Exposing the network: support for topology-sensitive applications

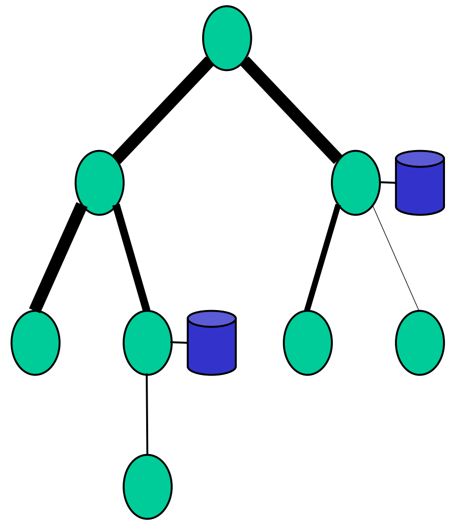
Y. Chae   S. Merugu   E. Zegura
Networking and Telecomm Group
College of Computing
Georgia Tech

S. Bhattacharjee
Dept. of Computer Science
University of Maryland

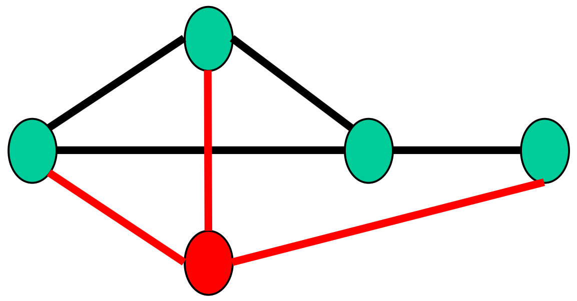
OpenArch 2000, Tel Aviv, Israel

Motivation

- Repair server placement
  - determine locations for mcast repair servers



- Secure overlay creation
  - identify secure links and nodes



Key: requires knowledge of topology

Problem statement

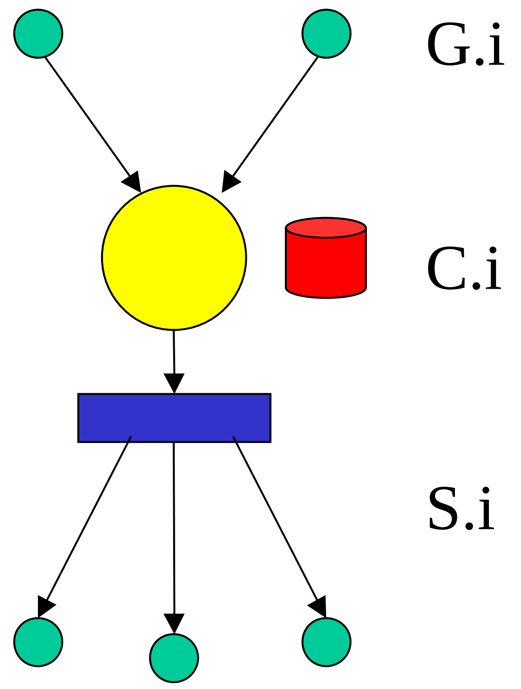
- Define service that allows:
  - constrained query and synthesis of network topology information
- Assume:
  - nodes maintain attributes for local information (including link info)
- Restrict:
  - computation functionality
  - access to node state

Outline

- Problem statement
- Solution overview
- Implementation
- Performance results
- Conclusions

Iterative GCS overview

- Gather
  - collect messages on specified channels
- Compute
  - perform computation using gathered messages, stored state, node/link attributes
- Scatter
  - send message on specified channels

