as Napster and Gnutella [7]. The second approach has focused on overlay multicasting to support group communications. Here, a transport-layer overlay, on top of the underlying Internet, between the members of a multicast group establishes group communications [8].

The fundamental difference between these two approaches is that in peer-to-peer networks, the topology tends to be random relative to the underlying physical topology which results from the loosely coupled relationship between the peers. The impact on the service is that latency can be very high as information might pass across many peers some of which might be slow as well as have long physical paths between them in the underlying network. Also, large periods of message flooding can occur in peer-to-peer networks which can cause congestion and inefficient use of network capacity. By contrast, an overlay multicast protocol can be more centrally controlled by managing the resources of a service node and by efficiently managing link stress. In this case, link

stress is the number of times a message transmits across the same underlying network link.

The overlays are constructed from two different strategies: mesh or tree. The mesh strategy provides for more than one path between a pair of nodes. In the tree case, a single path is established between any pair of nodes. It is also feasible to apply a mesh-first, followed by a tree construction algorithm to implement overlay multicast where the idea is to take advantage of both strategies.

There are distinct differences in these two strategies that directly impact the control mechanisms of implemented overlay protocols. Tree overlays are sensitive to partitioning of the overlay because they are acyclic graphs. A graph that contains no simple cycles is defined to be acyclic where a simple path is a path that contains no repeated arcs and no repeated nodes, except the start node and the end node are the same [9]. This means that if any non-leaf member of the overlay tree leaves the overlay, voluntarily, or by failure, the tree is broken and there will be no way for members of the multicast group to communicate. The clear advantage of trees is that, inherently, there are no routing loops formed during tree construction. This greatly simplifies the routing algorithm.

Mesh based overlays provide multiple or redundant connections between members of the group. This means that the overlay is less likely to be partitioned by node failure or departure. Alternate paths will already exist without the need to re-construct a path as is the case in a tree overlay. This certainly has advantages when considering needs for routing stability and offering quality of service (QoS) in the overlay. The down side to the mesh is that it is necessary to run a routing algorithm for construction of loop-free forwarding paths between group members [8] such as a path vector algorithm. Mesh overlays may also result in some inefficiencies as more than one copy of a message may use a link in the forward direction, e.g. link stress increases. This is not the case in a tree nor is it necessary to run a routing algorithm once the tree is established in order to prevent loops.

Traditional tree approaches use core based or route point based approaches for forwarding messages. This approach works well for one-to-many multicast. The idea is that a sender that desires to send a message to the multicast group sends the message to the core of the tree or the route point node, which in turn then forwards the message along the tree to all receivers. There is some inefficiency that results because all sender messages must first be routed to the core or route point before distribution across the tree. Current IP layer multicast routing generally uses this approach. The network inefficiency can be overcome by using source based tree algorithms in which each source builds the optimal routing tree from the source to all receivers in the group, however, this approach results in more overhead as each node must now run a routing algorithm and maintain larger amounts of supporting information. Though storing and managing larger amounts of information is easier to accomplish on an overlay host than on an ordinary network router where processing and information resources tend to be more limited.

Another important aspect of overlays is whether or not they are constructed with knowledge of the Internet topology. An awareness of the underlying Internet topology improves the efficiency of the overlay. Data forwarding in overlay networks is done at the application level. Therefore, data may traverse the IP network several times before it reaches its destination or destinations. This may result in inefficient use of network capacity and increased delays compared to transmission at the IP layer. This disadvantage

is reflective of all overlay protocols but is least pronounced if the overlay network is constructed with respect to the underlying Internet topology.

There are a couple of factors that influence the scalability of the overlay, where we define the scalability as the achievable size in terms of number of nodes or possibility overall performance like end-to-end latency. The number of nodes for example, is influenced by the amount of information that a node might need to retain. If the information needs of a node grow faster than the number of nodes in the overlay, then this very well becomes the limiting factor. Limiting node information to only knowing neighbors, not the entire overlay, allows greater scalability. The level of effort required to build and maintain the overlay can also influence scalability. While processing power and network capacity continue to grow, it is important to keep the overhead of the protocol in balance with the stated objective of efficient communications.
Typically multicast paths are unicast paths and are the shortest paths in term of hops. The resulting shortest-path trees are good for best-effort traffic. However, when QoS is considered, such shortest-path trees may not have the resources to support the quality requirement. Therefore, it is desirable to include other resource availability considerations in the overall optimization of best path for offering QoS.

