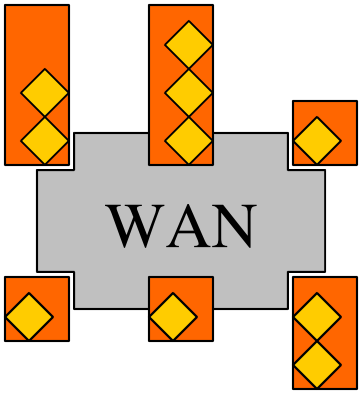
Ad Hoc Distributed Servers

Michal Szymaniak
Guillaume Pierre
Mariana Simons-Nikolova
Maarten van Steen

Problem

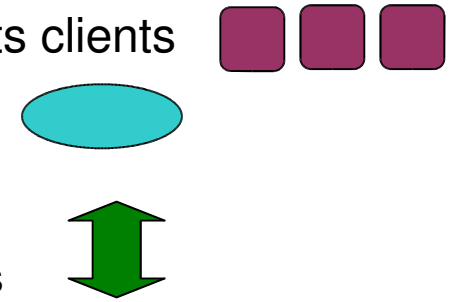
- Large-scale distributed system
  - E.g., CDN, Desktop Grid, P2P
- Node properties:
  - Distributed over WAN
  - Potentially with slow network connections
  - Potentially unreliable (come and go)

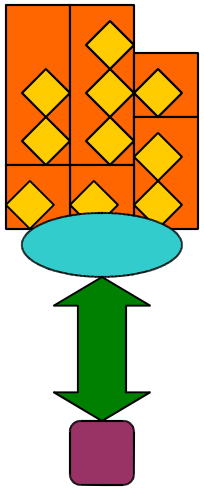


- Nodes provide resources
  - E.g., disk space, CPU power, network bandwidth
  - Resources inconvenient to use because of distribution and unreliability
- How to aggregate resources into stable wholes..
  - ..so that clients can use them efficiently?

Solution

- Ad Hoc Distributed Server: organized group of nodes
  - Looks like a big server to its clients
  - Despite node unreliability
    - Stable contact address
  - Despite node distribution
    - Single-connection access





- 2 design parts:
  - Internal organization – future work
  - External 'single machine' illusion – today

- We use Mobile IPv6 to implement the external part
- This talk: MIPv6/AHDS overview and example applications

Agenda

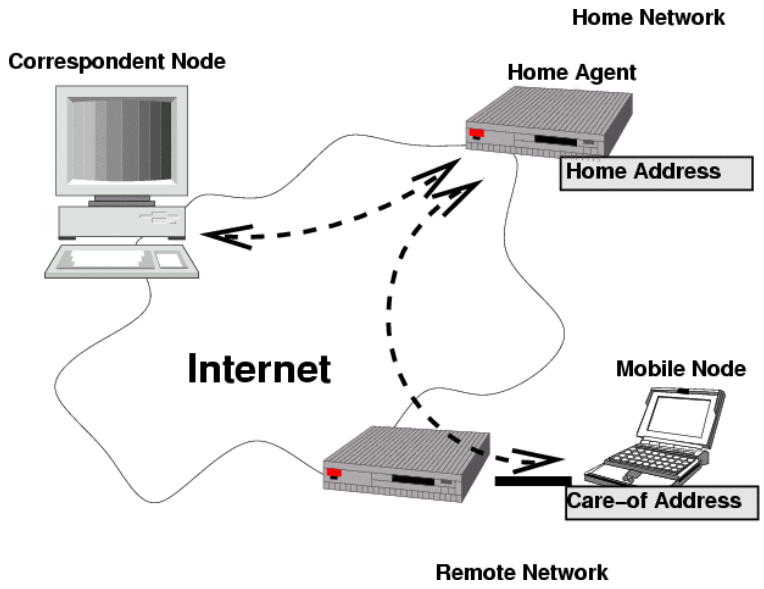
- Mobile IPv6 Overview
- Adapting Mobile IPv6 in AHDS
- Example Applications
- Open Questions

Mobile IPv6 Overview

- Mobile nodes reachable while away from home networks
  - Any node talking to mobile node is called correspondent node
- Routers in home networks represent mobile nodes
  - Any such router is called home agent
- Two addresses assigned to each mobile node:
  - Home address - identifies mobile node, never changes
  - Care-of address - represents mobile node's current location
- Goal: mobile nodes always reachable at their home addresses
- But how does it actually work?