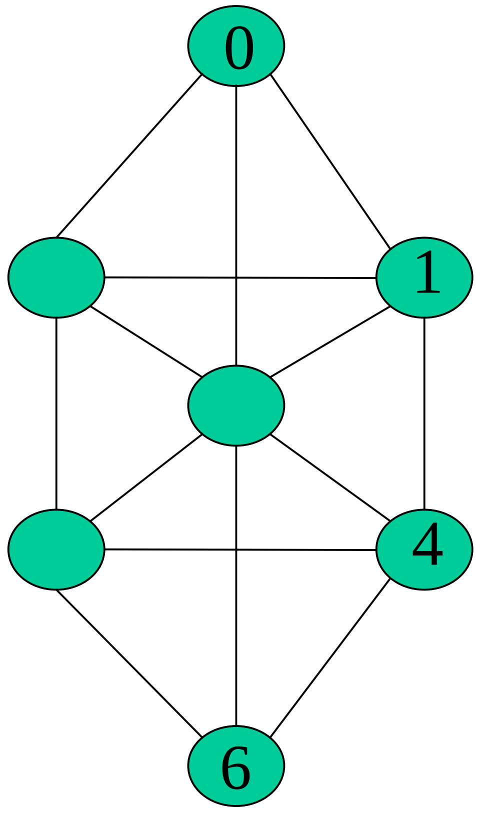


IGCS Specification

- Set of three-tuples: {(G.i, C.i, S.i)}, where tuple i describes i-th iteration
- G.i, S.i: sets of link descriptors
- C.i: computation function
  - may change local state Π (including subsequent gather and scatter sets) and create output msg
  - oMsg $\leftarrow$ C({iMsg},Σ.node,Σ.link,Π)

Example: path info retrieval (I)

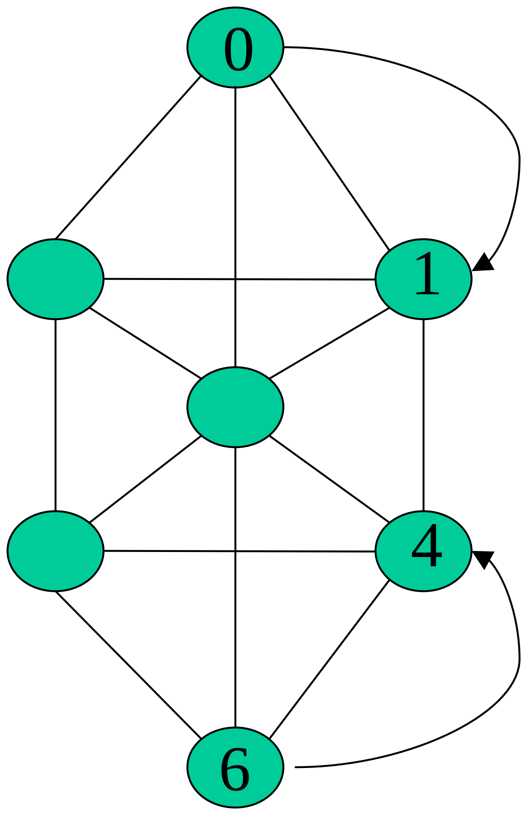
- Iteration 0 (set up)
  - messages flow from src to dst
  - compute downstream path and establish sets S.0, G.1, S.1
- Iteration 1 (work)
  - messages flow from dst to src
  - compute using message content and local state (e.g., min)



src=0, dst=6

Example: path info retrieval (II)

- Iteration 0 details:
  - copy src msg to out msg
  - record incoming link as S.1
  - lookup next hop to dst (h)
  - record h as G.1and S.0
  - scatter out msg to S.0
- Iteration 1 details:
  - wait for message on G.1
  - copy src msg to out msg
  - take min of src msg and local state
  - scatter out msg to S.1



