Chapter 1

Introduction

This dissertation presents a new Application Layer Multicast Protocol, called Appcast for providing multicast services over Internet. Multicast protocols aim at eliminating the redundant movement of data packets in a network. Application Layer Multicast protocols, construct overlay network for efficient data transmission over large networks like Internet. In Application Layer Multicast protocols, hosts take over the job of multicast routing, which is in contrast to IP Multicast [23] wherein routers do the additional job of multicast routing. The first part of this dissertation is about Application Layer Multicast protocols and Appcast. Application Layer Multicast protocols construct and manage overlays. The first protocol in this series is ESM [1-2] and is considered very inefficient from control and management perspective i.e. it has a very high control overhead of \( O(n^2) \). While protocols like CAN[18-19], DT Protocolp 1] do not consider the topology information at all, others like NICE[15-17], HMTP[13] etc could not exploit topology information to maximum. Our protocol constructs overlay in alignment of the underlying network topology and also keeps the control overhead to \( O(n) \). The second part deals with how to architect efficient multicast applications over Appcast. The third part is concerned about how TCP/IP can be improved to leverage upon the broadcast nature of the media for multicast applications. While Application Layer Multicast protocols are efficient in reducing the redundant data packet movements, they introduce slight delay compared to native IP Multicast. So, the fourth part of this dissertation shows how this delay can be minimized using efficient XPath[78] stream processing. New Internet standards like XML[80], SOAP[47-49] etc. have given us opportunities to architect multicast services over Appcast as an end-to-end solution.

1.1 State of the Art Technologies

Internet is popular for its efficiency in distributing information: mail, news, web pages (HTML), files of all types like jpg, mpg etc, chat channels, DNS records, audio and video broadcast and so on. While majority of the information that flows across Internet is for mass distribution of static content, the dynamic and two way interactive applications could not scale up to Internet. This is quite evident from DNS that used centralized
system ultimately moved to a distributed environment by arranging the DNS servers in a hierarchical fashion. The zone transfers replicate DNS information on to many servers, to make the two-way interactive query-response system of DNS work efficiently by keeping the responses nearer to users who make queries.

The 1999 Victoria's secret webcast with 1.5 million hits [105], ultimately could not serve any request. The 1998 distribution of Starr Report is another example. Servers like Netscape and major content distributors are forced to look into alternatives like distributed networks of servers, simply to carry the load. Akamai [106] is one such network, with almost 4200 servers across the globe located in 50 countries. To reduce the load on servers and increase the network efficiency, many schemes like Content Distribution Networks, Active Networks, IP Multicasting, Overlay Networks etc are proposed. In this section, we briefly describe each area of technology that greatly increases the utilization of information, bandwidth and computing resources on the Internet.

1.1.1 Load Balancing
Even though in recent years both network and server capacity have improved, response time continues to challenge researchers, as centralized web systems are not able to scale. Improving the power of a single server does not solve the web scalability problem in a foreseeable future. Another solution is to deploy a distributed web system [70] composed of multiple server nodes and distributing the load reaching this web site evenly among the server nodes, so as to improve system performance. Therefore, any distributed web system must include some component (under the control of the service provider) that routes client requests among the servers with the goal of maximizing load sharing. This component can be deployed at various levels of TCP/IP stack like - NAT at the MAC and IP Level [108]; round robin DNS at DNS level [109] and proxies at application level. Adding more nodes, complete with processors, storage and bandwidth, expands the system capabilities.

1.1.2 Active Networks
The design of inter-network protocols abstracts away how messages are routed or forwarded through out the network, keeping the applications away from the complexities
of underlying network. This design makes it hard for the applications to exploit detailed knowledge of the network in order to enhance their performance and reduce network load. This abstraction is actually degrading the performance of both the application and network [66]. *Active Networks*’ prime concern is to make the network dynamic and more intelligent by allowing new protocols and application code to be downloaded on to any network device - like routers, hosts, clients etc. The code modifies the behavior of router for particular type of packets specific to application needs. ANTS[107] is the most successful toolkit. Deployment problems and serious security issues are discouraging the ISPs to deploy *Active Networks*. A variation of *Active Networks* is *Active Services Framework* that recommends programmable service architecture. With this architecture, the core of the network remains unchanged, but it allows applications and users to download and execute code at strategic locations for the application i.e. code is executed on hosts instead of on routers.

### 1.1.3 Overlay Networks

*Overlay Networks* inherently use the concept of *Active Networks* [66]. In overlay networks, hosts form an overlay of unicast connections to cooperate and communicate over the general Internet infrastructure.

