Basics of Multicast technology

from
“State-of-the-art in group communications: from protocols to applications”,
tutorial given at ICT 2003, Papeete, French Polynesia
February 23rd, 2003
and
“Multicast Technologies: past, present, future”,
tutorial given at DFMA 2005, Besançon, France
February 6th, 2005

C. Pham
University Claude Bernard Lyon 1
LIP/INRIA RESO
Congduc.Pham@ens-lyon.fr
www710.univ-lyon1.fr/~cpham
From unicast...

- Sending same data to many receivers via unicast is inefficient
- Popular WWW sites become serious bottlenecks
...to multicast on the Internet.

- Not n-unicast from the sender perspective
- Efficient one to many data distribution
- Towards low latency, high bandwidth

Introduction
The delivery models (1)

- model 1: streaming (e.g. for audio/video)
  - multimedia data requires efficiency due to its size
  - requires real-time, semi-reliable delivery

Introduction
The delivery models (2)

- **model 2: push delivery**
  - synchronous model where delivery is started at $t_0$
  - usually requires a fully reliable delivery, limited number of receivers
  - **Ex:** synchronous updates of software

![Diagram showing delivery process]

- **t0, tx starts...**
- **time**
- **receiver ready...**
- **receiver ready...**
- **ok, receiver leaves**
- **ok, receiver leaves**

**Introduction**
The delivery models (3)

- **model 3: on-demand delivery**
  - popular content (video clip, software, update, etc.) is continuously distributed in multicast
  - users arrive at any time, download, and leave
  - possibility of millions of users, no real-time constraint

![Diagram of delivery models](image-url)
Basic of the IP multicast model

Early group management

IP multicast routing

First steps in reliability
Multicast BONE at the ENS Lyon

Basics IP multicast
MBone tools - RAT

- The Robust Audio Tool (RAT) is an open-source audio conferencing and streaming application that allows users to participate in audio conferences over the internet. These can be between two participants directly, or between a group of participants on a common multicast group.
MBone tools - VIC

- **VIC** is a video conferencing application developed by the Network Research Group at the LBNL in collaboration with the University of California, Berkeley.
WBD is a shared whiteboard compatible with the LBL whiteboard, WB. It was originally written by Julian Highfield at Loughborough University and has since been modified by Kristian Hasler at UCL.
MBone - Advertising sessions

- **SDR** is a *session directory* tool designed to allow the advertisement and joining of multicast conferences on the Mbone. It was originally modelled on **sd** written by Van Jacobson at LBNL.
A look back in history of multicast

- **History**
  - Long history of usage on shared medium networks
  - Resource discovery: ARP, Bootp.

Timeline:
- 1973: Ethernet radio network
- 1983: ARP (RFC 826)
- 1985: Bootp (RFC 951)
- 1986: Deering's work IP multicast (RFC 966, 988, 1054, 1112)

Basics IP multicast
The Internet group model

- multicast/group communications means...
  - $1 \rightarrow n$ as well as $n \rightarrow m$
- a group is identified by a class D IP address (224.0.0.0 to 239.255.255.255)
  - abstract notion that does not identify any host!
The group model is an open model

- anybody can belong to a multicast group
  - no authorization is required
- a host can belong to many different groups
  - no restriction
- a source can send to a group, no matter whether it belongs to the group or not
  - membership not required
- the group is dynamic, a host can subscribe to or leave at any time
- a host (source/receiver) does not know the number/identity of members of the group
Example: video-conferencing

The user's perspective

Multicast address group 224.2.0.1

Basics IP multicast

Receivers must be able to subscribe to groups, need group management facilities
A communication tree must be built from the source to the receivers
Branching points in the tree must keep multicast state information
Inter-domain routing must be reconsidered for multicast traffic
Need to consider non-multicast clouds

IP multicast TODO list

good luck…
incremental deployment

groups management

session advertising

tree construction

address allocation

duplication engine

forwarding state

routing

Basics IP multicast
Multicast and the TCP/IP layered model

Application

security
reliability mgmt
congestion control
other building blocks

Socket layer

TCP
UDP

ICMP
IP / IP multicast
IGMP
device drivers

user space
kernel space

higher-level services

multicast routing

Basics IP multicast
The two sides of IP multicast

- local-area multicast
  - use the potential diffusion capabilities of the physical layer (e.g. Ethernet)
  - efficient and straightforward