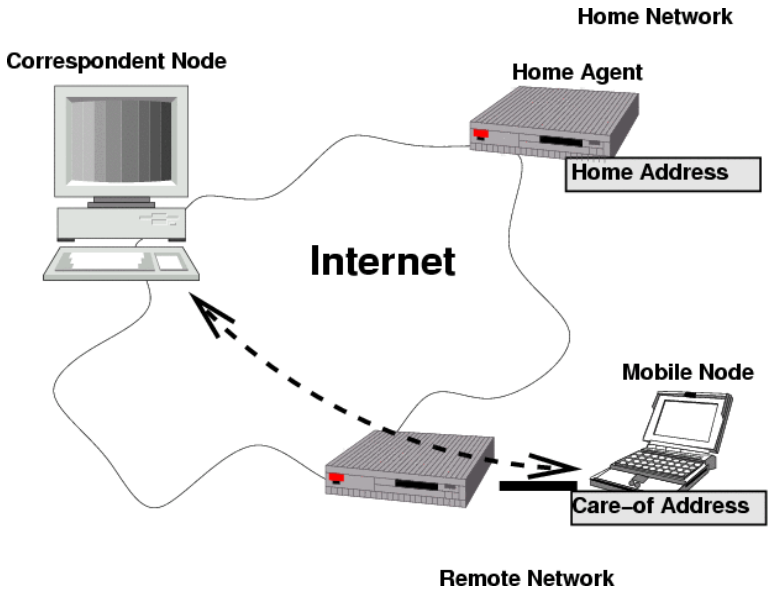
MIPv6: Tunneling

- When away, mobile node (MN) reports its current care-of address to its home agent (HA)
- HA tunnels traffic between MN's home address and MN's care-of address
- Transparent to correspondent nodes
- But:
  - Suboptimal routing
  - HA can become bottleneck