src=0, dst=6

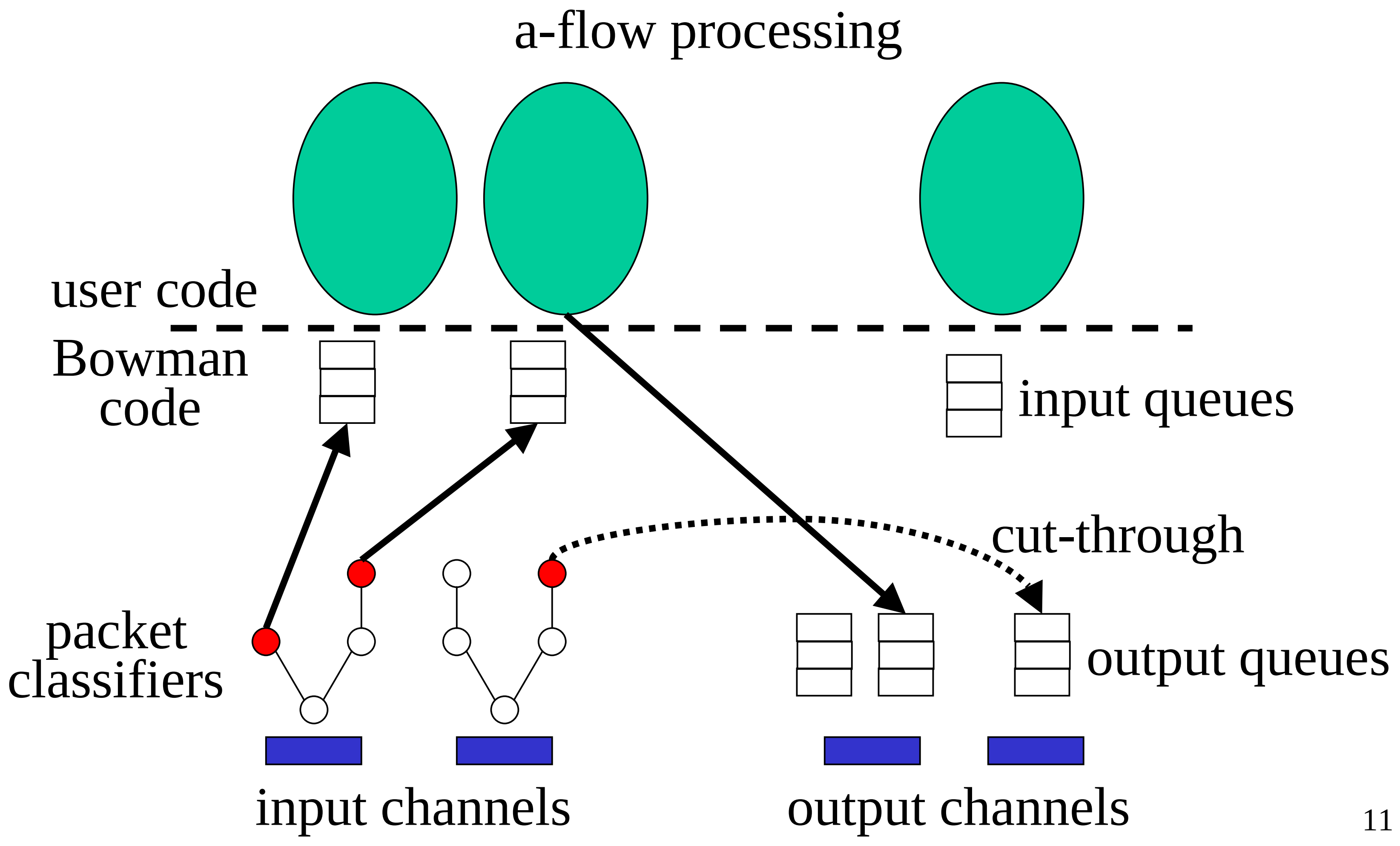
Generalizations

- Multicast tree information retrieval
  - change dst unicast address to group address
  - assume routing table lookup returns set of interfaces for multicast tree
- Repair server location
  - change computation to check if packet loss on downstream links exceeds threshold
  - record identity of links in reply message