### 1.1.4 Content Distribution Systems

IP *Content Distribution Networks* (CDN) are special purpose overlay networks that provide scalability by distributing many servers across the Internet close to consumers. Consumers obtain content from these edge servers directly rather than from the origin server. One of the major techniques used for content distribution is *Caching*. *Caching* [68, 69] technique’s main aim is to relieve the source from servicing many connections. *Caching* allows storing the initial service responses in intermediary servers like proxy/cache servers, which will serve subsequent requests on behalf of the actual source server. The *caching* concept can be extended in hierarchical manner. Content distribution involves pushing the content onto these cache servers. PRISM’s content naming, management, discovery and redirection mechanisms [71] support high quality streaming media services in an IP based *Content Distribution Network*. *Internet Backplane Protocol* [72], or IBP supports logistical networking to allow applications to control the movement
and storage of data between nodes. A number of service providers (e.g., Adero, Akamai and Digital Island) operate content distribution networks, but in-depth information describing their internals is not public information.

1.1.5 Peer-to-Peer Network
As the computer and communication technology is advancing, the distinction between server and client is diminishing in terms of computing power and they become a network of peers. CDNs’ main purpose is to distribute content and is designed from the perspective of a distributor who establishes infrastructure and controls the same across Internet. In contrast, Peer-to-Peer network emphasizes on communal sharing of computer resources [73,74], without any controlling authority. Peer-to-Peer computing opposes traditional client-server computing. In Peer-to-Peer, an application is split into components that act as equals. Gnutella [110] has become the most successful development, though it is inefficient. Gnutella broadcasts thousands of messages per request. Peer-to-Peer systems provide limited support for applications that go beyond file sharing. It somehow, could not go beyond this fundamental application, probably due to its very objective of file sharing and unreliability of the protocols as peers come and go at unpredictable times.

1.1.6 Grid Computing
Networks’ communication speed is outpacing computers’ processing speed making communication essentially free. The doubling of network performance relative to computer speed every 18 months [75] has made the researchers think about many collaborative applications from different areas like life sciences - genomics; engineering - air craft control; physical science - astronomy, particle, nuclear; business – fraud detection, anti money laundering etc that are resource and process intensive. A grid is a kind of Peer-to-Peer network, in which an application can draw computational power equivalent to super computer, transparently from the grid. The grid allows sharing of resources like memory, processor etc on loosely coupled systems on Internet.

1.2 Multicast
Though much communication over Internet is for group distribution (inherently) in the form of replications, caching etc, every distribution application on the Internet runs over
the unicast infrastructure and derives its distribution functionality from application specific mechanisms. For instance, RFC 822 mail headers have mechanisms for detecting loops among mail forwarders[111]; NNTP has the \textsc{newnews} command [112] to manage flooding of news articles and HTTP has redirection and mechanisms for handling caches [113] and so on.

All the above applications are written independently and use the transport layer API and achieved some sort of multicast capability by sending the same information multiple times but to different destinations, though crossing the same routers at times, as no multicast infrastructure available over Internet.

The goal of multicast or broadcast mechanism is to eliminate redundant packet replication in a network, when a group of computers participate in a communication. Multicast can no more be just a choice but it is requirement as it can greatly improve the efficiency of Internet, by reducing unnecessary traffic movement. Multicast is an efficient mechanism that transmits only one copy of data to the receivers by replicating it as necessary while traveling through the Internet till it reaches the receivers. This is in contrast to unicast, where in multiple copies are sent by the source to the receivers, requiring much processing capacity at source and more bandwidth on network links.

1.2.1 Global Wide Area Multicast Applications
Internet RFC 3170 has listed many multicast applications[114]. One can define multicast communication as one that has parallel/simultaneous communication from/to a group of entities like hosts, people etc. Based on this definition we can arrive at three general categories of multicast applications:

\textbf{One-to-Many (1-to-M):} A single host communicates to two or more (n) receivers in this kind of multicast application.

\textbf{Many-to-Many (M-to-M):} In this model, any number of hosts can send information to the same group, as well as receive from it.

\textbf{Many-to-One (M-to-1):} Any number of hosts can send information to a single host in this type of multicast application.
1.2.1.1 One-to-Many (1-to-M) Applications

a) Scheduled audio/video (a/v) distribution: Lectures, presentations, meetings, or any other type of scheduled event like news (i.e. television and radio "broadcasts") etc fall into this category of applications.

b) Push media: News headlines, weather updates, sports scores etc applications represent this kind of multicast where in information is pushed onto users' appliances like TV, phone, computer etc. Relatively low-bandwidth data characterize these applications.

c) File Distribution and Caching: Applications like distributing web site content, executable binaries and other file-based updates to end-user or replication/caching sites etc use multicast.

d) Announcements: Network time, multicast session schedules, random numbers, keys, configuration updates etc applications can exploit multicast. Their bandwidth demands can vary, but generally they are very low bandwidth.

e) Monitoring: Stock prices, sensor equipment (seismic activity, telemetry, and meteorological or oceanic readings), security systems, manufacturing or other types of real-time information can reach users effectively using multicast. Bandwidth demands vary from constant-bit-rate to bursty (if event driven) traffic.