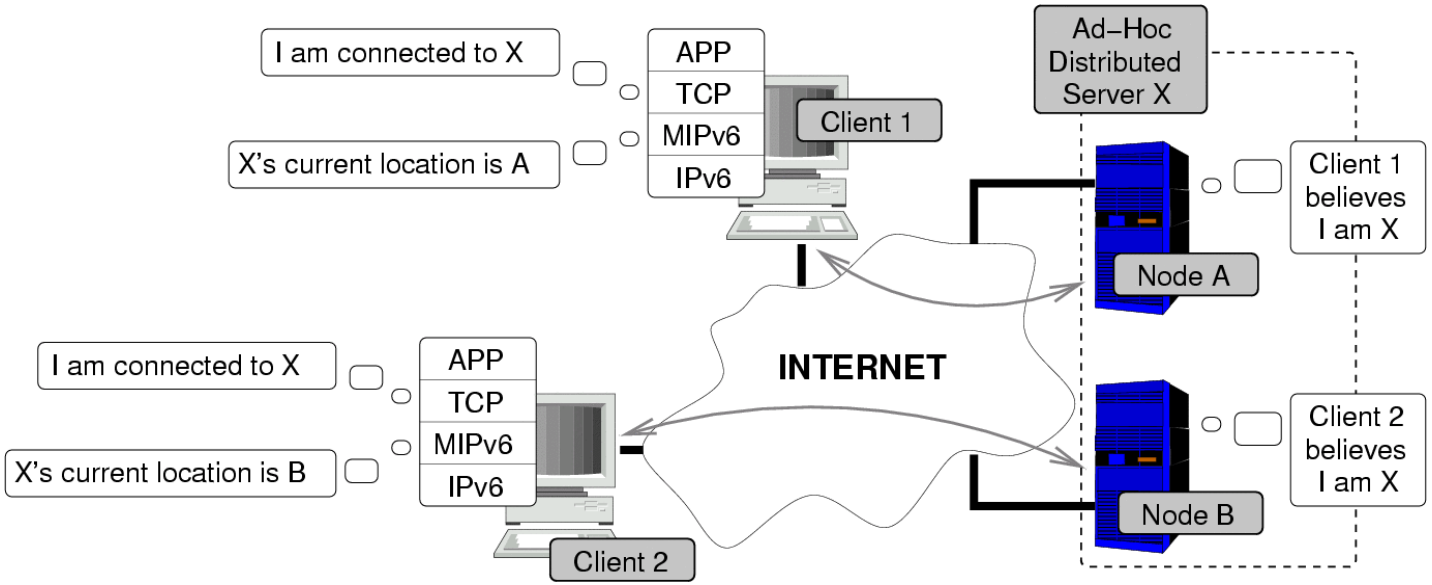
MIPv6: Route Optimization

- MN reveals its care-of address to correspondent node (CN)
- CN creates a translation mapping
  - Home address <=> Care-of address
  - Address translation in CN's IP layer
  - Higher layers see home address only
- Result:
  - Direct MN-CN communication..
  - ..with MN movements transparent to applications running on CN
- How can we use it in distributed servers?



Adapting Mobile IPv6 in AHDS

- Ad hoc distributed server pretends to be a SINGLE mobile node:
  - Home address == server's contact address (advertised to clients)
  - Care-of addresses == addresses of individual nodes
  - Server distribution transparent to clients



Example Applications

- Scalable/reliable super-peers, e.g., BitTorrent trackers
  - Unreliable p2p nodes organized into stable indexing facilities!
- Wide-area TCP handoff, e.g. for content delivery networks
  - Bi-directional traffic switching, no triangular routing!
- Stable joining/contact address, e.g., for p2p systems
  - Contact address held by some p2p member, no matter which one

Open Questions

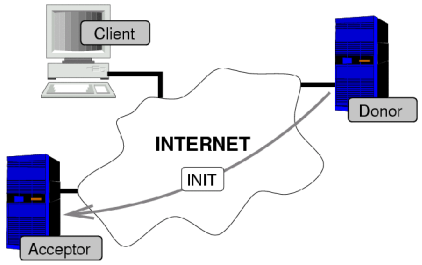
- Do you think your applications could run on AHDS?
  - Transparent handoffs / distributed nodes / single address
  - Any problems that come to mind?
- How interesting can it be to Grid people?
  - E.g., computations might transparently migrate among grid nodes
  - Any particular applications that could benefit from that?
- How about implementing AHDS on network cards?
  - Address mappings / redirection handled directly by NIC
  - NIC-based Home Agent?

Thank you!

Wide-Area Handoff

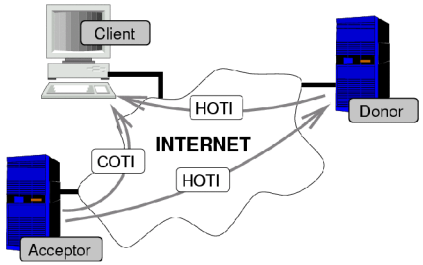
- Server must control client-side address translation
  - Translation mappings updated during route optimizations
  - Server mimics the route optimization protocol (b, c, d)
  - Slang: donor handoffs client; acceptor takes over client

a)



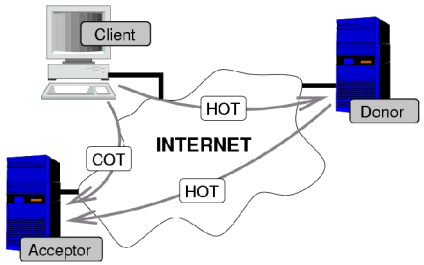
- Please take over this client

b)



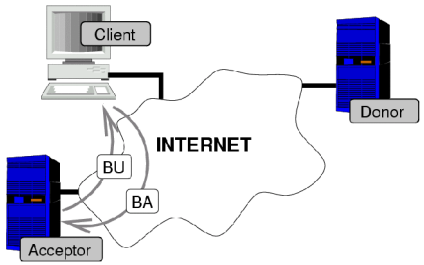
- Hi Client, here is my new care-of address

c)



- Really? Let me verify it then

d)