- wide-area multicast
  - requires to go through multicast routers, use IGMP/multicast routing/...(e.g. DVMRP, PIM-DM, PIM-SM, PIM-SSM, MSDP, MBGP, BGMP, MOSPF, etc.)
  - routing in the same administrative domain is simple and efficient
  - inter-domain routing is complex, not fully operational
IP Multicast Architecture

Service model

- Host-to-router protocol

- Multicast routing protocols

Basics IP multicast
Basic of IP multicast model

Early group management

IP multicast routing

First steps in reliability
Internet Group Management Protocol (RFC 1112)

- IGMP: “signaling” protocol to establish, maintain, remove groups on a subnet.
- Objective: keep router up-to-date with group membership of entire LAN
  - Routers need not know who all the members are, only that members exist
- Each host keeps track of which mcast groups are subscribed to

Early group mngt
IGMP: subscribe to a group (1)

224.0.0.1 reach all multicast host on the subnet

Early group mngt
early group mngt
IGMP: subscribe to a group (3)

Early group mngt
Data distribution example

Early group mngt
IGMP Join

Low bandwidth video (10–25 kb/s) with views from all over the world.

Contact Details
- Format: H.261
- Protocol: RTP
- Addr: 224.2.172.238
- Port: 51482
- TTL: 127
- Key:

Heard from 128.253.115.224 at 21 Mar 2003 15:57 CET

Frame 2520 (46 on wire, 46 captured)
- Ethernet II
- Internet Protocol
- Internet Group Management Protocol
  - Version: 1
  - Type: 6 (Host response (v2))
  - Unused: 0x00
  - Checksum: 0x5d0e
  - Group address: 224.2.172.238 (224.2.172.238)
IGMP: leave a group (1)

Sends *Leave (IGMPv2)* for 224.2.0.1 at 224.0.0.2

224.0.0.2 reach the multicast enabled router in the subnet

Early group mngt
IGMP: leave a group (2)

Host 1

224.2.0.1

Host 2

224.2.0.1

Host 3

224.5.5.5

224.5.5.5

Sends IGMP Query for 224.2.0.1

Early group mngt
IGMP: leave a group (3)

Hey, I'm still here!

Early group mngt
IGMP: leave a group (4)

Early group mngt
IGMP: leave a group (5)

Early group mngt
IGMP

IGMP Leave

Contact Details
- Format: H261
- Proto: RTP
- Addr: 224.2.172.238
- Port: 51482
- TTL: 127
- Key:

Heard from 128.253.115.224 at 21 Mar 2003 15:57 CET

Protocol Information:
- Encryption: none (NOENC)
- Authentication: none (NCAUTH)

Application Details:
- Low bandwidth video (10–25 kb/s) with views from all over the world.

Network Traffic:
- Frame 760
  - (46 on wire, 46 captured)
  - Ethernet II
  - Internet Protocol
  - Internet Group Management Protocol
    - Version: 1
    - Type: 7 (Leave group (v2))
    - Unused: 0
    - Checksum: 0x5c0e
    - Group address: 224.2.172.238 (224.2.172.238)
IGMP: leave a group (5)

Sends IGMP Query for 244.5.5.5

Early group mngt
OK, now I can express local interest, so what?

Early group mngt
Does all paths lead to Roma?
Before going further...

- Multicast on Ethernet LAN
  - How can an end-host get link-layer (MAC) packets?

- Review of Ethernet filtering
  - By default, the Ethernet device listens on
    - its (Ethernet) MAC address fixed in a PROM
    - The broadcast MAC address FF:FF:FF:FF:FF:FF
  - Other Ethernet addresses must be explicitly programmed into the driver
  - For multicast, one must listen at:
    - the Ethernet-equivalent of 224.0.0.1 (all multicast host in the LAN)
    - The Ethernet-equivalent address on which multicast sessions are advertised
Mapping of IP multicast address

- A MAC address is built from a mapping of IP multicast addr (Deering88)

![Diagram showing the mapping of IP multicast address to MAC address]

- Organizationally Unique Identifier (OUI, see RFC 1700 Assigned Number)
- Special OUI for IETF: 0x01-00-5E

LAN multicast address

Early group mngt
Basic of IP multicast model
Early group management
**IP multicast routing**
First steps in reliability
IP multicast routing