1.2.1.2 Many-to-Many (M-to-M) Applications

In many-to-many (M-to-M) applications two or more of the receivers also act as senders. In other words, M-to-M applications are characterized by two-way multicast communications.

The M-to-M capabilities of multicast enable the most unique and powerful applications. Each host running an M-to-M application may receive data from multiple senders while it also sends data to all of them. As a result, M-to-M applications often present complex coordination and management challenges.

a) Multimedia Conferencing: Audio/Video and whiteboard comprise the classic conference application. Having multiple data streams with different priorities characterizes this type of application. Co-ordination issues, such as determining who gets to talk when, complicate their development and usability. There are common heuristics and "rules of play", but no standards exist for managing conference group dynamics.
b) **Synchronized Resources**: Shared distributed databases of any type exchange information on schedules, directories etc in M-to-M way.

c) **Concurrent Processing**: Distributed parallel processing is an M-to-M process where in multiple processors collaborate with each other.

d) **Collaboration**: In shared document editing many users can edit the same document simultaneously.

f) **Chat Groups**: Chat group applications allow many users exchange information.

g) **Distributed Interactive Simulations [DIS]**: In this kind of applications, each object in simulation multicasts descriptive information (e.g., telemetry). So all other objects can render the object, and interact as necessary. The bandwidth demands for these can be tremendous, as the number of objects and the resolution of descriptive information increases.

h) **Multi-player Games**: Many multi-player games are simply distributed interactive simulations and may include chat group capabilities. Bandwidth usage can vary widely, although today’s first-generation multi-player games attempt to minimize bandwidth usage to increase the target audience (many of whom still use dial-up modems).

1.2.1.3 **Many-to-One (M-to-1) Applications**

Unlike the one-to-many and many-to-many application categories, the many-to-one (M-to-1) category does not represent a communications mechanism at the IP layer. M-to-1 applications have multiple senders and one (or a few) receiver(s), as defined by the application layer. M-to-1 applications have many scaling issues. Too many simultaneous senders can potentially overwhelm receiver(s), a condition characterized as an "implosion problem.

a) **Resource Discovery**: Service location, for example, leverages IP Multicast to enable something like a "host anycasting service" capability [AnyCast in RFC 1546]: A multicast receiver to send a query to a group address, to elicit responses from the closest host so that they can satisfy the request. The responses might also contain information that allows the receiver to determine the most appropriate (e.g., closest) service provider to use.

b) **Data Collection**: This is the converse of a 1-to-M "monitoring" application described earlier. In this case there may be any number of distributed "sensors" that send
data to a data collection host. The sensors might send updates in response to a request from the data collector, or send continuously at regular intervals, or send spontaneously when a pre-defined event occurs.

c) **Auctions:** The "auctioneer" starts the bidding by describing whatever it is for sale (product or service or whatever), and receivers send their bids privately or publicly (i.e., to a unicast or multicast address) in auction applications.

d) **Polling:** In this kind of applications, the "pollster" sends out a question, and the "pollees" respond with answers.

### 1.2.2 IP Multicast

IP Multicast service [8, 23-31, 33] is proposed as an efficient multi-packet delivery mechanism. IP Multicast is a bandwidth conserving technology that reduces traffic by simultaneously delivering a single stream of information to thousands of corporate recipients and homes. To reduce the complexity and address the shortcomings of IP Multicast, SSM [32] - Source Specific Multicast has been developed. IP Multicast is designed at a very low level as a network primitive and is already deployed over major portions of Internet and is an inherent part of IPv6. IP Multicast delivers datagrams to logical addresses known as multicast groups identified by Class-D addresses. IGMP[115] protocol manages the group management and routers construct distribution trees for each group by using multicast routing protocol like DVMRP[116]. Though, majority of the routers on Internet are IP Multicast enabled, very few applications take advantage of the same, as not many developers are aware of multicast API and are conversant with network technology.

