Overlay Networks

Enhancing the Internet ...

Seminar in Distributed Computing 2007

Ramon Schwammberger
ramons@ethz.ch
Contents

Part One

- overlay networks
- INS - intentional naming system
- i3 - Internet Indirection Infrastructure
- Active Names

Part Two

- delivery modes:
  - anycast
  - multicast
- late binding
- inter node routing
- caching
- soft state
- scalability
- security
Overlay Network

• build a new layer which forms an overlay network on top of the “IP network”

![Diagram]

- application
- transport (TCP/UDP)
- network (IP)
- link
- physical
goals - overview

**INS**
- expressiveness
- responsiveness
- robustness
- easy configuration

**i3**
- multicast
- anycast
- mobility
- end-to-end principle

**Active Names**
- customization / extensibility
- composability
- efficient resource management
- location independent execution
From Metanet to INS

- use a central network
- have a name and an address for everything
  “model the world”

“Metanet View”

- IPv4 Internet
- LAN
- GPRS Network
- WLAN
- Sensor Network
From Plutarch to INS

“INS View”

Node Network

Service

Service / Client

Service / Client

Client
INS – resolver network

- INR
- DSR

Service / Client

Service

Client
INS – name-specifier

- building blocks: attribute=value pairs

```
[city=zurich
 [building=IFW
  [floor=B
   [room=42]]]
[service=camera
 [data-type=picture
  [format=jpg]]
[resolution=640x480]]
[accessibility=public]
```
INS - example

INS message:

\{ B \mid D \mid \text{destination pointer} \mid \text{source pointer} \mid \text{body}\}

B – binding flag (early or late)
D – delivery flag (true -> multicast)

destination

[\text{service=camera [entity=transmitter]] [room=510]}

source

[\text{service=camera [entity=receiver]] [id=r] [room=500]}]
From INS to i3

- introduce rendez-vous based sending / receiving
- remove “model the world”
  - use a simple naming scheme
  - name rendez-vous points not services
- change the architecture of the resolver network
## goals - overview

<table>
<thead>
<tr>
<th>INS</th>
<th>i3</th>
<th>Active Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>expressiveness</td>
<td>multicast</td>
<td>customization / extensibility</td>
</tr>
<tr>
<td>responsiveness</td>
<td>anycast</td>
<td>composability</td>
</tr>
<tr>
<td>robustness</td>
<td>mobility</td>
<td>efficient resource management</td>
</tr>
<tr>
<td>easy configuration</td>
<td>end-to-end principle</td>
<td>location independent execution</td>
</tr>
</tbody>
</table>
i3 – rendez-vous point

Host A
IP_A

{ 200022 | data }

2.

500000
{ 100011 | IP_A }
{ 200022 | IP_B }

i3 node

1.

3.
IP_B { data }

Host B
IP_B

address space: m=6
ID = XXXXXX
i3 - details

• packet header

{ ID | source | data }
{ ID_1, ID_2, … , ID_N | source | data }

• trigger

{ ID | receiver }
{ ID | receiver1, receiver2, … , receiverN }
Plutarch to Active Names

• Plutarch's interstitial functions
Plutarch to Active Names

- Active Names running programs
# goals - overview

<table>
<thead>
<tr>
<th>INS</th>
<th>i3</th>
<th>Active Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>expressiveness</td>
<td>multicast</td>
<td>customization / extensibility</td>
</tr>
<tr>
<td>responsiveness</td>
<td>anycast</td>
<td>composability</td>
</tr>
<tr>
<td>robustness</td>
<td>mobility</td>
<td>efficient resource management</td>
</tr>
<tr>
<td>easy configuration</td>
<td>end-to-end principle</td>
<td>location independent execution</td>
</tr>
</tbody>
</table>
Active Names - details

- hierarchical name space
  - programs define how remaining name shall be resolved
- micro kernel approach for resolver
  - loader
  - execution environment
  - interface for remote invocation of programs
Active Names - details

• after methods
  – describe the “path” from the service back to the client
  – can be reordered by any program
Active Names - example

sensor network

node 2

gathers
data

node 1

routes
request

node 3

processes
data

Host

(Client)
Contents

Part One

- overlay networks
- INS - intentional naming system
- i3 - Internet Indirection Infrastructure
- Active Names

Part Two

- delivery modes:
  - anycast
  - multicast
- late binding
- inter node routing
- caching
- soft state
- scalability
- security
delivery modes

**INS**
- switch between anycast and multicast (groupcast)

-> delivery flag

**i3**
- anycast
- multicast (groupcast)
- anycast between groups

-> address space xxyy
x-part for service
y-part for anycast

**Active Names**
- anything that can be programmed
late binding

**INS**
- switch between late and early binding

-> binding flag

**i3**
- always kind of late binding
- goes always through rendezvous point

**Active Names**
- is programmable
- most useful with late binding (send data for nodes)
inter node routing

**INS**
- always to next hop in INR spanning tree
- flooding to announce services

**i3**
- distributed hash table (Chord)

**Active Names**
- name tells the program where to send the packet to
caching

INS
• cache content

{ B | D | dst ptr | src ptr | cache lifetime | body }

i3
• cache IP of node which is responsible for ID

Active Names
• is programmable
• (content caching)
soft state

INS
• service advertisements are soft state
• easy recovery
• easy logout
• up to date

i3
• triggers are soft state
• up to date
• easy recovery

Active Names
• state is handled by programs
## Scalability

<table>
<thead>
<tr>
<th>INS</th>
<th>i3</th>
<th>Active Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>• not designed for wide area application</td>
<td>• designed for wide area</td>
<td>• hierarchical name spaces scale</td>
</tr>
<tr>
<td>• DSR not hierarchical</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• scales to several thousand services</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**scalability**

**INS**
- not designed for wide area application
- DSR not hierarchical
- scales to several thousand services

**i3**
- designed for wide area

**Active Names**
- hierarchical name spaces scale
security

INS
• not addressed in the paper
• needs to be explored

i3
• target: at least as secure as IP layer
• suggestions for
  – eavesdropping
  – trigger hijacking
  – DoS attacks

Active Names
• depends on “sandbox”
• depends on resolver programs
## conclusion

### INS
- good for services
- small scale
- good prototype

### i3
- useful enhancement of the internet
- wide area
- good prototype

### Active Names
- most adaptive solution
- complexity?
- security?
Thank you for your attention