- Find a tree (dedicated, shared) between the source(s) and the receivers

- Dense Mode
  - Assumes that there are many many receivers willing to get multicast traffic
  - Uses the «push» model: everybody can receive

- Sparse Mode
  - Assumes that the number of receivers is small
  - Uses the «pull» model: requires an explicit query from the receivers.
Dense mode protocols, DVMRP

- The Ancestor: DVMRP (Distance Vector Multicast Routing)
- Based on Reverse Path Forwarding (RPF)

A multicast router forwards packets received from a link which is on the shortest path to the source, and drops other packets.
DVMRP... (cont')

- resulting multicast distribution tree

- different sources lead to diff. trees
  ⇒ improves load distribution on the links

- creates a spanning tree...

IP multicast routing
DVMRP... (cont')

- add “flood and prune” algorithm to dynamically update the tree

**Step 1: Flood the Internet (only limited by the packet’s TTL)**

**Step 2: Prune useless branches**

IP multicast routing
DVMRP... (cont')

- flooding/pruning is done periodically to update the tree
  - required to discover new receivers and remove branches to receivers who left the session

- limitations:
  - creates signaling load (PRUNE message)
  - periodically creates important traffic (flooding)
  - all routers keep some state for all the multicast groups in use in the Internet

IP multicast routing
DVMRP deployment

- Large scale deployment of DVMRP in the MBONE (multicast backbone) since 1992
- Tunnels are set up to link “multicast islands” through unicast areas

IP multicast routing
Multicast tunnelling illustrated

IP multicast routing
The early MBone with tunnels

Mixing tunnels and native multicast

### Virtual Interface Table

<table>
<thead>
<tr>
<th>Vif</th>
<th>Name</th>
<th>Local-Address</th>
<th>M</th>
<th>Thr</th>
<th>Rate</th>
<th>Flags</th>
<th>group host (time left)</th>
<th>IGMP querier</th>
<th>Nbr bitmaps:</th>
<th>pkts/bytes in:</th>
<th>pkts/bytes out:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>eth0</td>
<td>193.253.175.161 subnet: 193.253.175.128/26</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>leaf</td>
<td>239.2.11.73</td>
<td>193.253.175.135</td>
<td>0x0000000000000000</td>
<td>772010/38687700</td>
<td>/0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>239.2.11.72</td>
<td>193.253.175.142</td>
<td>0x0000000000000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>239.2.11.71</td>
<td>193.253.175.134</td>
<td>0x0000000000000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>224.0.0.4</td>
<td>193.253.175.161</td>
<td>0x0000000000000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>224.0.0.2</td>
<td>193.253.175.161</td>
<td>0x0000000000000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IGMP querier: 193.253.175.129</td>
<td>up 50:21:15 last heard 0:00:40</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Nbr bitmaps: 0x0000000000000000
- pkts/bytes in: 772010/38687700
- pkts/bytes out: /0

| 1   | eth1 | 193.253.175.249 subnet: 193.253.175.248/30 | 1 | 1  | 0    | querier leaf | 224.0.0.4          | 193.253.175.249 | 0x0000000000000000 | 0/0          |
|     |      |                |   |     |      |             | 224.0.0.2          | 193.253.175.249 | 0x0000000000000000 |               |
|     |      |                |   |     |      |             |                         | IGMP querier: 193.253.175.249 (this system) |
|     |      |                |   |     |      |             |                         | Nbr bitmaps: 0x0000000000000000 |
|     |      |                |   |     |      |             |                         | pkts/bytes in: 0/0 |
|     |      |                |   |     |      |             |                         | pkts/bytes out: 7936/10780820 |

| 2   | eth2 | 193.253.175.253 subnet: 193.253.175.252/30 | 1 | 1  | 0    | querier leaf | 224.0.0.4          | 193.253.175.253 | 0x0000000000000000 | 0/0          |
|     |      |                |   |     |      |             | 224.0.0.2          | 193.253.175.253 | 0x0000000000000000 |               |
|     |      |                |   |     |      |             |                         | IGMP querier: 193.253.175.253 (this system) |
|     |      |                |   |     |      |             |                         | Nbr bitmaps: 0x0000000000000000 |