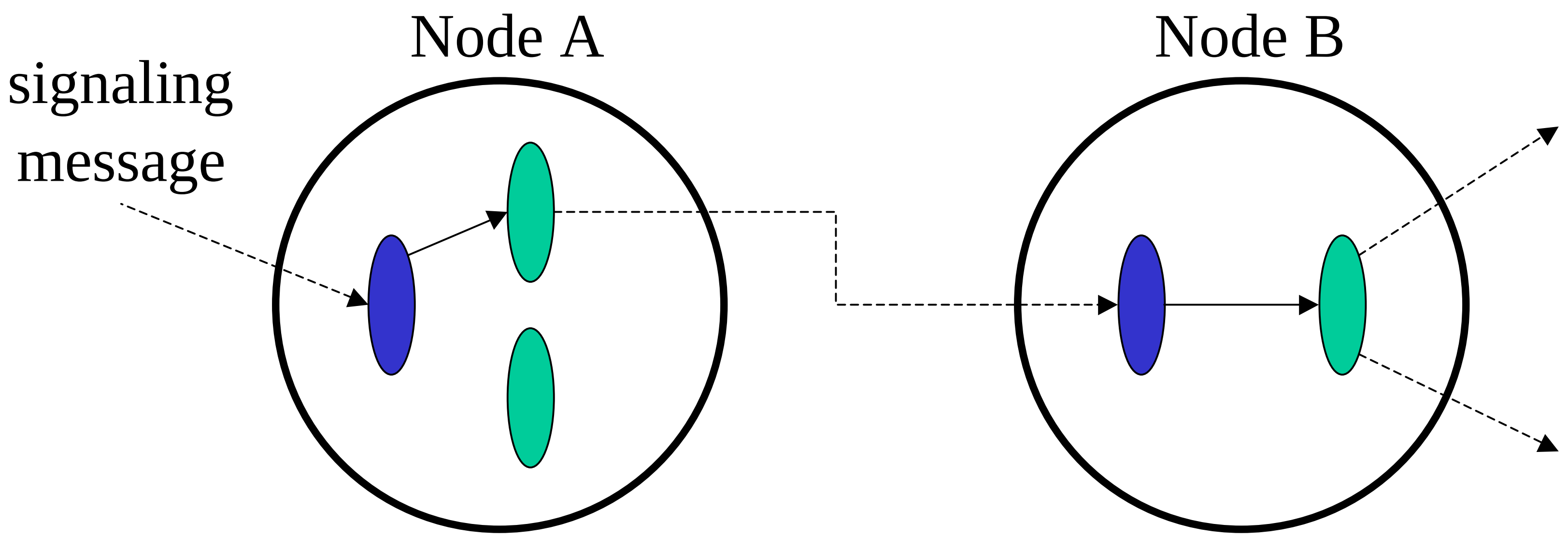
Implementation: Background

- CANEs
  - execution environment supporting composition
  - underlying program (skeleton pkt processing)
  - injected programs (customize skeleton)
- Bowman
  - active node OS (built over hostOS)
  - provides channels, a-flows, state-store,...
  - plug: Infocom paper/talk Thursday morning

Packet Processing Path



IGCS system architecture



● IGCS daemon: node-resident; parses signaling messages, initiates new instance with proper code

● IGCS instance: underlying IGCS program and injected programs for specific activity

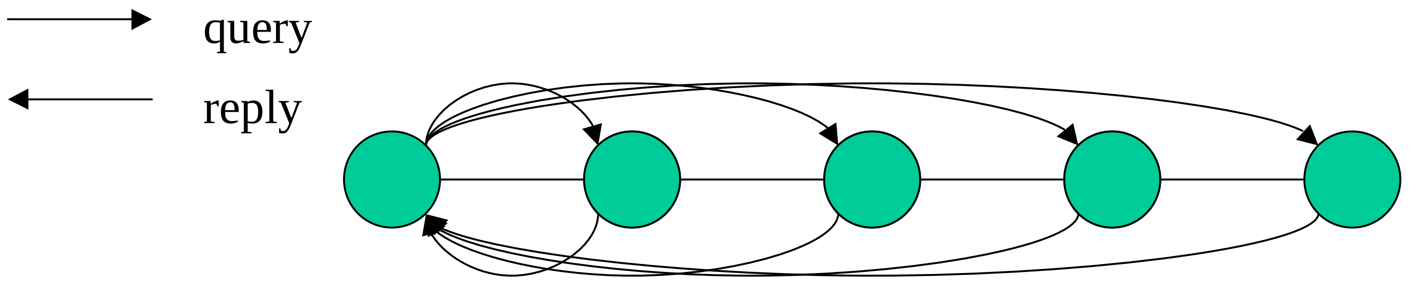
Underlying program

```
for (i = 0; i < igcs_get_iteration(); ++i) {
    /* Gather Phase */
    igcs_get_gather(i, tmp_io);
    if (tmp_io->num_ch != 0) {
      igcs_install_gather_filter(tmp_io);
      cur_in = igcs_gather_msg(tmp_io);
      c_Ep(inMsg) = cur_in;
      igcs_uninstall_gather_filter(tmp_io);
    }
    /* Compute Phase */
    igcs_raise_slot(Compute);
    /* Scatter Phase */
    igcs_get_scatter(i, tmp_io);
    igcs_scatter_msg(c_Ep(outMsg),tmp_io);
  }
```