Clearly, there are many alternatives and trade-offs for consideration in developing the optimal overlay multicast protocol and are represented in a large number of initiatives in this area. The Table below presents a summary of some of these initiatives that are in various states of experimentation and development. One observation is that it seems each of these efforts tends to focus on a specific optimization parameter that is reflective of a unique characteristic of a targeted application environment. While the intent of this research effort is not to draw a conclusion about this observation, it does however support the original proposal of this research. That is, that there are unique characteristics of the RT-DVS application environment that can be explored to enable open network overlay multicast services. The main characteristics of these applications are: real time, many-to-many, and receptive to network communication performance feedback. For example, unlike streaming video or streaming audio which are also real-time where the sender is not necessarily network aware and the transmission is one-to-many. Thus it is imperative to understand which combination of overlay strategies is optimal for RT-DVS such that the end systems cooperate to construct a good overlay structure to support many-to-many multicast.

To help answer that question, it is of value to review some of the most relevant efforts in overlay multicast protocol development. The row headings of the table indicate comparison criteria that reflect key performance elements for an overlay protocol. The criteria used for comparison are:

- Application: A general description of the targeted application environment, e.g. message information exchange, query, conferencing, streaming video.
- Overlay Topology: The term describes the nature of the organization of elements in the network. Examples would be, mesh, tree, ring, or multi-tier.
- Routing Algorithm: The routing algorithm refers to the specific algorithm used to develop the routing rules. Examples are, distance vector, Floyd's shortest path, Steiner tree, etc.

- Group formation: A general description of how groups might be formed and managed in the overlay.
- Scalability measures: A description of the scalability of the protocol and measures used for determination.
- QoS considerations: A description of quality of services that might be offered or are part of the guarantee of the protocol. Included are considerations for priority, message loss, and path failure and recovery mechanisms.
- Consideration for Node characteristics/resources: A discussion of whether the protocol considers the characteristics of a node in the development and dynamic management of the overlay. It is a recognition or consideration given to the ability of a node or host to act as an overlay relay agent.
- Node Join/leave/failures: A discussion of the technique associated with nodes joining and leaving the network either by choice or fault.

The desired outcome is to have a protocol that is QoS sensitive even though the underlying Internet is not able to provide services at a consistent QoS [10]. These comparison criteria are chosen as representative characteristics of a protocol that enable QoS sensitivity while being resource efficient and flexible. The criteria also represent areas or features that are typically traded off based on targeted application environment. In our application of interest environment, the distributed real time simulation applications require a protocol that support many-to-many multicast while being sensitive to end-to-end latency [3]. The environment must also be scalable to large number of users which implies a protocol sensitive to efficiency or minimum control messages and flexible to manage many multicast groups.

Mesh Overlays

Overlay meshes provide the underlays that allow message forwarding mechanisms between members or nodes of the overlay. Essentially these meshes provide managed tunnels between nodes across the underlying IP network. Various strategies are considered in performing establishment of these meshes including use of graphical shapes that have well known geometric routing principles as well as information about the underlying network or Internet.

Mithos [11] uses a geometric approach where the network is embedded into a multi-dimensional space, with every node being assigned a unique coordinate in this space. The geometric approach greatly simplifies routing as routing is easily enabled with knowledge of the local grid coordinates. Hypercast [12, 13] also uses this strategy. This approach uses properties of regular geometric shapes like rectangles, hypercubes, or Delaunay triangles to greatly simplify routing tables. In fact, in the cast of Hypercast, once the overlay is established, no routing protocol is necessary for the overlay.

In the rectangular approach, each node is assigned an enclosing axis-parallel multidimensional rectangle. Message forwarding is easily accomplished by sending to the rectangle abutting at the point where the vector to the destination intersects with the current node's rectangular boundary.

For the Delaunay triangle [12], links are established according to a Delaunay triangulation of the nodes and forwarding is accomplished similar to the rectangular

approach. Delaunay tangles main characteristic is that for each circumscribing circle of a triangle formed by three nodes, no other node of the graph is in the interior circle.