### 1.2.3 Application Layer Multicast

Application layer multicast protocols, construct overlay network for efficient data transmission. E-mail distribution and USENET news distribution are early Internet applications that fall into the broad general category of **Application Layer Multicast.** Application Layer Multicast protocols do not change the network infrastructure. These protocols implement multicast forwarding mechanism at the end hosts. These kinds of protocols are being designed extensively for content distribution networks. Unlike native multicast, where data packets are replicated at routers inside the network, in application
Layer multicast, data packets are replicated at end hosts. Many researchers are finding ways to push the multicast routing capabilities onto hosts than onto routers. The idea here is to construct an overlay network of hosts, on top of routers. Routers do normal unicast routing, while the hosts attached to them will take care of multicast group management and routing among themselves. While IP Multicast protocol works at network layer, that requires applications to be aware of multicast and programs [76] be written specifically using IP Multicast API; protocols like End-system Multicast [1-2], Scattercast [3-4] and Overcast [5] require an application-layer overlay infrastructure to be laid off. All these protocols, share the common goal of providing the benefits of multicast without requiring direct router support or the presence of a physical broadcast medium [4].

1.3 Problem Definition
We aim at improving the network performance on the existing inter network infrastructure. We mainly look at improving performance and simplifying the multicast application development by taking four different tasks.

1.3.1 Exploiting the Topology
It is quite evident that when a host has to send information to multiple stations, the packets travel over common links. The figure 1.1 shows unicast and overlay multicast topologies.

The goal of multicast is to reduce the redundant packet movement. While there are many overlay multicast protocols like [11] JDT Protocol, CAN[18-19] etc that do not consider underlying network topology, we create an overlay in alignment with the underlying
topology, such that the overall link utilization is optimum. For instance, in figure-1, the link R1-R2 is very costly, and if the overlay2 is created as A->C; C->B and B->D then this overlay becomes more inefficient compared to unicast. This is because, in unicast, the R1-R2 link is utilized twice, whereas in overlay it is used thrice.

We propose to build an overlay such that the overall link utilization in the overlay is kept optimal i.e., from the network perspective, the constructed overlay must ensure that redundant transmission on physical links is kept minimal. In overlay multicast, hosts take over the routing functionality of normal network routers and forward the packets among the participating hosts. The topology-building algorithm has to consider each joining hosts’ capacity and each host can specify to how many hosts it can forward the data in the overlay. As the overlay grows, the delay increases for the data to reach from the source to the destination, as the number of intermediary routing hosts increase. So, each host can specify how much delay it can sustain. The protocol must allow for the overlay to incrementally evolve into a better structure as more information becomes available.

1.3.2 Exploiting Media Characteristics

The Transmission Control Protocol/Internet Protocol (TCP/IP) standard is the dominating protocol in the communication world. TCP/IP standard has achieved this dominance, and become a de facto standard for any network, because its superiority in handling high-speed data traffic has been proved over a long period in the largest global network: the Internet.

In principle, broadcasting via satellite is simpler than for terrestrial networks, which are not naturally broadcast networks. However, TCP/IP has shielded this nature. TCP/IP follows a layered approach in which the lower level protocol details are shielded by high-level protocols. Broadcast nature of the media, is a physical line property, which falls into physical layer in OSI layered model that naturally gets shielded by TCP - transport layer i.e., 1000 hosts, which are connected by the same satellite media, though can communicate to all by just transferring the data once, due to lack of availability of this specific media property to the application developers, applications are written to transfer the data 1000 times as if the hosts are connected through 1000 different network links.
Geo-stationary satellite systems have a fixed round trip delay of 600ms that cannot be avoided. The problem with TCP/IP over satellite links is that the protocol will not let data be sent, beyond a certain limit or window, without an acknowledgement from the receive end. Thus if the traffic is affected by latency, the system has to permit the data flow to increase, although unacknowledged, and allow the receiver to control it. This control is done by using a receive buffer whose parameters are sent to the transmit end during the set up phase so that it can adjust the traffic flow accordingly. The buffer size is reduced each time data is transmitted, so that when it is empty no more data can be sent without a TCP acknowledgement. The window data rate limit is set because the TCP/IP only has 16 header bits available to describe the packet size, which sets a maximum throughput of 216 bytes, corresponding to 840kbit/s. Satellite vendors using a technique called 'TCP Spoofing' have solved this latency problem of satellites communication.

While there are many vendors improving TCP efficiency over satellite for only specific applications like video broadcast etc, there is no way for a general application developer to exploit the broadcast property in his application. We propose a method 'Multicast Spoofing', using which we pass on the broadcast benefit to any application.

1.3.3 Exploiting Query Redundancy
Search tools till date have insufficient capabilities to keep pace with the information generated. Particularly getting right and relevant information has become a nightmare. In this scenario, SDI - Selective Dissemination of Information, a popular concept used by libraries is gaining importance. In SDI, users register with servers that are nearer to them with their interests, which are stored as profiles. Based on the profiles, the servers filter right information and push the same to the user. A centralized server cannot scale up to the requirements of large number of users spread all over Internet.

