

NETSERV: PROGRAMMING NETWORKS (AGAIN)

Jae Woo Lee, Roberto Francescangeli, Wonsang Song, Jan Janak,
Suman Srinivasan, Michael S. Kester, Salman Baset, Eric Liu
and Henning Schulzrinne
Internet Real-Time Lab, Columbia University

*In collaboration with Volker Hilt at Bell Labs/Alcatel-Lucent and
Zoran Despotovic and Wolfgang Kellerer at DOCOMO Euro-Labs*

What is NetServ?

- In-network service container
- Java-programmable, signal-driven router
- “GENI Lite” – deploy modules, not VMs
- Active networking 2.0

Bridging two worlds

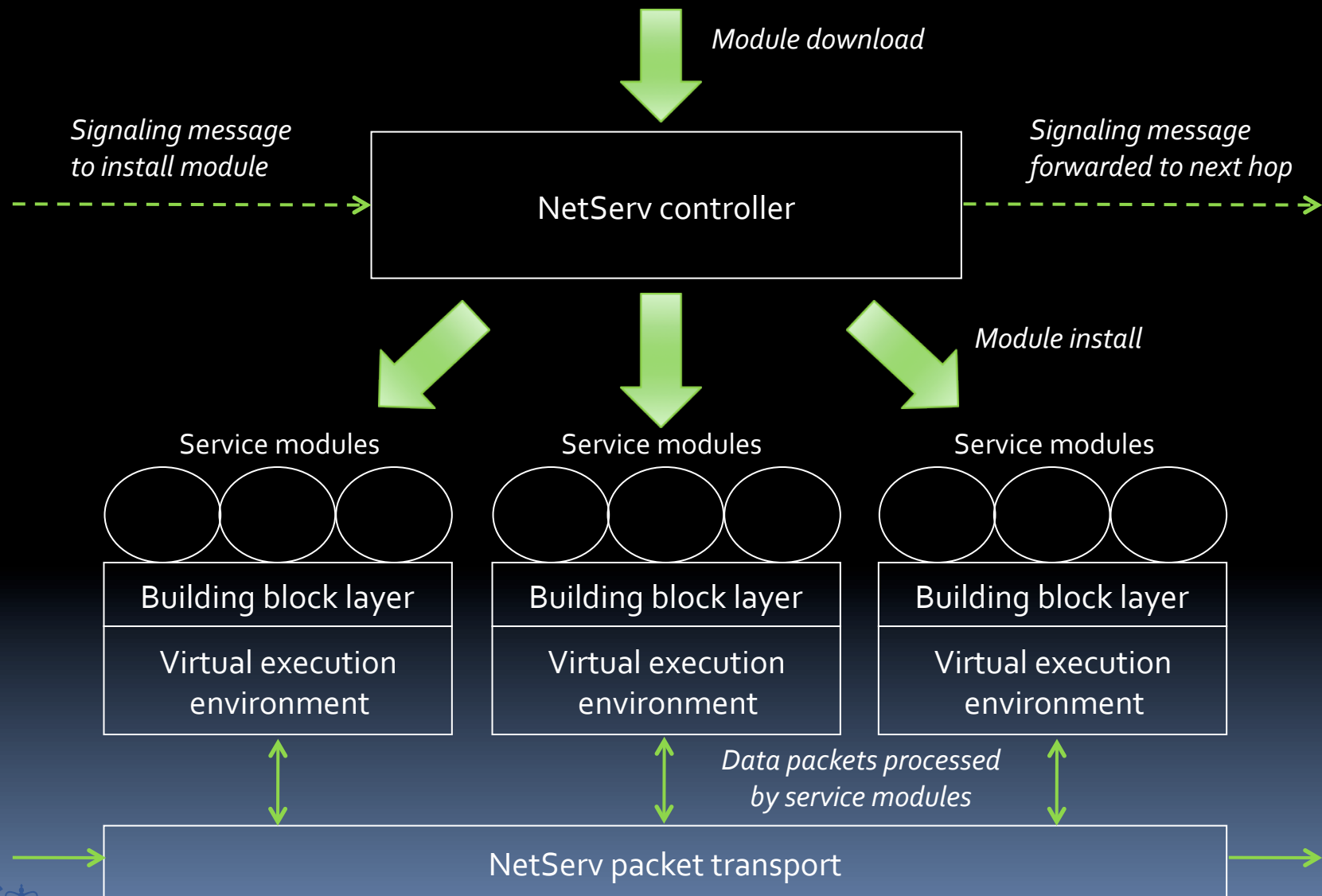


