Overlay Multicast/Broadcast

- Broadcast/Multicast Introduction
- Application Layer Multicast
- Unstructured Overlays
  - Centralised
  - Distributed
- Structured Overlays
  - Flooding: CAN & Prefix Flood.
- Tree-based:
  - Scribe/ SplitStream/ PeerCast Bayeux
- Additional Design Mechanisms
We need Multicast/Broadcast Services for …

- Public Content Broadcasting
- Voice and Video Conferencing
- Collaborative Environments
- Gaming
- Rendezvous Processes / Neighbour Discovery
- Self Organisation of Distributed Systems
- …

All of this seamless and ubiquitous!
Broadcast

- Special mode of group communication: all nodes
- Operates without active participation of nodes
  - No signalling involved
  - Simple to map to lower layers (shared media)
  - Potential of increased efficiency
  - Well suited for rendezvous processes
- Results in flooding - typically bound to limited domains (locality)
IP Multicasting

Service for Transfering IP Datagrams to Host-Groups

- Originally: RFC 1112 (S. Deering u.a., 1989)
- Addresses a host-group by means of one group address
- Two types of Multicast:
  - Any Source Multicast (ASM)
  - Source Specific Multicast (SSM)
- Client protocol for group membership management (IGMP/MLD)
- Internet core left with complex Multicast Routing
IP Mcast Deployment Issues

- Complexity versus Performance Efficiency
  - IP Multicast most efficient, but burdens infrastructure
- Provider Costs
  - Provisioning of knowledge, router capabilities & maintenance, Interdomain multicast routing problem
  - Business model: Multicast saves bandwidth, but providers sell it
- Security
  - ASM assists traffic amplification for DDoS-attacks
- End-to-End Design Violation?
  - Service complexity objects implementation at lower layer
  - But for efficiency: Multicast needs lowest possible layer
Multicast: Alternative Approaches

- Application Layer Multicast (ALM)
  - Solely built with end-user systems
  - Free of any infrastructure support

- Overlay Multicast
  - Built on fixed nodes / proxies
  - Nodes connect to local proxies
  - Proxies responsible for routing
Application Layer Multicast

Advantages:
- Easy to deploy

Disadvantages:
- High control overhead
- Low efficiency
- Degradation by end system instability
Overlay Multicast

Advantages:
- Improved efficiency in tree management
- Enhanced scalability
- Reduced control overhead

Disadvantages:
- Deployment complexity
Unstructured ALM: ALMI

- Relies on Session Controller
  - Dedicated server or group member node
  - Computes minimal spanning distribution tree
  - Assigns tree neighbours
- Controller unicast messages per member
- Scalability issues
ALMI Self Organisation

Node Arrival:
- New node sends \textit{JOIN} to controller, in response receives its ALM ID + parent location
- New node submits \textit{GRAFT} to initiate parent forwarding

Node Departure:
- Departing node sends \textit{LEAVE} to controller, which then updates tree neighbours

Overlay Maintenance:
- Group members probe on others and report to controller (up to $O(n^2)$ messages)
Unstructured, distributed: End System Multicast/ Narada (Chu et al. 2000)

- Group management equally distributed to all nodes
  - Each overlay node keeps track of all group members
  - Periodic heartbeat broadcasts of all members
- Construct overlay tree from a mesh
  - Overlay nodes first organize in a redundantly meshed graph
  - Source specific shortest path trees then constructed from reverse paths
- Regulates node fan-out degree to balance load
Narada Components

- **Mesh Management:**
  - Ensures mesh remains connected in face of membership changes

- **Mesh Optimization:**
  - Distributed heuristics for ensuring shortest path delay between members along the mesh is small

- **Spanning tree construction:**
  - Routing algorithms for constructing data-delivery trees
  - Distance vector routing, and reverse path forwarding
  - Analogue DVMRP
Optimizing Mesh Quality

Members periodically probe other members at random

New Link added if
Utility Gain of adding link > Add Threshold

Members periodically monitor existing links

Existing Link dropped if
Cost of dropping link < Drop Threshold

A poor overlay topology
Desirable properties of heuristics

- **Stability**: A dropped link will not be immediately readded
- **Partition Avoidance**: A partition of the mesh is unlikely to be caused as a result of any single link being dropped

Delay improves to Stan1, CMU but marginally.
Do not add link!

Delay improves to CMU, Gatech1 and significantly.
Add link!
Used by Berk1 to reach only Gatech2 and vice versa.