---

**IP multicast routing**

DVMRP on Linux: the mrouted daemon
DVMRP summary

- it works but... this is far from perfect
  - periodical flooding creates a heavy load on routers/links
  - each multicast router must keep some forwarding state for each group
  - tunneling quickly became anarchic
  - this is a flat architecture (the same protocol is used everywhere)

- conclusion: “dense mode protocols” like DVMRP are not scalable enough for WAN multicast routing
  - dense mode assumes a dense distribution of receivers, wrong in practice!
DVMRP uses Source-based Trees

Source Shivkumar Kalyanaraman

IP multicast routing
Moving to a Shared Tree

Source Shivkumar Kalyanaraman

IP multicast routing
Shared vs. Source-Based Trees

- **Source-based trees**
  - Shortest path trees - low delay, better load distribution
  - More state at routers (per-source state)
  - Efficient in dense-area multicast

- **Shared trees**
  - Higher delay (bounded by factor of 2), traffic concentration
  - Choice of core/RP affects efficiency
  - Per-group state at routers
  - Efficient for sparse-area multicast
Sparse mode protocols

- The newcomers: PIM-SM/MSDP/MBGP
  - **PIM-SM**: Protocol Independent Multicast - Sparse Mode
  - **MSDP**: Multicast Source Discovery Protocol
  - **MBGP**: Multi-protocol Border Gateway Protocol

- domain ≡ site, or ISP network
  similar to “autonomous systems” of unicast routing

- intra-domain mcast routing uses PIM-SM
- inter-domain mcast routing requires MBGP
- the discovery of sources in other domains requires MSDP
PIM-SM Protocol Overview

- Basic protocol steps
  - Shared trees are unidirectional
  - Routers with local members Join toward Rendezvous Point (RP) to join shared tree
  - Routers with local sources encapsulate data in Register messages to RP
  - Routers with local members may initiate data-driven switch to source-specific shortest path trees
- PIM v.2 Specification (RFC 2362)
PIM-SM: Build Shared Tree

- Shared tree after R1,R2 join
- Join message toward RP

Source 1

R1

(*,G)

R2

(*,G)

Source Shivkumar Kalyanaraman

RP

(*,G)

Receiver 1

Receiver 2

Receiver 3

IP multicast routing
Source router unicasts encapsulated data packet to RP in Register.

RP de-capsulates, forwards down shared tree.
RP Send Join to High Rate Source

Source 1
(S1,G)

Source Shivkumar Kalyanaraman

IP multicast routing
Build Source-Specific Distribution Tree

Build source-specific tree for high data rate source

IP multicast routing

Source Shivkumar Kalyanaraman
PIM-SM... (cont')

- Moving to a per-source tree is efficient for bulk data transfer, but has a higher cost in case of multiple sources.

- One tree per source versus a single shared tree.
PIM-SM on Internet routers

- PIM-SM is implemented on all major Internet routers (CISCO, JUNIPER, Alcatel AVICI, PROCKET...)
- A linux package exists, see http://netweb.usc.edu/pim/ (I haven't tried it yet)
Example: PIM-SM on VTHD

IP multicast routing

Source doc VTHD
Configuration on CISCO routers

- Enabling PIM
  ```
  ip multicast-routing distributed
  !
  interface XX/XX
  ip pim sparse-dense-mode
  !
  ```
  For each interface

- Declaring the RP
  ```
  ip pim rp-address w.x.y.z
  ```
  IP addr of the RP

IP multicast routing
Basic of IP multicast model
Early group management
Early IP multicast routing
First step in reliability
The Wild Wild Web

UDP data

heterogeneity, link failures, congested routers, packet loss, packet drop, bit errors...

Reliability
Reliability Models

- Reliability => requires redundancy to recover from uncertain loss or other failure modes.

- Two types of redundancy:
  - **Spatial redundancy**: independent backup copies
    - Forward error correction (FEC) codes
    - Problem: requires huge overhead, since the FEC is also part of the packet(s) it cannot recover from erasure of all packets
  - **Temporal redundancy**: retransmit if packets lost/error
    - Lazy: trades off response time for reliability
    - Design of status reports and retransmission optimization important
Temporal Redundancy Model

- Packets
  - Sequence Numbers
  - CRC or Checksum
- Status Reports
  - ACKs
  - NAKs, SACKs
  - Bitmaps
- Retransmissions
  - Packets
  - FEC information

Timeout

Reliability
End-to-end reliability models

- **Sender-reliable**
  - Sender detects packet losses by gap in ACK sequence
  - **Easy resource management**

- **Receiver-reliable**
  - Receiver detects the packet losses and send NACK towards the source
Challenge: scalability (1)

- many problems arise with 10,000 receivers...