Overlays can help meet this requirement, however, we have to devise ways to reduce the latency/delay introduced by overlays. Figure 1.2 shows an example of overlay. Stream processing allows data to be processed by intermediate servers as data streams in thereby not adding any delay at processor side. Using overlays, we can register profiles with the servers nearer to the users (consumers) and these servers in turn send the combined profiles to the servers above them. This profile grouping and sending can further go up the tree till the producer server.
Whenever the producer sends information, this has to be filtered as per their profiles at each server at lower levels of the tree, till it reaches the consumer. Data is exchanged in the form of XML[80] in SDI systems and each profile is represented in the form of an XPath[78] query and we filter the information based on multiple profiles (queries), registered at each server.

When thousands of profiles are grouped together, naturally there will be lots of commonality in the kind of information in which the users are interested. Our objective is to exploit this redundancy such that while filtering the information as per the profiles, we need not traverse the same information for thousand times and rather we should filter the information for all users in just one pass.

1.3.4 Generic Application Development Framework

Our aim is to provide broadcast/multicast advantage to any kind of application that sends same information to many receivers, without writing separate applications - one for broadcast and the other for unicast. Unlike point-to-point communication, where sender and receiver only co-operate, in broadcast communication sender and many receivers must co-operate. Naturally, not all receivers are in homogeneous environment. Lot of following for general applications protocols like SMTP, HTTP, FTP, Telnet etc made easy for deployment on any platform because almost every vendor has a product for it that can interact with other similar products just because all of them follow protocol specifications and API. But the same is not the case for other business specific
applications. Applications written for one platform are now not able to communicate with other applications on other platforms due to many reasons like dependency on specific language, communication protocol, database etc. Therefore we looked for a platform independent distributed computing environment. We rely on standard SOAP [47-49] - Simple Object Access Protocol specification that is fast becoming the standard middleware to provide a generic simple to use architecture for multicast applications.

1.4 Contributions
The main contribution of this thesis is to give a simple DNS kind of infrastructure based solution for multicast services. We emphasize that multicast can better be provided at application level rather than at router level.

Application Layer Multicast (Appcast): We propose a new application layer multicast that exploits the topology information. Any application layer multicast protocol that does not exploit topology may in fact decrease the efficiency in terms of bandwidth and increase delay. We develop new algorithm Appcast that creates a multicast overlay topology, which we further optimize by two more extensions to it taking delay and processing power as parameters.

Advantage Broadcast Media: The application developers are not able to take advantage of the media nature like broadcast for satellite networks. We show ways of how the broadcast nature can best utilized by developers.

A Generic Application Architecture for Multicast Services: We design generic application architecture and develop three different kinds of applications. We take 'scheduled file pushing' application as one-way multicast application, 'database replication' as event and rule based one-way multicast application and 'auctions' as interactive multicast application and implement all three.

Xpath Query Processing (YALXP): Hosts are organized as an overlay in application layer multicast. Overlays disseminate information in a hierarchical and incremental fashion. Also, not all members in an overlay or multicast group need exactly same
information. In this case, if we can filter the information at different levels of overlay, while the information is distributed, we can minimize the bandwidth and processing requirement. Each member's interests are represented by XPath[78] query and the information to be distributed is in the form of XML[80] document as XML is gaining importance as the standard for information exchange. We look at the XML document-processing problem particularly in the context of application layer multicast and propose a new XPath processing algorithm called YALXP. YALXP algorithm processes multiple XPath queries over a XML document in single document traversal i.e. in one pass it can answer multiple queries.

1.5 Thesis Outline
We organize the thesis as follows. Chapter 2 does a survey on already proposed application layer multicast protocols. It classifies them based on criteria like type of algorithm - centralized, distributed; type of topology created - tree, mesh etc; and analyses their merits and demerits. In Chapter 3 we propose a new algorithm Appcast. We simulate and compare the performance of Appcast and other closely related application layer multicast protocols like TAG [22], NICE [15-17] and HMTP [13]. Also, Chapter 3 studies various ways of enhancing the TCP/IP over broadcast networks and details the approach we proposed i.e., 'multicast spoofing'. Chapter 4 deals with generic multicast application architecture design, development and implementation. In Chapter 5, we propose new XPath[78] processing algorithm YALXP. In this chapter we surveyed the existing XPath processors, simulated algorithms like XAOS [79] and compared them with our algorithm. We conclude the thesis in Chapter 6 with discussion on future work,