Optimizing Xcast Treemap Performance with NFV and SDN

T. Khoa Phan

*Joined work with* David Griffin and Miguel Rio

University College London

Next Generation Networking workshop, July 2016
98% of user requests can be served immediately by edge caches
Each edge cache can serve up to 200,000 users simultaneously

What is Xcast Treemap?

Breath-first tree traversal

A  B  C  D  E  → List of IP addresses
2  0  2  0  0  → Treemap

Sender S creates packets:

<table>
<thead>
<tr>
<th>Src: S, Dest: A</th>
<th>A  B  C  D  E</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2  0  2  0  0</td>
<td></td>
</tr>
</tbody>
</table>

Unicast part ↔ Xcast treemap part (optional)

Today router only understands unicast part.

Xcast router lookups and forwards for each IP in the list.

Xcast end-host and Xcast router software are available (in IPv6):
http://www.ee.ucl.ac.uk/~uceetkp/Xcast_software.zip
How Xcast Treemap works?

Full Xcast packet header created by S:

<table>
<thead>
<tr>
<th>Src: S, Dest: A</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 0 2 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Dest in unicast part

List dests in X6T part

Unicast routing table at R1

<table>
<thead>
<tr>
<th>Dest</th>
<th>Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B, C, D, E</td>
<td>R2</td>
</tr>
</tbody>
</table>

Unicast routing table at R3

<table>
<thead>
<tr>
<th>Dest</th>
<th>Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>E</td>
<td>E</td>
</tr>
</tbody>
</table>

Fig. 1: only today routers
How Xcast Treemap works?

Full Xcast packet header created by S:

<table>
<thead>
<tr>
<th>Src: S, Dest: A</th>
<th>A B C D E</th>
<th>2 0 2 0 0</th>
</tr>
</thead>
</table>

Dest: S, Next hop: A
Payload:

Fig. 2: mixture of Xcast and today routers

Unicast routing table at R1:

<table>
<thead>
<tr>
<th>Dest</th>
<th>Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B, C, D, E</td>
<td>R2</td>
</tr>
</tbody>
</table>

Unicast routing table at R3:

<table>
<thead>
<tr>
<th>Dest</th>
<th>Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>E</td>
<td>E</td>
</tr>
</tbody>
</table>

Xcast router
Today router
Facebook Livestream System

![Diagram of Facebook livestream system]

Fig. 4: Facebook livestream system - CDN [1]

- 98% of user requests can be served immediately by edge caches
- Each edge cache can serve up to 200,000 users simultaneously

Facebook Livestream System

Fig. 5: Xcast with Facebook live stream

- The edge cache uses only 1 slot instead of 5 slots
- In this example, each edge cache can serve up to 1,000,000 users
Facebook Livestream System

Xcast Treemap is a stateless protocol. The overlay tree can be changed by the edge cache in silence!!!
Xcast Treemap with NFV

• Integrate Xcast router service in lightweight containers (Docker and Kubernetes) (on-going work)

• Users can deploy their own Xcast gateways, no touch the core network

• Smart placement (on demand) for Xcast routers (on-going work)
Xcast Treemap with SDN

• Use SDN to separate and control Xcast traffic from other traffic (on-going work)

• Traffic engineering for Xcast traffic (on-going work)

• SDN controller helps to build a good overlay tree (on-going work)
Xcast Treemap with SDN

Shortest path routing

Traffic engineering for Xcast traffic

SDN controller
Conclusions

• **Xcast Treemap vs. P2P:**
  - Quick adaptation when end-hosts join/leave.
  - Significantly reduce latency and save bandwidth if we can deploy Xcast routers/gateways.

• **Advantages of Xcast Treemap vs. IP multicast:**
  - Xcast cost is as low as unicast (no need routers upgraded, no multicast protocol, resilience with network failure, stateless protocol).
  - No problem with inter-domain multicast as Xcast works like unicast and P2P.

• **On-going work:**
  - NFV to ease Xcast router deployment
  - SDN to manage and to optimize traffic engineering for Xcast traffic

Xcast end-host and Xcast router software are available (in IPv6):
http://www.ee.ucl.ac.uk/~uceetkp/Xcast_software.zip
Backup slides
A bit of Xcast history


  - Internet community preferred IP multicast over Xcast concept.

  - No overlay tree, only work for small multicast group
  - Need Xcast router

• Our work: Xcast6 Treemap
  - Support overlay multicast and work for large multicast group
  - Xcast routers are optional
How Xcast work with large group?

Apply similar approach as how we scale the Facebook edge cache server if the sender has very high capacity, otherwise partition the overlay tree and send data as follows:

By selecting some end-hosts (A and C in this Figure) to behave like the original sender. By this way, each packet header contains only 5 IP addresses (but can support thousands of users like P2P).
How Xcast Treeamap works?

B has no child in the overlay tree but B sees [C D E] when receiving packets. It means that the overlay tree is broken by Xcast routers located somewhere on the network. Therefore B will forward the packets to C (the first in the list [C D E]).
How Xcast works?

Today router
Xcast router
How Xcast works?

Full Xcast packet header created by S:

<table>
<thead>
<tr>
<th>Src: S, Dest: A</th>
<th>A B C D E</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 0 2 0 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Dest in unicast part

List dests in Xcast part

<table>
<thead>
<tr>
<th>Dest</th>
<th>Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B, C, D, E</td>
<td>R2</td>
</tr>
</tbody>
</table>

Unicast routing table at R1

Fig. 3: all routers are Xcast routers
**Facebook Livestream System**

New viewers join: the edge caches will serve them directly as how the system is working now, and will move them to an overlay tree when they are in the session long enough (stable viewers).

Edge caches can change the pkt header in silence!!!

Nothing happens to D and E when C leaves the session because Xcast router X3 forwards data to them.