While Pastry [14] is a peer-to-peer based protocol, the substrate self-creates a messaging routing overlay on the Internet that operates in a way that makes the overlay look like a mesh. This is accomplished by each node having a unique 128-bit node ID. Using this unique ID, Pastry routes a message to the active node that is numerically closet. This approach provides a level of reliability since the idea is based on an active node. No further routing protocol is necessary for the local node to make this decision unlike a tree based approach were tree re-construction is likely required in the case of a node going inactive. Pastry also uses a metric for closest node such as latency so that optimum choice for forwarding is always made.

The HyperCast [13] protocol builds logical overlays based on geometric properties of a logical graph. HyperCast currently implements both the hypercube [13] and Delaunay [12] triangles. In each case, applications communicate with its neighbors in the geometric overlay, both in one-to-many multicast and many-to-one or incast. The key advantage to using geometric logical relationships is that once the overlay is established, there is no further need for a routing protocol. The key disadvantage of this approach is that the underlying physical network is completely ignored which makes it difficult to end-to-end latency considerations in performance. Another disadvantage is that hypercube overlays must be formed sequentially with the result that for a large set of nodes, it is likely that it will take a long time to construct the overlay and also complicates departure or joining of a single node. In the case of Delaunay triangles, overlay construction can be accomplished faster since as they can be built in a distributed fashion.

The logical hypercube overlay network topology organizes the applications into a logical n-dimensional hypercube. Each node is identified by a label (e.g., "010"), which indicates the position of the node in the logical hypercube. Message forwarding is easily accomplished by logical reference to nearest neighbor, hence no need for a routing protocol once the overlay is established.

A Delaunay [12] triangulation uses the special characteristic that for each circumscribing circle of a triangle formed by three nodes, no other node of the graph is in the interior of the circle. Each node in a Delaunay triangulation has (x,y) coordinates which depict a point in the plane. This approach allows each application to derive the next hop forwarding information without the need of a routing protocol.

Tapestry [15] is a unicast overlay network that provides the infrastructure for other multicast protocols like Bayeux [16]. The Tapestry routing mechanism is similar to longest prefix routing used in the CIDR IP infrastructure of the Internet [17], RFC 1518. The routing mechanism is a hash-prefix system which essentially results in every destination node being the root of its own tree that is the unique spanning tree across all nodes. The approach is inherently decentralized. Tapestry includes fault tolerant mechanisms that provide redundant paths to all destinations. This strategy effectively routes around failed nodes, in essence providing a mesh type infrastructure.

Tree Overlays

oSTREAM [18] is a tree based overlay that is specifically designed for one-to-many on-demand media distribution. The approach is to establish minimum spanning tree and use media buffering at the host to aid in the distribution of asynchronous service requests for the same streaming media. While this strategy is not particularly useful for a many-to-many environment, there are some similarities in forwarding the same information to asynchronous requests in the web services model. This might apply, for example, in the case where more static background information such as terrain models or weather information might be asynchronously requested from group members in a simulation. In these types of data distribution requests, there would be link and server efficiency advantages to using this approach.

Yoid [19] employs a single shared tree fro all members of the group. The links are unicast multi-router-hop paths. Trees are managed by a concept of child/parent relationship amongst members of the overlay. A member with no parent is the root member, and members with no children are leaf members and stub members are always leaf members. Each member divides the set of all other members into two groups called parent-side and child-side members. The groups are defined such that parent side members are all members reachable via the parent and all others are child-side members. Each member must mange this and understand how to protect from partition and make decisions on a new parent of the tree is portioned.

Topology Aware Grouping (TAG) [20] exploits underlying network topology information to build efficient overlay tree networks among multicast group members. TAG uses information about path overlap among members to construct a tree that reduces the overlay relative delay penalty, and reduces the number of duplicate copies of a packet on the same link. Each member of a TAG multicast session determines the path from the root of the session to itself and determines its parent and children. TAG nodes need only the IP addresses and paths of their parent and children nodes. TAG is unusual in that it constructs its overlay tree based on delay and considers bandwidth to break ties among paths with similar delays when constructing the overlay. This approach has merit for consideration as it provides the opportunity for guaranteeing end-to-end latency performance for the overlay. TAG does this by taking advantage of the underlying network shortest path topology information maintained in the underlying network IP routers.