1 interface
TB disk
1-32 multi-core processors

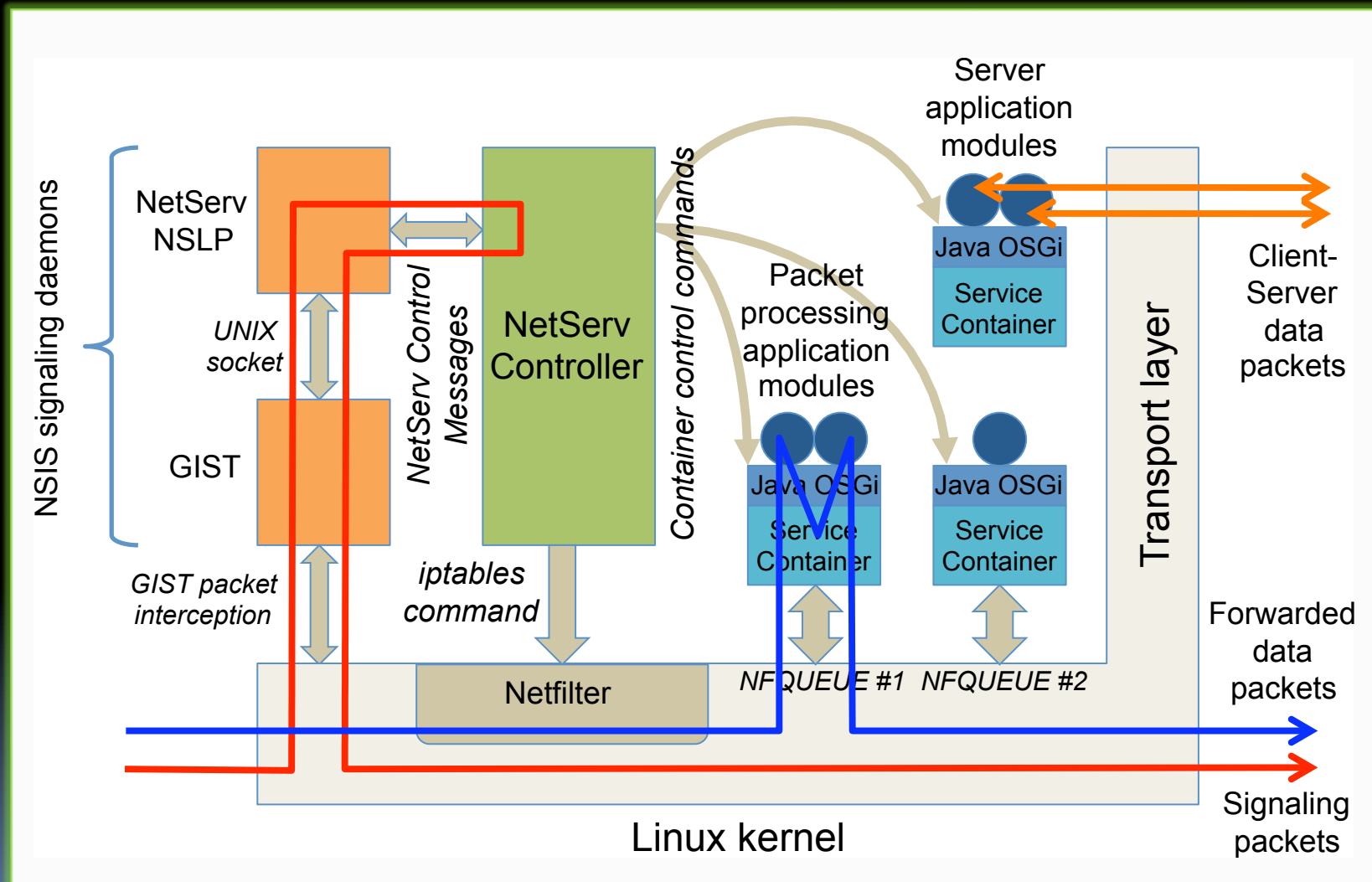


10+ interfaces
0 GB disk
1 low-end processor

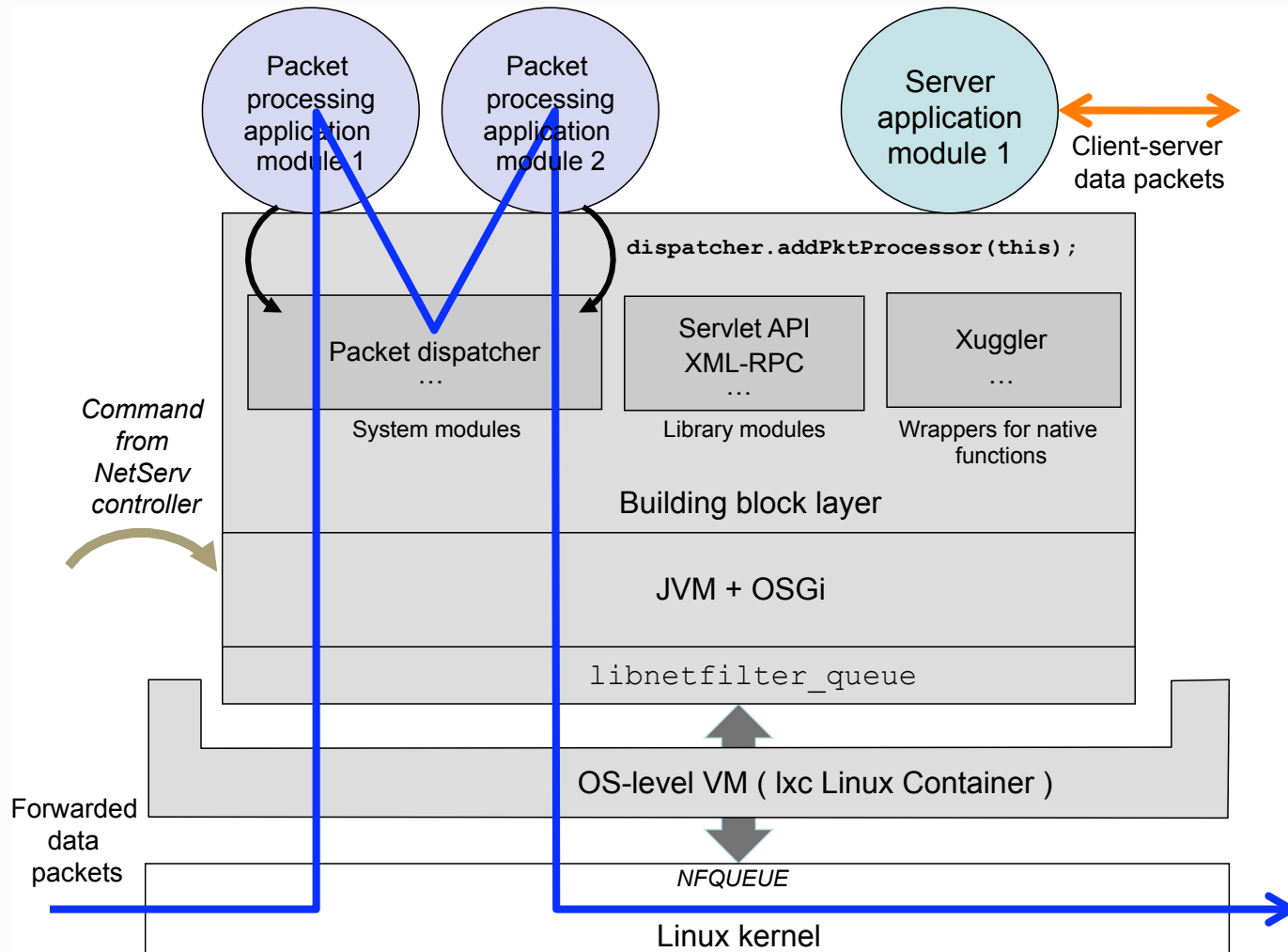
NetServ node architecture



Current prototype



Service container in detail



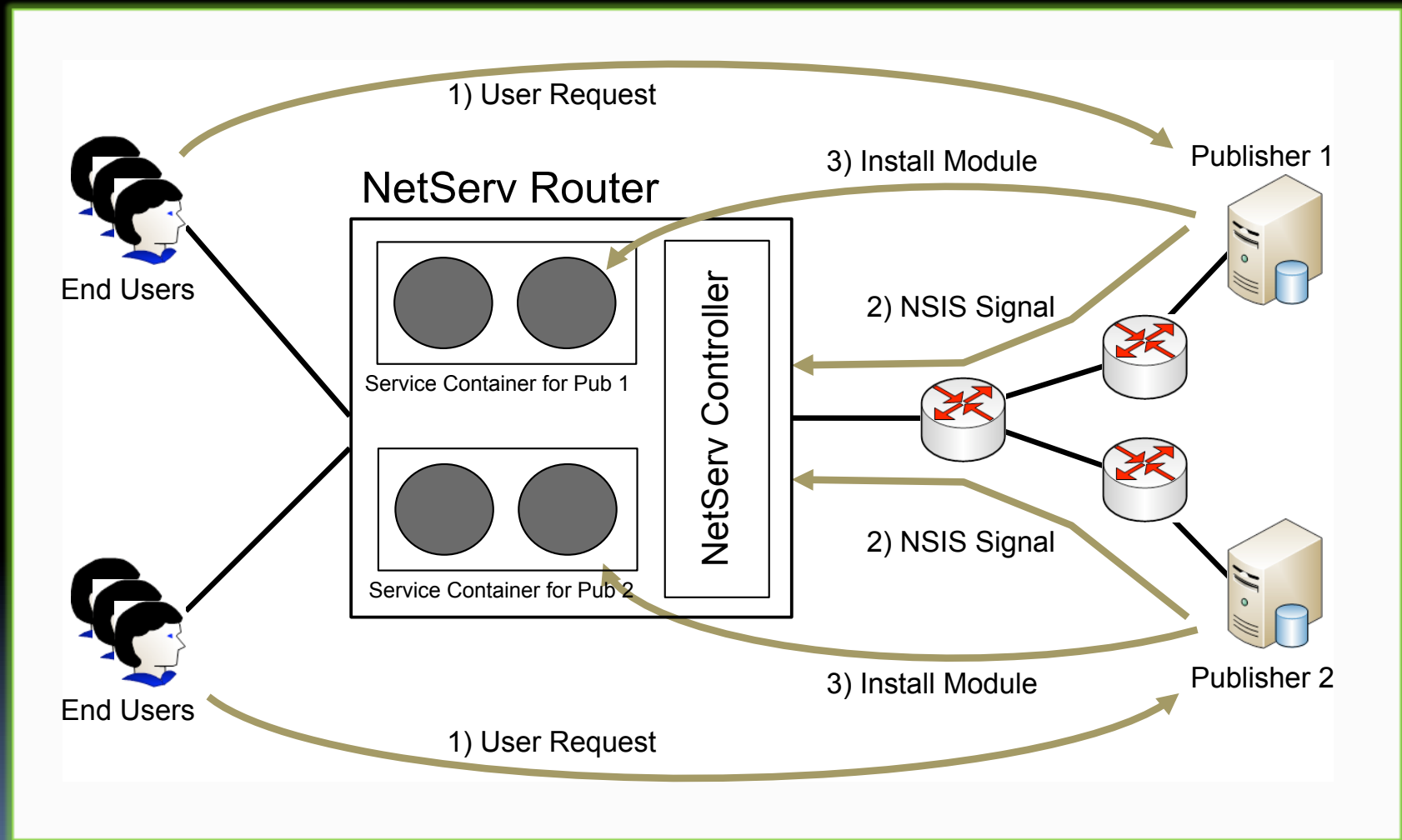
The grand vision

- NetServ everywhere
 - Common service API on router, PC, set-top box, ...
 - Storage and computation on network nodes
 - Enabling platform for NGI
- Internet is a multi-user computer
 - Code modules run anywhere
 - Secure and extensible
 - Active networking redux!