- Problem 1: scalable control traffic
  - ACK every 2 packets (à la TCP)... oops, 10000 ACKs / 2 pkt!
  - NAK (negative ack) only if failure... oops, if pkt is lost close to the source, 10000 NAKs!

source implosion!
Challenge: scalability (2)

- problem 2: scalable repairs/exposure
  - receivers may receive several times the same packet
A piece of the solutions (1)

- solutions to problem 1: scalable control traffic
  - solution 1: feedback suppression at the receivers
    - each node picks a random backoff timer
    - send the NAK at timeout if loss not corrected
  - solution 2: proactive FEC (forward error correction)
    - send data plus additional FEC packets
    - any FEC packet can replace any lost data packet
  - solution 3: use a tree of intelligent routers/servers
    - use a tree for ACK aggregation and/or NAK suppression
    - PGM, ARM, DyRAM

Reliability
A piece of the solutions (2)

- **solutions to problem 2: scalable repairs**
  - **solution 1: use TTL-scoped retransmissions**
    - repair packets have limited scope
  - **solution 2: use proactive/reactive FEC**
    - proactive: always send data + FEC
    - reactive: in case of retransmission, send FEC
  - **solution 3: use a tree of retransmission servers**
    - a receiver can be a retransmission server if he has the requested data
Scalable Reliable Multicast

Floyd et al., 1995

- Receiver-reliable, NACK-based
- NACK local suppression
  - Delay before sending
  - Based on RTT estimation
  - Deterministic + Stochastic
- Every member may multicast NACK or retransmission
- Periodic session messages
  - Sequence number: detection of loss
  - Estimation of distance matrix among members

Reliability
SRM Request Suppression

from Haobo Yu, Christos Papadopoulos
SRM Request Suppression

from Haobo Yu, Christos Papadopoulos
SRM Request Suppression

from Haobo Yu, Christos Papadopoulos
SRM Request Suppression

each node picks a random backoff timer

from Haobo Yu, Christos Papadopoulos
SRM Request Suppression

Each node picks a random backoff timer.

From Haobo Yu, Christos Papadopoulos

Reliability
SRM Request Suppression

Each node picks a random backoff timer.
SRM Request Suppression

from Haobo Yu, Christos Papadopoulos

Reliability
SRM Request Suppression

from Haobo Yu, Christos Papadopoulos

Reliability
SRM Request Suppression

from Haobo Yu, Christos Papadopoulos

Reliability
SRM Request Suppression

from Haobo Yu, Christos Papadopoulos
Deterministic Suppression

\[ \text{Delay} = C_1 \times d_{S,R} \]

Distance = \((T_4 - T_3 + T_2 - T_1) / 2\)

Reliability

from Haobo Yu, Christos Papadopoulos
Simple TTL-scoped of repairs

- use the TTL field of IP packets to limit the scope of the repair packet
Summary: reliability problems

- What is the problem of loss recovery?
  - feedback (ACK or NACK) implosion
    - ACK/NACK aggregation based on timers are approximative!
  - replies/repairs duplications
    - TTL-scoped retransmissions are approximative!
  - Heterogeneity of receivers (crying baby, congestion control)
  - difficult adaptability to dynamic membership changes

- Design goals
  - reduce the feedback traffic
  - reduce recovery latencies
  - improve recovery isolation
**FEC (Forward Error Correction)**

- Add some redundancy to the data flow
- A single FEC packet can recover different losses at different receivers \(\Rightarrow\) improves scalability
- We only consider packet-based erasure channels (like the Internet)
  - packets are either perfectly received or lost
  - mimics the effects of congested routers
  - FEC operates on a packet basis

Adv. reliability  FEC-based
MDS property

- **Maximum Distance Separable (MDS) FEC code**
  - **sender:** FEC (k, n)
    - for k original data symbols, add n-k FEC symbols
    - ⇒ total of n symbols (or packets) sent
  - **receiver:**
    - as soon as it receives any k symbols out of n, a receiver can reconstruct the original k symbols
    - a FEC code with this property is called “MDS”
FEC classification

