An Efficient and Scalable Engine for Large Scale Multimedia Overlay Networks

Pier Luca Montessoro
Department of Electric, Managerial and Mechanical Engineering (DIEGM)
University of Udine
Italy

Stefan Wieser
Laszlo Böszörmenyi

Institute of Information Technology (ITEC)
Klagenfurt University
Austria
Motivations

- Increasing demand of multimedia streaming and remote storage
- No control on the network infrastructure and limited cooperation from ISPs
- To make overlays a feasible solution we must provide:
  - Scalability
  - Flexibility through dynamic self-organization
  - Performance (fast packet forwarding)
Architecture

overlay network layer

sender

A

B

C

D

E

receiver

host A

node B

node D

host F

Resource reservation and keepalive messages

collected indexes
reservation info

host A

node B

DSDL table

forw. table

DSDL table

forw. table

physical links

ON virtual links

flow route
• REBOOK
  REBOOK/DLDS
  • deterministic, dynamic, per-flow resource reservation
  • It IS NOT another reservation protocol
  • It IS a distributed algorithm for efficient status information handling within intermediate nodes
• DLDS (Distributed Linked Data Structure)
  • the enabling algorithm

REBOOK: a deterministic, robust and scalable resource booking algorithm. (JONS 2010)

A novel algorithm for dynamic admission control of elastic flows. (FITCE 2011)

Distributed Linked Data Structures for Efficient Access to Information within Routers. (ICUMT 2010)

DLDS (Distributed Linked Data Structure)

During setup: “pointers” collection

- store resource reservation information in routers AND
- keep track of pointers (memory addresses or indexes in tables) along the path

Afterwards

- use the pointers to access status information without searching

→ constant cost to access the per flow resource reservation info
→ virtual circuit performance for packet forwarding
Resource reservation and pointers collection

NOTE: in this context, ON nodes acting as software router
Fast packet forwarding

sender

R1

R2

R3

R4

receiver

Data Packet

Reservation Table

Resource Reservation Table

Forwarding Table

Destination Output Port

Reservation Info Local Index Next Index

Data Packet

Reservation Info Local Index Next Index

Forwarding Table

Resource Reservation Table

Reservation Info Local Index Next Index

Forwarding Table

Resource Reservation Table

Reservation Info Local Index Next Index
A Few Problems

- Route changes, disappearing flows, end nodes or routers faults
  - High speed consistency check
  - Low priority table cleanup process

- Need to dynamically change assigned resource amounts
  - Partial release
  - Distributed control function for optimality and fairness
Does it work?

10 UDP flows, Rmin=15 Rreq=25

this link is down between T₁ and T₂

number of booked flows per sender node

total packet rate per sender

T₁: route change

T₂: route change
Does it work?
(cont’d)
Flocks: Flexible and Self-Organizing Overlay

• Gossip-based protocol
  • Uses “Interest” to build the overlay
  • Idea: Interested nodes “stick” together
  • Interest in neighbours computed locally

• QoS estimation and QoS-aware routing
  • QoS-estimate to any node, fast and scalable to large networks

• Topology aggregation to generate a hierarchical representation for scalability
  • Nodes only need a small local view

Decentralized topology aggregation for QoS estimation in large overlay networks. (NCA2011)

Flocks: Interest-based construction of overlay networks. (MMEDIA10)
Combined Overlay

• Why combine Flocks with REBOOK?
  • Flocks provide a flexible and scalable overlay
  • REBOOK allows to keep track of bandwidth allocation and avoid overcommitting links
  • DLDS enable fast routing table access and higher performance once routes are established (O(1))
Evaluation: Overlay Setup

- Overlay with 160 Flock-REBOOK nodes
  - Nodes only have a small local view
  - Two random groups connected by bottleneck
  - Within each group: Interest in high bandwidth
- Small local view of only 4 other nodes
- Artificial bottleneck crossed by all flows
  - Alternative paths outside local view
  - Must refer to aggregated information
  - REBOOK tracks allocated resources
- New reservation
  - Use path with the highest estimated bandwidth
Experimental Results: Admitted Flows

- Admitted flows increased significantly
  - Overlay avoids overcommitting links
  - Discovers alternative path w. QoS estimation
Experimental Results: Admitted Flows

- Bandwidth per flow increases as well
  - Overlay uses alternative paths used before
  - REBOOK needs to reduce resources

![Graph showing admitted flows with and without REBOOK and QoS estimation](image-url)
Experimental results (cont’d)

Resource reservation in a small congested network
# Performance

<table>
<thead>
<tr>
<th>Activity</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource reservation setup</td>
<td>200 ns per flow</td>
</tr>
<tr>
<td>Keepalive message handling</td>
<td>100 ns per flow</td>
</tr>
<tr>
<td>Resource reservation release</td>
<td>25 ns per flow</td>
</tr>
<tr>
<td>Forwarding table access</td>
<td>10.6 ns per packet</td>
</tr>
</tbody>
</table>
Conclusion

• Flocks provide a flexible and scalable overlay for multimedia delivery

• Hierarchical aggregation and QoS estimation of Flocks allow discovering alternative paths using small, local views

• REBOOK allows a scalable way for overlays to keep track of aware of resource utilization

• REBOOK and DLDS enable efficient and fast performing routing for overlays
Thank you!

www.montessoro.it
Flocks: Interests

• Each node has two sets of name-value pairs
  • May differ from node to node, and over time
  • Used to rank neighbours
• Property-Set ("what I have")
  • Shared with neighbours: virtual, uncertain, inheritable
• Interest-Set ("what I want")
  • Not shared, evaluated locally
FLOCKS PERFORMANCE: TRAFFIC

- Average incoming traffic per node (kbit/s)

![Traffic Chart]

- Stabilization
- Failure
- Churn
- Recovery

Traffic in kbit/s

Simulation Time in Seconds

Kilobit per second sent, per node
**Performance: QoS**

- Measures closeness to the global optimum
Thank you!

www.montessoro.it