Drop!!

An improved mesh!!
Evaluation: Relative Delay Penalty

128 Group Members within 1024 Nodes with 3145 Links
Structured Overlay Multicast

- Flooding-based approaches
  - Packet broadcasts within a structured overlay
  - Selective broadcast (multicast) by group-specific DHT
    - Multicast on CAN & Prefix Flooding
- Tree-based approaches
  - Shared trees: Routing via group-specific rendezvous point
    - Scribe/Splitstream
  - Source-specific trees: Construction of source-specific shortest path trees after source announcements
    - Bayeux
Multicast on CAN (Ratnasamy et al 2001)

- Within a previously established CAN overlay members of a Group form a “mini” CAN
  - Group-ID is hashed into the original CAN
  - Owner of the Group key used as bootstrap node

- Multicasting is achieved by flooding messages over this mini CAN

- Number of multicast states is limited by $2d$ neighbours
  - independent of multicast source number!

- Can Multicast scales well up to very large group sizes
  - Replication load limited to neighbours ($2d$)
  - But tends to generate packet duplicates
Improved Flooding

- Source of a message forwards it to all neighbours.
- Receiver of a message (from dimension $i$) only forwards along dimensions lower than $i$ and along $i$ in opposite direction.
- A node does not forward to a dimension where the message has already travelled half way from source coordinate.
- Nodes cache sequence numbers already forwarded to prevent duplicate forwarding.
Can Forwarding
Evaluation: Relative Delay Penalty
Prefix Flooding

- DHT Nodes are identified by hash codes
- Idea:
  - Arrange IDs in a prefix tree
  - Flood prefix neighbours (w.r.t. longest common prefix (LCP))
- Defines broadcast for any DHT, Multicast per mini-DHT analogue to CAN
- Packet delivery unique: no duplicates
- Particularly well suited for proximity-aware prefix routing like in Pastry
Prefix Flooding Algorithm

Routing requires:

- Destination prefix $C$ for on-tree context
- Proactive routing maintenance: prefix neighbour entries needed for forwarding

**Prefix Flooding**

▷ On arrival of a packet with destination prefix $C$
▷ at a DHT node

1. **for** all $N_i$ IDs in prefix neighbor set
2. **do if** ($LCP(C, N_i) = C$) ▷ $N_i$ downtree neighbor
3. **then** $C_{new} \leftarrow N_i$
4. **FORWARD** PACKET TO $C_{new}$
Analysis of Prefix Flooding

- Structural analysis relatively simple due to the recursive nature of k-ary trees
- Distinguish between fully and sparsely populated tree

Closed expressions for:
- Replication Load
- Hop Count
Performance Values

$p$ is sparseness parameter
Résumé on Flooding Approaches

- Defines a natural broadcast mechanism on the KBR
  - Transparent for sources & receivers:
    no signalling, no additional states
  - Problem duplicates & efficiency,
    solved with Prefix Flooding over Pastry
- Multicast requires construction of sub-DHTs
  - Tedious & slow – high overheads
  - Group management based on DHT membership management
Shared Distribution Tree: Scribe (Castro et al 2002)

- Large-scale distribution service based on Pastry
- Rendezvous Point chosen from Pastry nodes
  - Choice according to group key ownership
  - RP roots shared distribution tree (analogue PIM-SM)
- Shared tree created according to reverse path forwarding
  - Nodes hold *children tables* for forwarding
  - New receiver routes a *SUBSCRIBE* towards the RP
  - *Subscribe* intercepted by intermediate nodes to update children table, reverse forwarding done, if node not already in tree
Scribe API

- **Create (credentials, topic ID):** Creates a group identified by a unique topic ID (hash of textual description + creator ID), credentials administrative

- **Subscribe (credentials, topic ID, eventHandler):** Initiates a local join to group, asynchronously received data passed to the eventHandler

- **Unsubscribe (credentials, topic ID):** Causes a local leave of group

- **Publish (credentials, topic ID, event):** Multicast source call for submitting data (event) to group
Scribe Tree Construction
Can versus Scribe: Delay Penalty

(a) CAN

RMD: Relative Delay Maximum
RAD: Relative Average Delay

(b) Scribe

CAN may be configured to provide higher network efficiency
Scribe Performance: Hop Count

- Simulation in OverSim network simulator
- 1,000 Pastry nodes
- Hop Count evaluated for varying group sizes
Tree Characteristics in Scribe