The Overlay Multicast Network Infrastructure (OMNI) [21] is a two-tier approach to overlay multicast. The lower tier consists of a set of devices or service nodes that are distributed throughout an underlying network infrastructure like the Internet. The lower tier provides data distribution services to any host connected to an OMNI node over a directed spanning tree rooted at the source OMNI node. An end-host subscribes with a single OMNI node to receive multicast data service. The OMNI nodes organize themselves into an overlay which forms the multicast data delivery backbone. For the second layer, the data delivery path from the OMNI nodes to its clients is independent of the data delivery path used in the overlay backbone. This path can be built using network layer multicast, application-layer multicast, or a set of unicast paths.

Overcast [22] is a single source multicast overlay designed for on-demand and live data delivery. The protocol is single source tree based with some added features to

provide reliable delivery to multicast groups. Reliability is provided by using TCP e.g., HTTP over port 80.

Hybrid Mesh-Tree Overlays

As indicated earlier, there are two basic methods for construction of overlay trees for data delivery. First, one can construct a tree directly by members selecting their parents from amongst group members that they know. The second is to construct a well connected mesh of the group members and then use standard shortest path tree construction algorithms to establish the minimum distribution spanning tree. Protocols such as Narada [23] apply a two step process where in the first step and overlay mesh is constructed and then a tree is constructed using a shortest path algorithm on the nodes of the mesh.

While the mesh construction can be accomplished in an arbitrary fashion relative to the underlying infrastructure, there is value using knowledge of the underlying structure in building the mesh so as to improve over all performance of the overlay. The hypercube and Delaunay triangle techniques described above, for example, can easily be applied to build meshes without regard to underlying infrastructure. The technique might work well for peer-to-peer information exchange where end-to-end performance constraints are stringent.

Narada [23], however, tries to build the mesh in recognition of underlying network performance characteristics. Narada applies reverse shortest path algorithm in the second step to establish shortest path minimum spanning trees with each tree rooted at the source node. Several advantages result from this approach:

- Group management can be accomplished at the mesh level and more easily allows for the use of a standard group management protocol like IGMP at the local distributed level.
- Meshes are more resilient than trees; repair and optimization are easier to accomplish as loop avoidance is not required during this process.
- There are many existing algorithms for construction of shortest path trees on top of the mesh.

Another protocol that uses this general strategy is Tmesh [8], which uses an algorithm to determine shortcuts in the tree. The idea is to correlate measurable characteristics with a computed reduction in node-pair latencies attributed to each shortcut. This information is then used in a heuristic to select a shortcut with objective of improving overall latency.

Peer-To-Peer Overlays

There is another definition of overlay networking that is normally associated with information or message exchange at the application layer without using the services of an intermediary host, such as in a client server application, called peer-to-peer. Peer-to-peer is typically used to define an end point or application layer exchange of information. I have chosen not to treat them specifically as a separate category, as they apply routing principles similar to overlays in general, but the relationship requires some explanation as there are many developments in progress for implementation of self-organizing and

decentralized peer-to-peer overlay networks [24]. These efforts support new distributed applications which require information discover and message exchange across a network of loosely coupled applications.

The point-to-point overlays typically implement distributed hash tables that allow for location of an object within a bounded number of routing hops. They tend to exploit proximity in the underlying network topology in locating objects and routing. Multicasting is built on top of these peer-to-peer overlays. Examples of this strategy include Borg [24] and Scribe [25, 26] which are built on Pastry [14] and Bayeux [16] which is built on Tapestry [15]. Both Pastry and Tapestry provide unicast routing based on prefix-routing, and use a proximity neighbor selection mechanism to take advantage of the underlying physical network. Tapestry and Pastry also provide sensitivity to QoS by constraining the routing distance per overlay hop, resulting in efficient point to point routing between nodes in the overlay mesh.

Scribe uses reverse-path forwarding and Bayeux uses forward-path forwarding. The reverse-path construction of Scribe causes many short links in the multicast scheme which provides for lower link stress than Bayeux. Borg uses a hybrid multicast scheme to take advantage of asymmetry in routing in structured point-to-point networks. Borg builds the upper part of a multicast tree using a hybrid of forward-path forwarding and reverse-path forwarding and leverages the reverse-path multicast scheme for its low link stress by building the lower part of the multicast tree using reverse-path forwarding.

Peer-to-peer overlays networks generally are unpredictable and therefore impact quality of delivery of messages [27]. Strategies such as message preservation priority in queues and expiration times are sometimes used to enable some level of QoS. An example is to discard messages that have reached expiration times, or at least give priority to those that have not.