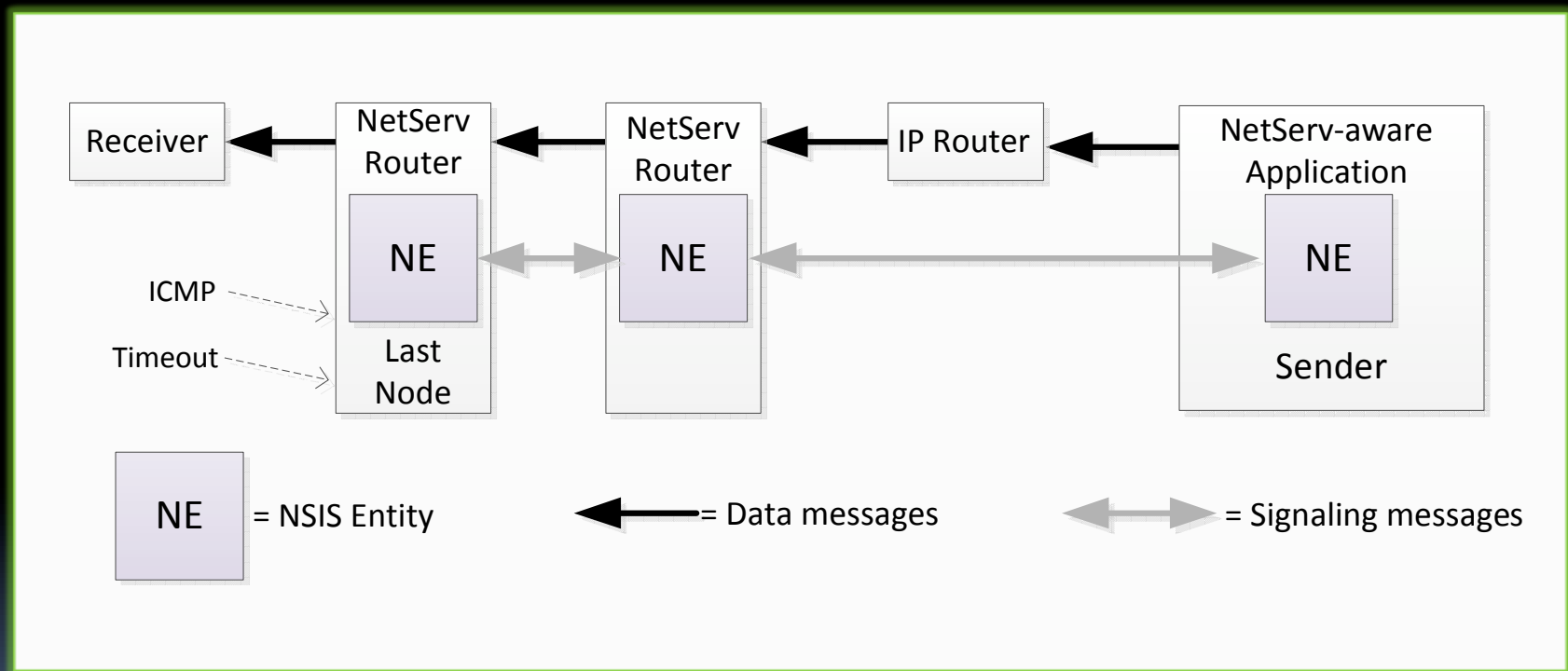
Not-so-grand initial focus

- Activate the network edge
 - Eyeball ISPs sell router resources to content publishers
 - Content publishers install servers and packet processors on edge routers
