CS 268: Project Suggestions

Ion Stoica

January 23, 2006

Overview

- Will present 25 project suggestions
- This lecture: class projects leveraging internal systems
- Next lecture: other projects
  - Architecture
  - Theory
  - Systems

Background

- Leverage and/or expand internal systems
- Internet Indirection Infrastructure (i3)
- Overlay Convergence Architecture for Legacy Applications (OCALA)
- Distributed debugging (LibLog)
- Next, three short presentations on each of these projects (see associated files)

3

Project 1: IP Multicast Support in OCALA

- Current implementation supports only IP unicast applications
- Add IP multicast support
  - If overlay/network architecture implements multicast, OCALA should enable IP multicast applications to use it
- Implement and evaluate IP multicast support on top of i3

4

Project 2: Distributed Firewall

- Today each company/user manages its/her own firewall
  - Hard to configure and maintain
- Provide firewall functionality as a service
  - A user can have all her packets forwarded through the firewall irrespective of where/how is connected to the Internet
  - Firewall functionality distributed across a set of servers
  - Centrally managed
- Possible implementation:
  - Use i3 (or DOA) for indirection, and for implementing signaling protocol between firewall servers
  - Use OCALA to support legacy applications

5

Project 3: Signaling Protocol for Middleboxes

- Design a signaling protocol to accommodate middleboxes/services
- Research issues:
  - Authentication of middleboxes
  - Transparent recovery: when one middlebox fails another equivalent middlebox can take over
  - Challenge: recovery transparent to end-hosts at transport layer

6

Projects 4-6: OCD Modules (Each Bullet → One Project)

1) Packet-level compression
   - Compress packet content / cache most common packets

2) Quality of Service
   - Both for outgoing and incoming (i.e., TCP) connections
   - Allow users allocate resources to each type of connection
   - Use weighted round-robin (WRR) or Hierarchical-Fair Service Curve (H-FSC) as scheduler

3) Allow a host behind a symmetric NAT or restrictive firewall behave as a host behind a cone-NAT that can be configured by the user
   - Application example: allow Bittorent clients behind NATs/Firewall run efficiently

7

Project 7: Composable OCD Architecture

- OCD modules cannot be composed
  - One cannot take the input from one OCD module and feed it into another OCD module
  - E.g., not possible to send i3 traffic over HTTP or DNS without modifying i3!
- Design an OCD Architecture in which modules can be layered on top of each other
- Challenges:
  - Come up with OCD descriptions and rules (language?) that say which OCDs can be composed and how
  - Configuration, management, …

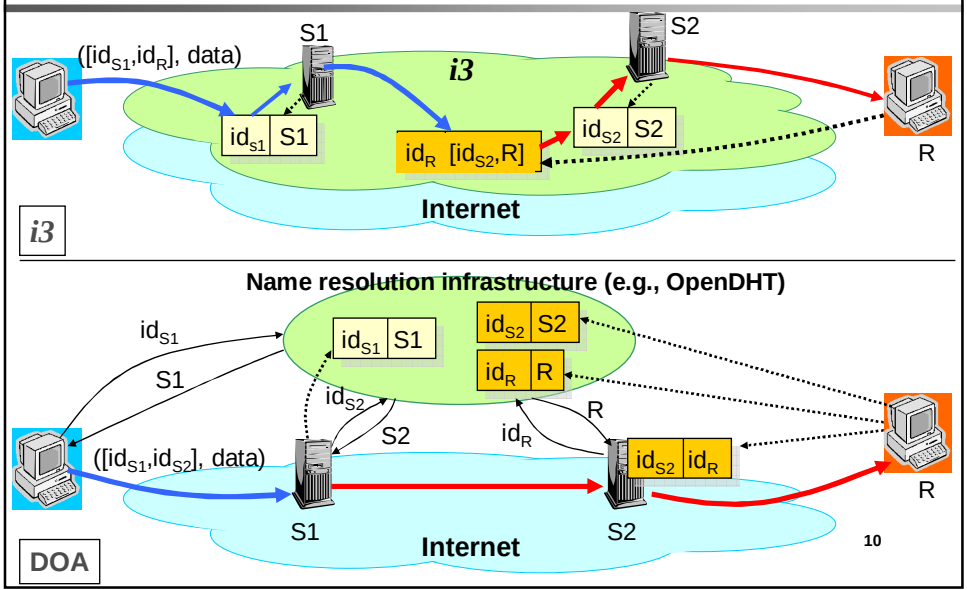
8

"ID-based" Architectures

- Decouple the identity of an end-host/service from its address
- At transport level, sender sends packet to an ID, not an address
- Examples
  - Delegation Oriented Architecture (DOA) [http://nms.lcs.mit.edu/doa/]
  - Host Identity Protocol (HIP) [http://www.ietf.org/html.charters/hip-charter.html]
  - Internet Indirection Infrastructure (i3) [http://i3.cs.berkeley.edu/]

9

i3 vs. DOA



10

Projects 8-9: i3 Applications (One Bullet → One Project)

- File sharing application
  - Firewall/NAT support
  - Leverage anycast
  - Anonymity
- Large scale multicast system
  - Decouple the data and the control planes
  - Can compute the multicast tree in a centralized fashion

11

Project 10: ID based Transport Protocols

- Design a transport protocol that allows end-hosts to be replaced in the middle of the transfer!
- Scenario:
  - You are talking at your cell-phone
  - You enter in the office
  - Your call is transferred to your computer, without interrupting the call
- Research: refactor transport such that
  - Congestion control state binds to address
  - Data transfer state binds to ID

12

Project 11: Connection-based Middlebox-aware Architecture

- Existing middlebox-aware architectures such as i3 and DOA present a network level interface
  - TCP/UDP connections are end-to-end
- Cannot support middleboxes implementing application-level functionality:
  - E.g., filtering on keywords in payload, caching, compression
- Design an architecture that terminates TCP connections at middleboxes
  - Avoid breaking TCP semantics

13

Project 12: Event Notification System

- Users specify events in which they are interested as a conjunction of attributes, e.g.,
  - (stock="msr") and (share_price > 60)
  - (source="Berkeley") and (destination="North Lake Tahoe") and (time < 3.5 hours)
- Research: create an efficient delivery tree
  - Users with the same interest grouped under the same tree
  - Users in the same geographic region grouped under the same tree

14

Project 13: Light-weigh LibLog

- Logging and replay is not feasible in distributed systems with limited resources (e.g., sensor nets, nanobots, etc.)
- Design a light-weigh approach to logging that is both power and space efficient
- Research questions:
  - Is it necessary to log everything?
  - What events can be discarded without masking the bug from the programmer?

15

Project 14: Distributed Invariants

- Goal: automatically figure out relevant invariants of the application so that programmers don't have to input them manually
- Possible approach: adapt Daikon to discover distributed system invariants during replay. When such invariants are violated, alert the user
  - Daikon: a dynamic invariant detector) to the distributed case
  - See: http://pag.csail.mit.edu/daikon/

16

Project 15: Efficient Checkpointing

- Build a checkpoint writing and transmission system based on LBFS (Low-bandwidth Network File System) to reduce communication and storage overhead
  - http://www.fs.net/sfswww/lbfs/
- Integrate it with LibLog

17