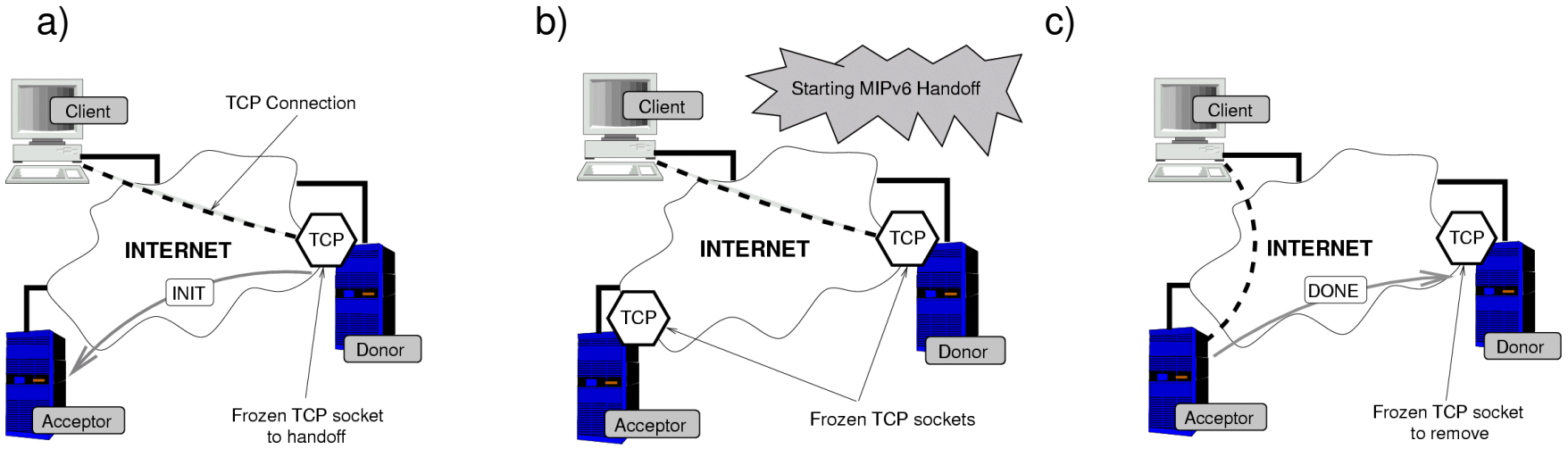


- Verification OK, please update your mappings

- Client now talks directly to acceptor on IP level, but..

Wide-Area Handoff ctd.

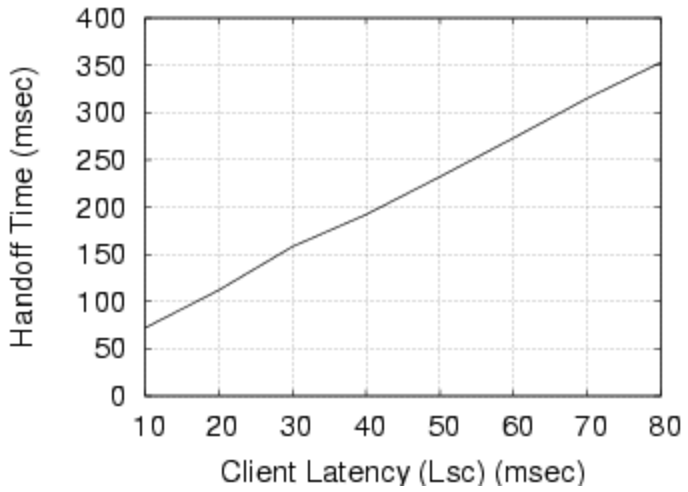
- ..we have just broken TCP connection :-(
- TCP connection state must be transferred to acceptor as well
  - Server-side TCP socket frozen to avoid changes in connection state
  - Two frozen socket instances to avoid accidental connection reset

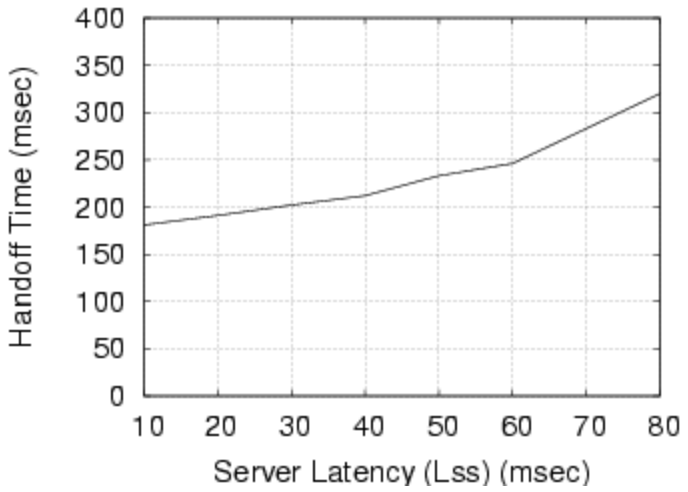


- Client is now connected to acceptor while believing it is donor :-)