- Economic incentives
 - New revenue source for ISPs
 - Alternative to CDN for content publishers

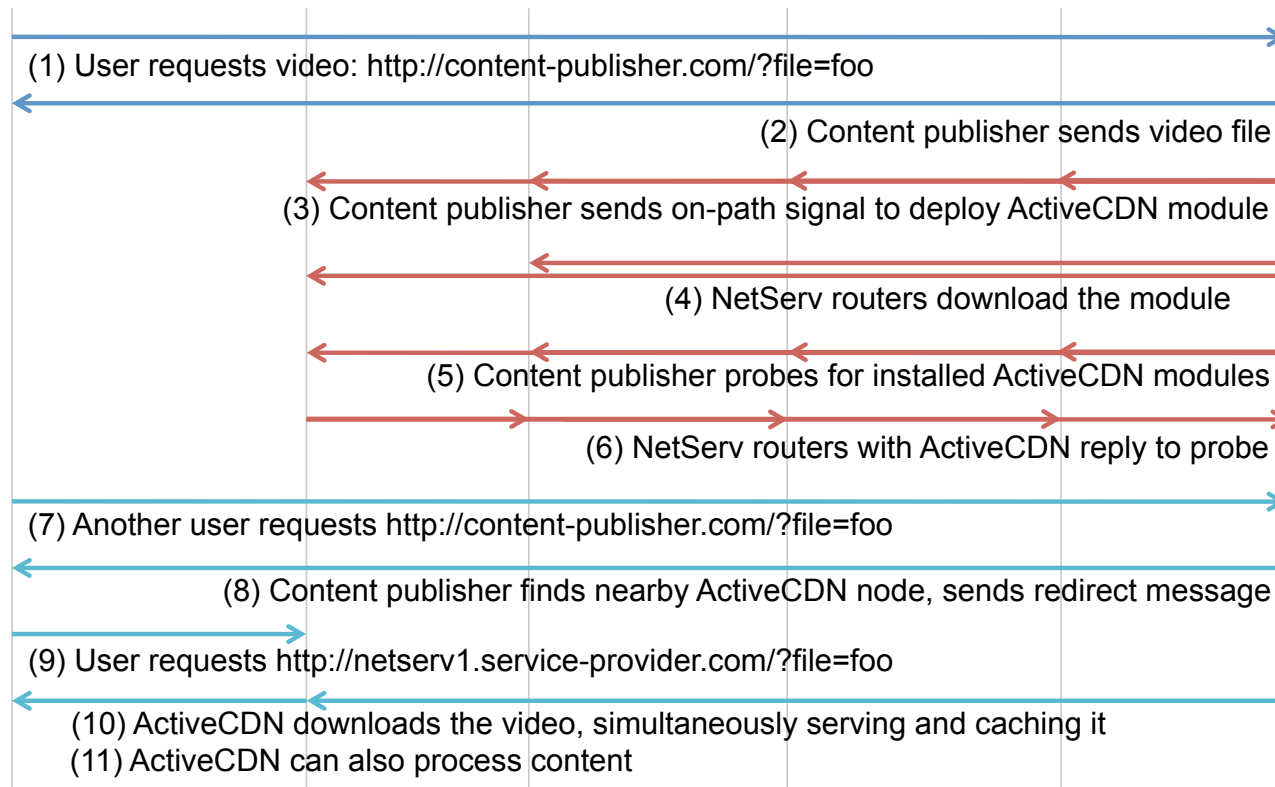
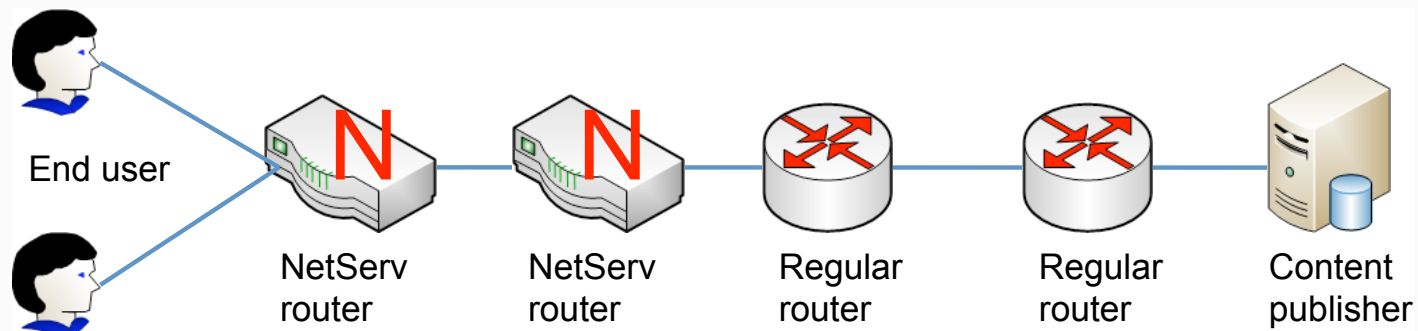
Overview of operation