- Classification based on the \((k, n)\) parameters
  - **small block FEC codes** (small \(k\))
    - Reed-Solomon (based on Vandermonde matrices, or Cauchy matrices), Reed-Muller...
  - **large block FEC codes** (large \(k\))
    - LDPC, Tornado
    - belong to the “codes on graph” category
  - **expandable FEC codes** (large \(k\) and \(n\))
    - LT

Adv. reliability | FEC-based
FEC classification... (cont')

- other codes exist but are
  - either lossy codes (ok for video/audio transmission)
  - or dedicated to bit stream transmissions over noisy channels
  - not for us!
Small block FEC codes

- e.g. Reed-Solomon codes [Rizzo97]
- this is an “MDS code”
  - any $k$ out of $n$ is sufficient to build original pkts
- the $k$ parameter is < a few tens for computational reasons
  - split large data objects into several blocks
  - limits correction capability of a FEC symbol
  - limits the global efficiency

Adv. reliability  FEC-based
Small block FEC codes… (cont’)

- an example of problem generated by a small $k$

- limited number of $n$ x FEC symbols created
  $\Rightarrow$ can lead to packet duplications

- high quality open-source implementation available
Large block FEC codes

- e.g. LDPC and Tornado codes
- (k,n) with a very large k
- but n is limited in practice (e.g. n = 2k)
- decoding requires \((1+\varepsilon)k\), i.e. a bit more than k symbols
  - \(\varepsilon\) is around %10 (for the best codes) to 40%
- this is not an MDS code
- high-speed encoding/decoding
Large block FEC codes… (cont')

- **an example: LDPC code**
  - based on XOR operations ($\oplus$)
  - uses bipartite graphs between source and FEC symbols
  - iterative decoding

$k$ data symbols  \quad  (n-k)$ FEC symbols

\begin{align*}
x1 & \oplus c1 = x1 \oplus x3 \oplus x4 \\
x2 & \oplus c2 = x1 \oplus x2 \oplus x5 \\
x3 & \oplus c3 = x3 \oplus x4 \oplus x6 \\
x4 & \oplus c4 = x2 \oplus x3 \oplus x5 \oplus x6 \\
x5 & \oplus c5 = x5 \oplus x6 \\
x6 &
\end{align*}

A receiver that knows $x3$, $x4$ and $c1$ can recover $x1$:

$$x1 = c1 + x3 + x4$$

Adv. reliability  \quad FEC-based
Multicast for video distribution

- Video contents are mostly distributed with unreliable mechanisms
- Main concern is scalability to a large number of users
- Need to worry about congestion control and heterogeneity
- Most approaches use layered approaches

Adv. reliability  Layered
Principles of multi-layering

- 1 multicast group is assigned to 1 layer
- Throughput on each layer could be identical or increasing
- Subscription to a layer means subscription to a new group

Segmentation in packets
Generation of redundancy packets
Layers construction

Layer 0
Layer 1

Adv. reliability Layered
Example of layer operations

- Assuming that
  - Throughput in each layer is the same
  - There are a maximum of 4 layers
Synchronizing joins and leaves

- Layered approaches rely on fast joins and leaves from receivers
- More efficient if joins/leaves operations are synchronized

Adv. reliability  Layered
Status and deployment of multicast technologies
The open model

no-security

**CONTRACT**

*Can not* control sources

*Can not* control receivers

*Can not* control groups

*Can not* control traffic

Please sign

Status?
Relative Size of the Multicast Enabled Internet

The Percentage of the Internet Supporting Multicast

- Multicast Prefixes / Total Prefixes
- Multicast AS / Total AS
- Multicast Addresses / Total Addresses

Source: www.multicasttech.com/status
CISCO IP/TV,

- Usages
  - Training, Business, Corporate Communications, Distance Learning, Videoconferencing...

Status?
XtremeCast from mPulse

- **Usage**
  - Used by financial firms for stock quotes broadcasting
  - Chat server
- Reliable multicast implementation with the JRMS library (©SUN)
- [http://www.mpulsetech.com/prod/xcast.htm](http://www.mpulsetech.com/prod/xcast.htm)
Digital Fountain products

- Implement ALC/LCT/WEBRC and rely on two highly efficient large block FEC codecs
  - http://www.digitalfountain.com
  - high implication in the IETF RMT standardization process

Status?
Multicast Monitor

monitor multicast traffic in the enterprise network