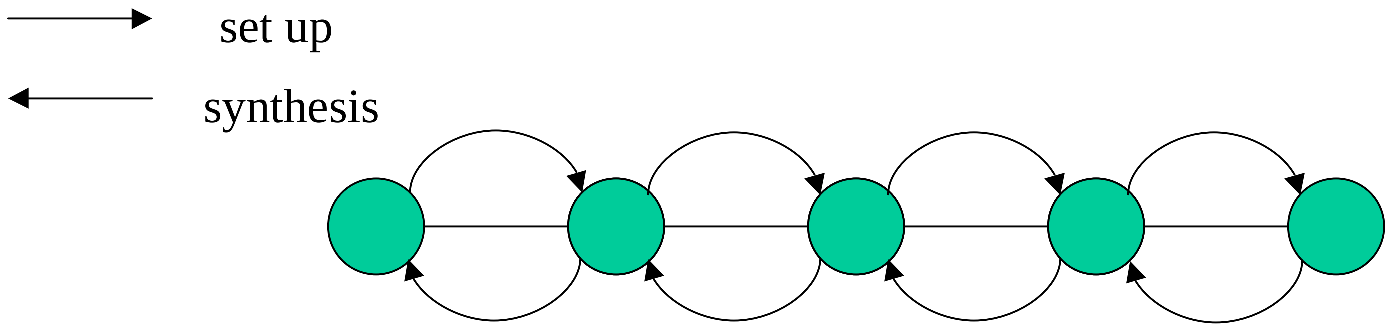
Injected program

```
memcpy(tmp_msg, (igcs_msg_t *)igcs_get_sigmsg(),
tmp_sig->len);

i = igcs_next_hop(tmp_msg->src_id);
tmp_io.num_ch = 1;
tmp_io.ch[0] = i;
igcs_set_scatter(1, &tmp_io);

igcs_get_all_vn_channel(&tmp_io);
_igcs_get_diff_channel(&tmp_io, i);
igcs_set_scatter(0, &tmp_io);
igcs_set_gather(1, &tmp_io);

tmp_msg->src_id = net_utils_local_ip_number();
tmp_msg->type = IGCS_SIG;
c_Ip(outMsg) = tmp_msg;
```