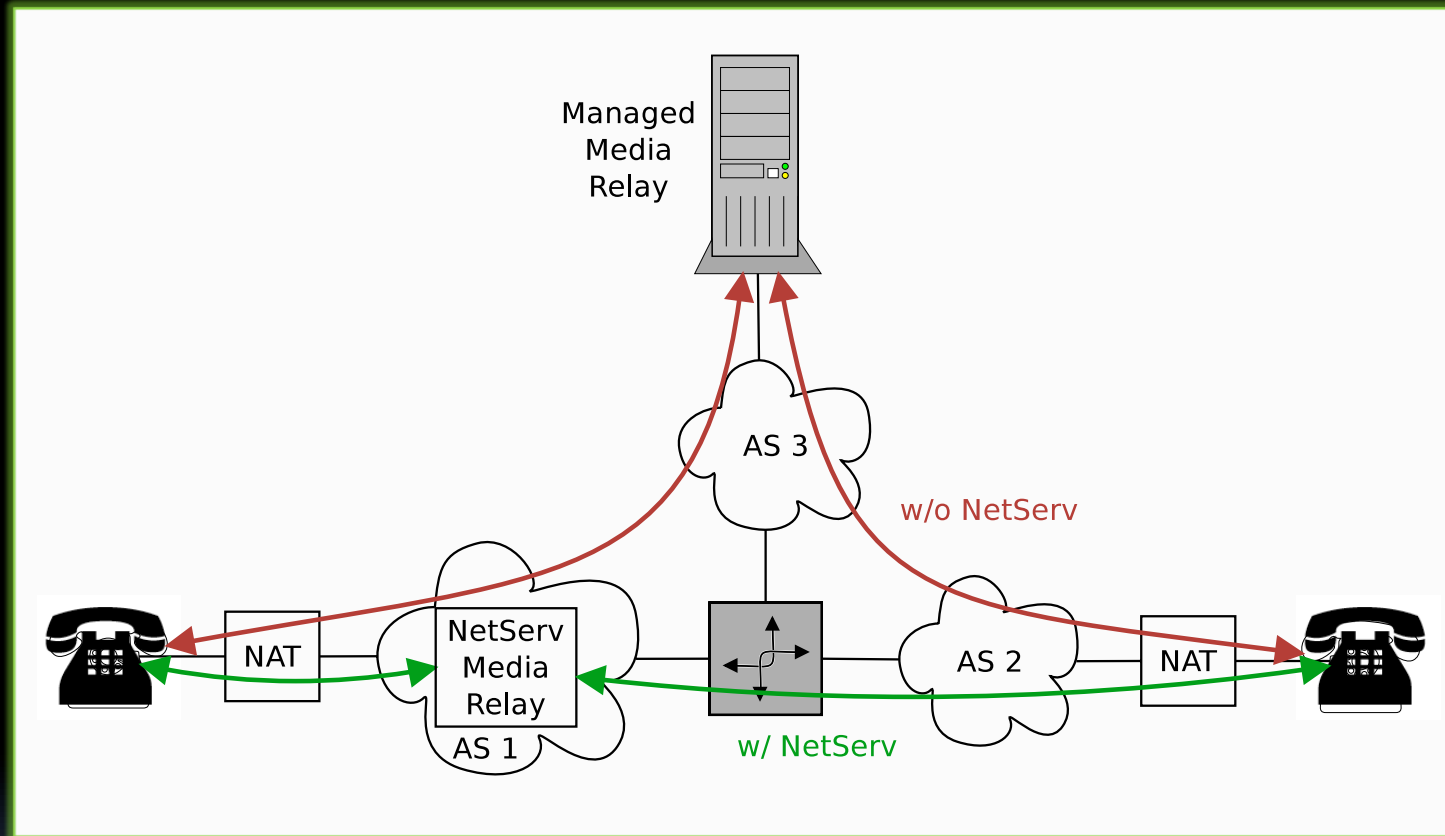
NSIS Signaling



Application: ActiveCDN