Overlay Multicast Protocol Summary Table

Protocol	Characteristic	Comment
Narada [23]	**Application**	**Not specific**
	Topology	**Mesh first. Begins with no knowledge of the underlying topology and over time continually refines the overlay as more information about the physical topology is obtained by probing and measuring latency**
	Routing Algorithm	**Distance vector Path algorithm on top of the mesh**
	Group Formation	**Shares group information with neighbor nodes. The mesh creation and maintenance algorithms assume that all group members know about each other and, therefore, does not scale well to large groups.**
	Scalability Measures	**No consideration to node degree or link stress—it is what it is**
	QoS Considerations	**The mesh is dynamically optimized by performing end-to-end latency measurements and adding and removing links to reduce multicast latency**
	Consideration for Node characteristics/resources	**n/a**

Protocol	Characteristic	Comment
	Node Join/leave/failures	Randomly chooses nodes for consideration to join and over time continues improvement. Maintains time of last neighbor pulse/keep alive message in a queue and responds to time outs for discovering dead nodes/connections
Yoid [19] ACIRI	Application	Data gram or stream for Peer-to-peer
	Topology	Tree-mesh
	Routing Algorithm	Root based trees and includes formation of clusters
	Group Formation	DNS reference. Each group generates IP multicast address produced via a hash of the group ID and related information
	Scalability Measures	Hop by Hop transport. Unicast is used when more than one hop exists between nodes. Multicast is used in the LAN.
	QoS Considerations	n/a
	Consideration for Node characteristics/resources	n/a
	Node Join/leave/failures	Tree reconstruction
oStream [18]	Application	On-demand media streaming (asynchronous)
	Topology	k-array tree
	Routing Algorithm	Minimal spanning tree-source based
	Group Formation	Based on request for like streaming media with adjustments to time of request
	Scalability Measures	Server capacity driven for buffering media
	QoS Considerations	Data buffering at relay nodes and at end host
	Consideration for Node characteristics/resources	Considers storage capacity for buffering
	Node Join/leave/failures	Tree joining is a local decision which implies only partial knowledge of the tree and recovery from node leaving is also accomplished locally.
Mithos [12, 28]	Application	Peer-to-peer applications
	Topology	Geometric relationship based on distance where a unique ID is assigned that is used in the formulation of routing table. Geometries used could include rectangle, Delaunay triangle, quadrant-based, or closest to axis
	Routing Algorithm	Geometric. Measures distance to nearest neighbors.
	Group Formation	n/a
	Scalability Measures	n/a
	QoS Considerations	n/a
	Consideration for Node characteristics/resources	n/a
	Node Join/leave/failures	Only interested in nearest neigh. Uses geometric relationship
Tmesh [8] Univ of Michigan Closely related work is something called Jungle	Application	Peer-to-peer
	Topology	Tree/mesh hybrid
	Routing Algorithm	Starts with an initial overlay and then uses Dijkstra shortest path first to discover shortcuts in the tree. For small number of members, not necessary to run a routing protocol.
	Group Formation	
	Scalability Measures	Adds shortcuts to improve tree delay performance

Protocol	Characteristic	Comment
Monkey	QoS Considerations	End-to-end path delay measurements to improve tree performance
	Consideration for Node characteristics/resources	Discovers shortcuts in the tree such that efficiency and performance is improved.
	Node Join/leave/failures	Since it uses a mesh, needs only to recover from tree partitions
Scribe [25, 26] Microsoft	Application	Messaging system for topic centric publish/subscribe messaging. Peer-to-peer
	Topology	Tree built on top of Pastry peer-to-peer network
	Routing Algorithm	Reverse-path forwarding Per group multicast spanning tree on top of Pastry nodes.
	Group Formation	Formation based on information topic subscription. Rendezvous point associated with a unique group ID To create a group, a Scribe node asks Pastry to route a create message using the group Id as the key. The node responsible for that key becomes the root of the group's tree.
	Scalability Measures	Large numbers of members per group
	QoS Considerations	none
	Consideration for Node characteristics/resources	None. Relies on Pastry to optimize the routes from the root to each group member based on some metric (e.g., latency).
	Node Join/leave/failures	Keep alive messages to children. Message loss on failure of a single node. Local restoration of subscribers required in case of node loss.
Pastry [14]	Application	Peer-to-peer
	Topology	Geometric (P2P) object location) based on nearest neighbor in terms of delay. Assigns a "proximity" metric that reflects the distance between a pair of nodes. Maintains a routing table with information about the distant nodes and a leaf set containing its direct neighbors.
	Routing Algorithm	Geometric based on nearest neighbors using unique node ID based on a secure hash
	Group Formation	Formation based on information topic subscription. Each group has a key called the group ID, which could be the hash of the group's textual name concatenated with its creator's name.
	Scalability Measures	Each entry in the routing table maps a node ID to its IP address. The routing table maintained in each Pastry node is created and maintained in a manner that enables Pastry to exploit network locality in the underlying network.
	QoS Considerations	None
	Consideration for Node characteristics/resources	None
	Node Join/leave/failures	Keep alive messages. Message loss on failure of a single node. Local restoration of subscribers required in case of node loss
TAG [20]	Application	Applications with large numbers of members
	Topology	Tree