Experiment

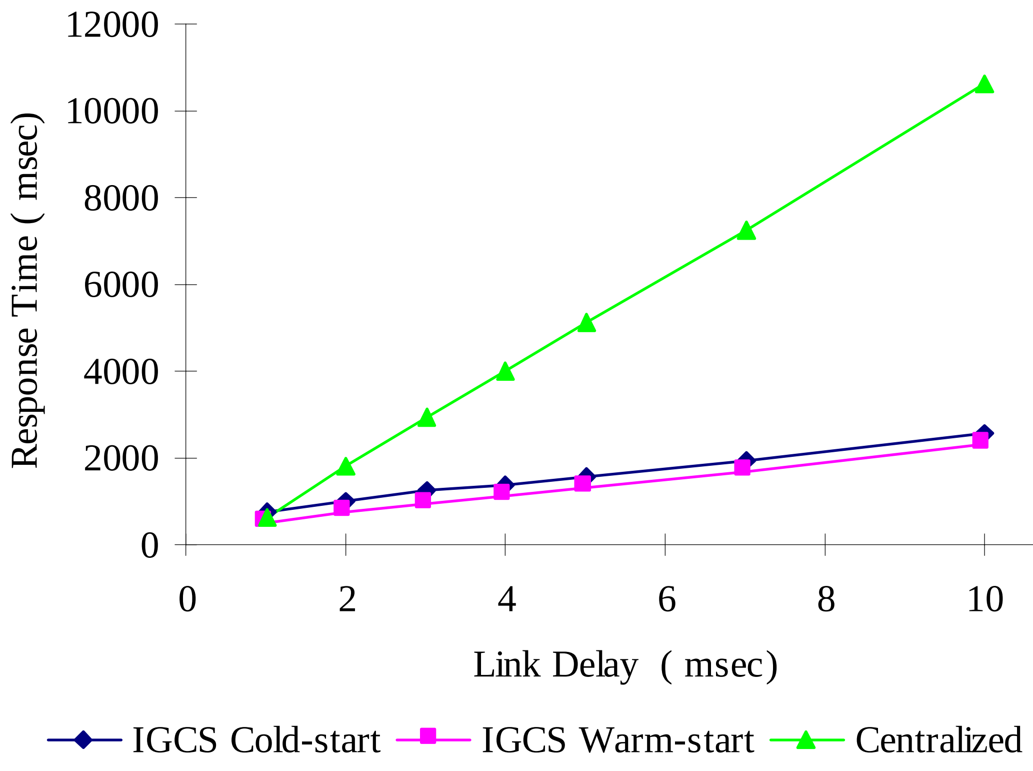


Centralized algorithm

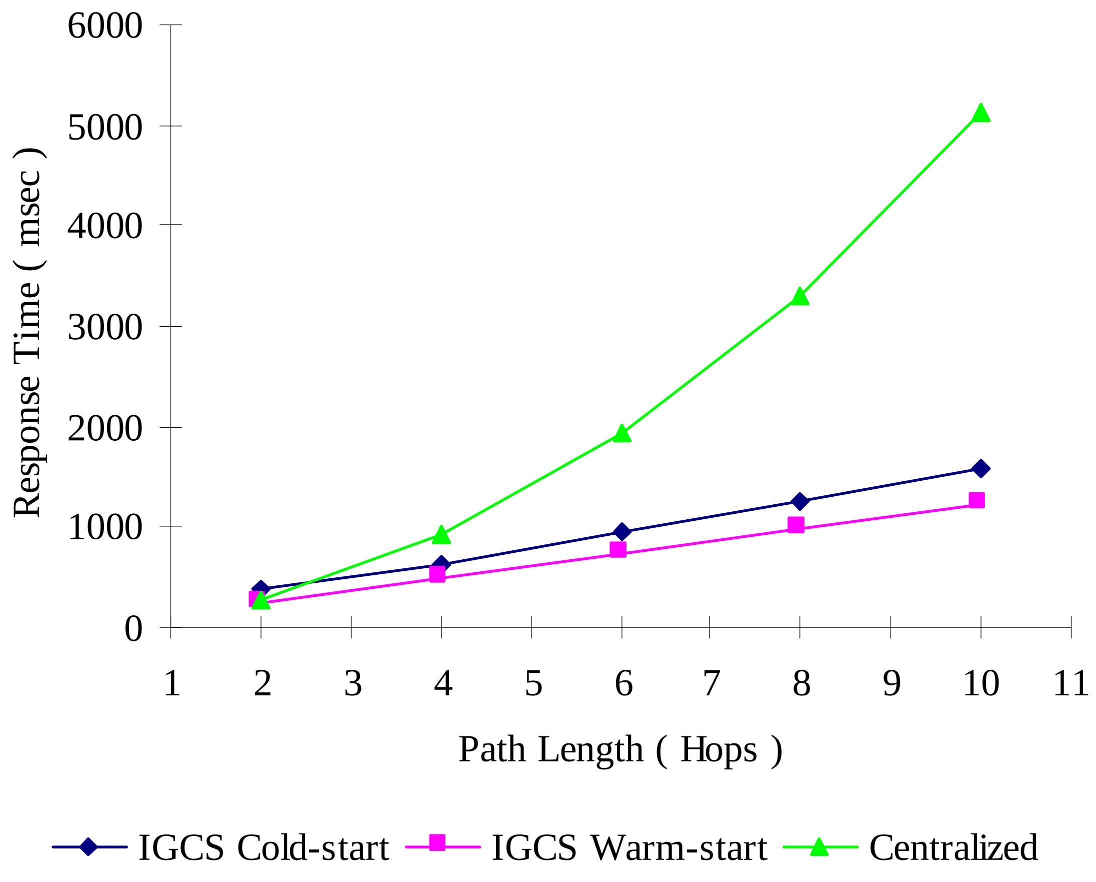


IGCS algorithm

Effect of link delay



Effect of path length



Concluding remarks

- IP provides black box to network topology
- Selectively opening the box makes some useful functionality possible (or easier)
- Key is to balance flexibility with performance and security
- Open issues: access control for node and link state, route changes during computations, integration w/virtual topos