- Almost all branches arise from Rendezvous Point
- Scribe foresees „manual“ load balancing
Improvement: SplitStream (Castro et al. 2003)

- Focus on media data distribution
- Idea: Split media streams into slices and distribute sliced streams via disjoint trees
- Disjoint trees created by modifying prefix initial
  - Pastry leads to disjoint prefix routes
  - Scribe distribution trees according to prefix routes
  - All group members are leaves in all trees
- Accounts for member bandwidth constraints
- Problem: Jitter explosion
PeerCast (Zhang et al. 2004)

- Multicast distribution service enhancing SCRIBE
  - Variation of PASTRY
  - Rendezvous-Point-based shared distribution tree
- Overlay structure adaptive to node capacities
- Landmark signatures to map proximity into key space
- Dynamic, passive replication scheme for reliable multicast distribution
- Two-tier approach:  - ES Multicast Management
  - P2P Network Management
PeerCast: P2P Management

- Proximity-aware DHT using landmarking
  - Landmark signature generated from distances to fixed landmark nodes
  - Landmark signature then substitutes a substring of each key identifier at the same “Splice Offset”
  - Neighbouring peers then clustered into “buckets”

- Accounting for node capabilities
  - Each node generates a multitude of keys, thus encountering multiple presence in the DHT ring
  - Key quantities are chosen according to node capabilities
PeerCast: ES Multicast Management

- Rendezvous Node chosen as group key owner
- Shared tree created according to reverse path forwarding
- Improvement – Neighbour Lookup:
  - Subscribers + forwarders check their neighbours prior to forwarding subscription request
  - If any neighbour has already joined the group, a ‘shortcut’ is taken
Performance of PeerCast

Figure 17: Relative delay penalty, $r = 8$
peers number = 50,000

Figure 18: Relative delay penalty, $r = 8$
peers number = multicast group size

$r$ is heterogeneity measure
Source Specific Distribution Tree: Bayeux (Zhuang et al, 2001)

- Based on Tapestry
- Creates a group by placing an empty file named by the hashed group ID
  - Announced by Tapestry location service
- Receivers learn about group ID and perform source-specific subscriptions
- Subscriptions are routed to the owner of the file, acting as the source & central controller
- Source (and intermediate branch nodes) perform full receiver tracking
Bayeux Group Management

- Distribution tree is built according to (forward) pushed TREE messages

- Leaves are routed to the source and trigger a PRUNE message
Bayeux Performance

- Bayeux suffers from scaling problems due to the central controller.
- Improvements are proposed to cluster receivers (hybrid) and to replicate via several roots.

<Diagram showing cumulative percentage of pairs of members vs. RDP.>
Research Issues

- Joined / combined / hybrid solutions for a global group communication layer
- Redundancy & robustness enhancements by Network Coding
- Multipath transport without jitter explosion
- Proximity under mobility – Constructions of distributions trees efficient w.r.t. the underlay topology
- Stability under mobility – Construction of efficient multicast distribution trees, which are robust
- QoS improvements & flow control, measures and guaranties to provide real-time capabilities
- Security & Robustness against malicious node behaviour
Additional Design Mechanisms

Two core problems arise in wide-area broadcast/multicast distribution:

- Reliability and redundancy without retransmission
  - In particular for file distribution: all blocks are needed
  - Promising approach: Network Coding
- Flow control / flow adaptation in heterogeneous environments
  - Data streams may meet network bottlenecks
  - Promising approach: Selective dropping after Backpressure Control
Network Coding (Li, Yeung, Cai, 2003)

- Original idea: network efficiency can be enhanced by linear combination of packets
- Useful in Wireless transmission to enhance efficiency
- In Overlay Multicast mainly to add 'universal' redundancy

Max. broadcast rate = 1.5  Max. broadcast rate = 2
Network Coding Simplified

Encoding

Block 1 \hspace{1cm} Block 2 \hspace{1cm} Block 3

File to Transfer
With Network Coding
Problem of Flow Control

- In a distribution system (e.g., Tree) there may occur at some part
  - Heterogeneous link transitions
  - Congestions
  - Fluctuating link conditions

- Problems
  - Long-range (e.g., receiver) feedback prevents scaling
  - How to decide locally on efficient flow forwarding (omit forwarding packets that are discarded later)?
Group Distribution without Flow Control
Backpressure Multicast: Simple Flow Control

- Intermediate Node can decide about dropping or delaying
References