Protocol	Characteristic	Comment
	Routing Algorithm	Each new member of a multicast session determines the path from the root of the session to itself, and uses path overlap information to partially traverse the overlay data delivery tree and determine its parent and children. The path computed under current Internet routing protocols serves as the basis for building the overlay network
	Group Formation	
	Scalability Measures	
	QoS Considerations	TAG constructs its overlay tree based on delay (as used by current Internet routing protocols), but uses bandwidth as a loose constraint, and to break ties among paths with similar delays
	Consideration for Node characteristics/resources	exploit the underlying network topology information for building efficient overlay networks where "underlying network topology," means the shortest path information IP routers maintain
	Node Join/leave/failures	Parent and its children periodically exchange reach ability messages in the absence of data. When a child failure is detected, the parent simply discards the child from its FT, but when a parent failure is detected, the child must rejoin the session
OMNI [21]	Application	Single source media streaming applications
	Topology	Service provider deploys nodes in a network to act as overlay relay agents
	Routing Algorithm	Degree constrained average-latency algorithm results in directed spanning tree routed at source
	Group Formation	Initialization is a simple sort from the root based on latencies from the next node
	Scalability Measures	Degree bounded directed spanning tree
	QoS Considerations	Latency sensitive
	Consideration for Node characteristics/resources	Degree constrained and capacity constraints of the nodes.
	Node Join/leave/failures	Fixed nodes in the network that organize iteratively and adapt to changing network conditions.
Borg [24]	Application	Best effort delivery of peer-to-peer messaging
	Topology	Built on top of Pastry
	Routing Algorithm	Builds the upper part of a multicast tree using a hybrid of forward-path forwarding and reverse-path forwarding and leverages the reverse path multicast scheme for its low link stress by building the lower part of the multicast tree using reverse-path forwarding. The boundary nodes of the upper and lower levels are defined by the nodes' distance from the root in terms of the number of overlay hops.
	Group Formation	Creates a group by asking Pastry to route a create message using the group ID as the key
	Scalability Measures	Large number of nodes. Uses Pastry network with average number of routing hops of 6.
	QoS Considerations	Best effort messaging

Protocol	Characteristic	Comment
	Consideration for Node characteristics/resources	Dependent on Pastry
	Node Join/leave/failures	Node join/leave messages are sent towards the root using the multicast group ID.
QMRP [29]	Application	Internet
	Topology	Tree and mesh. Uses single path and multi path to improve QoS
	Routing Algorithm	Behaves similar to PIM and is decentralized
	Group Formation	New member inquires the session directory and sends a request message to the core. The message includes information about the QoS of the path.
	Scalability Measures	Maximum branching degree of 10
	QoS Considerations	Switches between single-path routing and multiple-path routing according to the current network conditions.
	Consideration for Node characteristics/resources	Detects termination of routing processes to improve responsiveness
	Node Join/leave/failures	Detects the failure as well as the success of routing without the use of timeout
Hypercast [12/13] UVA	Application	One-to-many, Peer-to-peer
	Topology	Mesh based on hyper-cubes or logical Delaunay triangulations with source based trees for the data paths
	Routing Algorithm	Logical address encodes routing information from which the next hop information can be calculated without the use of a routing algorithm or table.
	Group Formation	
	Scalability Measures	Worst case is Delaunay triangle with 6 neighbors
	QoS Considerations	No QoS. Does not account for underlying network topology. The result is potentially large delay between node pairs
	Consideration for Node characteristics/resources	Only upper bound for number of neighbors-6
	Node Join/leave/failures	Node leaving requires more time for adjustment than a node joining
Overcast [22] CISCO	Application	Single source streaming media content distribution,
	Topology	Tree
	Routing Algorithm	Overcast builds a single source-rooted multicast tree using end-to-end measurements to optimize bandwidth between the source and the various group members
	Group Formation	Single source multicast groups. Users join at a Overcast node
	Scalability Measures	Scalability of the root to handle large volume service requests
	QoS Considerations	Bandwidth
	Consideration for Node characteristics/resources	Bandwidth at the node
	Node Join/leave/failures	Maintains global status at the root of the distribution tree