Client-observed Handoff Time

- Delay between receiving data from donor and from acceptor
- Lss - latency between donor and acceptor
- Lsc - latency between client and either donor or acceptor
- After all optimizations: handoff time = Lss + 4 * Lsc

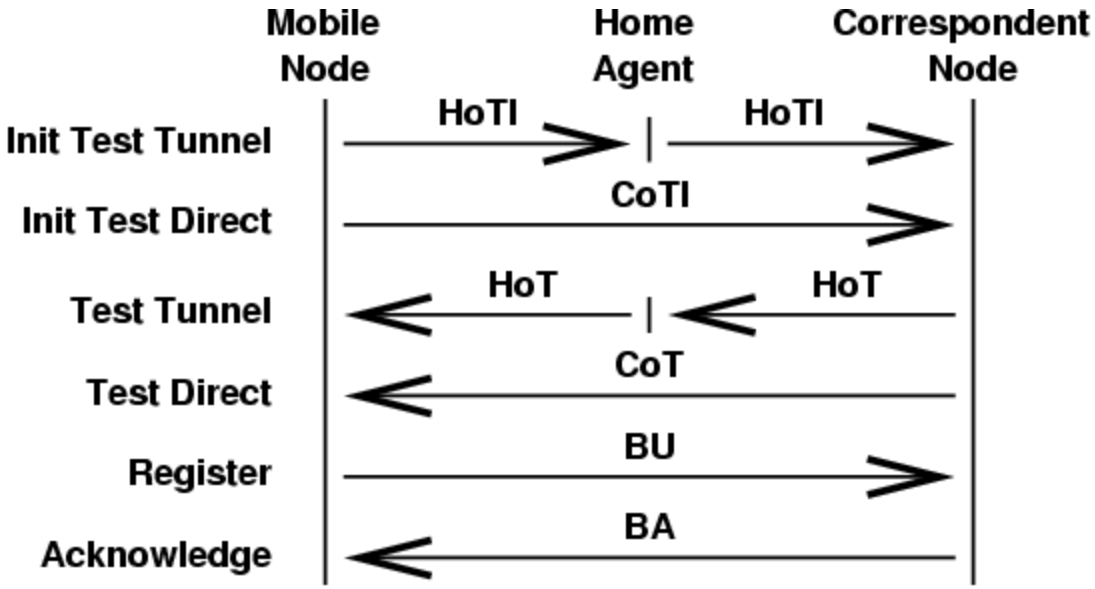




- Some optimizations assume low Lss; worst case: 3 * Lss + 6 * Lsc

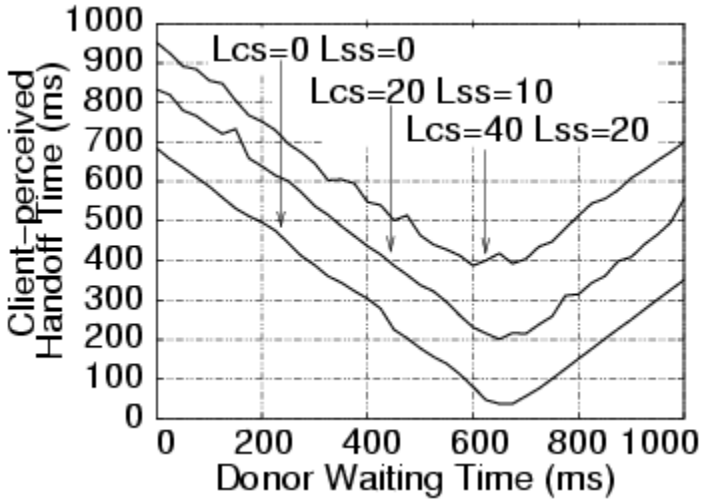
Route Optimization Protocol

- Tests prove that care-of address matches home address
- BU contains combined values of HoT and CoT
- Cryptography all over the place



State Transfer Optimization

- Server-side TCP socket might contain unsent/unacknowledged data
- Such data must be transferred to acceptor as well
- Better wait until socket buffers become empty:



Handoff Time Optimization

- Some messages (HoTI/CoTI and HoT/CoT) exchanged in advance
- Result: (3 * Lss + 6 * Lcs) reduced to (Lss + 4 * Lcs)
  - as long as messages are exchanged before actual handoff starts