Protocol	Characteristic	Comment
Tapestry [15]	**Application**	**Tapestry is a peer to peer, wide-area decentralized overlay routing and location network infrastructure. Each node can act as a server to store objects, a router to forward messages, and/or client as source of requests**
	Topology	**Mesh**
	Routing Algorithm	**Hash-suffix mesh and allows messages to locate objects and route to them across and arbitrary network. Essentially, every node is the root node of its own tree which is unique spanning tree to all nodes.**
	Group Formation	**Unicast network overlay service**
	Scalability Measures	**Routing is inherently scalable**
	QoS Considerations	**Path is linearly proportional to the underlying distance.**
	Consideration for Node characteristics/resources	**n/a**
	Node Join/leave/failures	**Distributed mesh provides inherent alternate paths.**
Bayeux [16]	**Application**	**Steaming multimedia applications Peer-to-peer**
	Topology	**Dependent on tapestry**
	Routing Algorithm	**forward-path forwarding**
	Group Formation	**Uses session ID. A new request is sent to the root indicated session to join and root node manages the membership**
	Scalability Measures	**Root of the tree is potential bottleneck and single point of failure**
	QoS Considerations	**Relative delay penalty and physical link stress**
	Consideration for Node characteristics/resources	**n/a**
	Node Join/leave/failures	**Implements four control messages: JOIN, LEAVE, TREE, and PRUNE**

References

[1] Waitzman, D., C. Partridge, and S. Deering, "Distance Vector Multicast Routing Protocol", IETF rfc 1075, November 1988.

[2] Estrin, d., D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", IETF rfc 2362, June 1998.

[3] Moen, Dennis, and J.M. Pullen, "Enabling Real-Time Distributed Virtual Simulation over the Internet Using Host-based Overlay Multicast", *Proceedings of the Seventh IEEE Workshop on Distributed Simulation and Real-Time Applications,* 2003, pp. 30-36.

[4] Deering, Stephen E., and David R. Cheriton, "Multicast Routing in Datagram Networks and Extended LANS," ACM Transactions On Computer Systems, Vol. 18, No 2, May 1990, pp. 85-110.

[5] Eriksson, H., "MBONE: The Multicast Backbone", Communications of the ACM, Vol. 37, No. 8, 1994, pp. 54-60.

[6] Chu, Yang-Hua, San jay G. Rao, Srinivasan Seshanand, and Hui Zhang, "Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture," pp. 55-67.

[7] Junginger, Markus Oliver and Yugyung Lee, "A Self-organizing Publish/Subscribe Middleware for Dynamic Peer-to-Peer Networks," IEEE Network Magazine, Vol. 18, No. 1, January/February 2004, pp. 38-43.

[8] Wang, Wenjie, David Helder, Sugih Jamin, and Lixia Zhang. "Overlay Optimizations for End-host Multicast", NGC02, ACM, 2002, pp. 154-161.

[9] Berstekas, Dimitr P., "Network Optimization: Continuous and Discrete Models", Athena Scientific, Belmont, Massachusetts, 1998.

[10] Yan, Shuqian, Michalis Faloutsos, and Anindo Banerjea, "QoS-Aware Multicast Routing For The Internet The Designing And Evaluation Of QoSMIC," IEEE/ACM Transactions on Networking, Feb 2002, Vol. 10, Issue 1, pp. 54-66.

[11] Waldvogel, Marcel, and Roberto Rinaldi, "Efficient Topology-Aware Overlay Network," ACM SIGCOMM Computer Communications Review, Vol. 33, No. 1, January 2003, pp. 101-106.

[12] Liebeherr, J., M. Nahas, and Si Weisheng, "Application-Layer Multicasting with Delaunay Triangulation Overlays", IEEE Journal on Selected Areas in Communications, Vol. 20, Issue 8, Oct 2002, pp. 1472-1488.

[13] Hypercast Team, "Hypercast 2.0 Design Document", http://www.cs.virginia.edu/~hpercast, Department of Computer Science, University of Virginia, 2002.

[14] Rowstron, Antony, and Peter Druschel. "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems", Proc. IFIP/ACM Middleware 2001, Heidelberg, Germany, November 2001.

[15] Zhao, Ben Y., Ling Huang, Jerry Stribling, Sean C. Rhea, Anthony D. Joseph, and John D. Kubiatowicz, "Tapestry: A Resilient global-Scale Overlay for Service Deployment", IEEE Journal on Selected Areas in Communications, Vol. 22, No. 1, January 2004, pp. 41-53.

[16] Zhuang, S. Q., B. Y. Zhao, A. D. Joseph, R. H. Katz, and J. Kubiatowicz, "Bayeux: An Architecture for Scalable and Fault-tolerant Wide-Area Data Dissemination", Proceedings of the Eleventh International Workshop on Network and Operating System Support for Digital Audio and Video, June 2001.

[17] Rekhter, Y., and Li, T., "An architecture for IP address allocation with CIDR," RFC 1518, http://www.isi.edu/in-notes/rfc1518.txt, 1993.

[18] Cui, Yi, Baochun Li and Klara Nahrestedt, "oStream: Asynchronous Streaming Multicast in Application–Layer overlay Networks ," IEEE Journal on Selected Areas in Communications, Vol. 22, No. 1, January 2004, pp.91-106.

[19] Francis, Paul, "Yoid Tree Management Protocol (YTMP) Specification", ACIR Center for Internet Research, Berkeley, CA, April 2000.

[20] Kwon, Minseok, and Sonia Fahmy "Topology-Aware Overlay Networks for Group Communication", ACM NOSSDAV 2002, pp. 127-136.

[21] Banerjee, Suman, Christopher Kommareddy, Koushik Kar, Bobby Bhattacharjee, and Samir Khuller, "Construction of an Efficient Overlay Multicast Infrastructure for Real-time Applications," IEEE 2003, pp. 1521-1531.

[22] Jannotti, John, David K Giffors, Kirk L. Johnson, M. Frans Kasschoek, James W. O'Toole, Jr. "Overcast: Reliable Multicasting with an Overlay Network", Cisco Systems.

[23] Chu, Uang-hus, Sanjay G. Rao, and Hui Zhang, " A Case for End System Multicast," ACM SIGMETRICS 2000, pp. 1-12.

[24] Zhang, Rongmei, and Y. Charlie Hu, "Borg: A Hybrid Protocol for Scalable Application-Level Multicast in Peer-to-Peer Networks," ACM NOSSDAV 2003, pp. 172-179.

[25] Castro, Miguel Peter Druschel, Anne-Marie Kermarrec, and Antony I. T. Rowstron "Scribe: A Large-Scale and Decentralized Application-Level Multicast Infrastructure," IEEE Journal On Selected Areas In Communications, Vol. 20, No. 8, October 2002, pp. 1489-1499.

[26] Castro, Miguel, "Scalable Application-Level Anycast for Highly Dynamic Groups", Microsoft Research, Cambridge, CB3 OFB, UK.

[27] Junginger, Markus Oliver and Yugyung Lee, "A Self-organizing Publish/Subscribe Middleware for Dynamic Peer-to-Peer Networks," IEEE Network Magazine, Vol. 18, No. 1, January/February 2004, pp. 38-43.

[28] Waldvogel, Marcel, and Roberto Rinaldi, "Efficient Topology-Aware Overlay Network," ACM SIGCOMM Computer Communications Review, Vol. 33, No. 1, January 2003, pp. 101-106.

[29] Chen, Shigang, and Klara Nahrstedt, "QoS-Aware Multicast Routing Protocol", IEEE Journal On Selected Areas In Communications, Vol. 18, No. 12, December 2000, pp. 2580-2592.