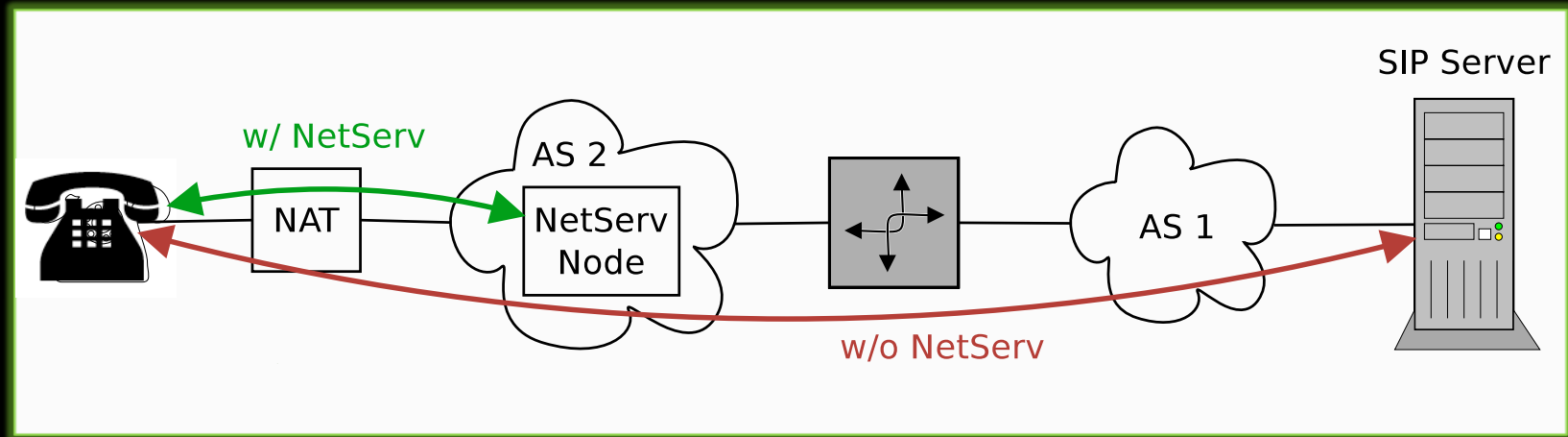


Application: Media relay



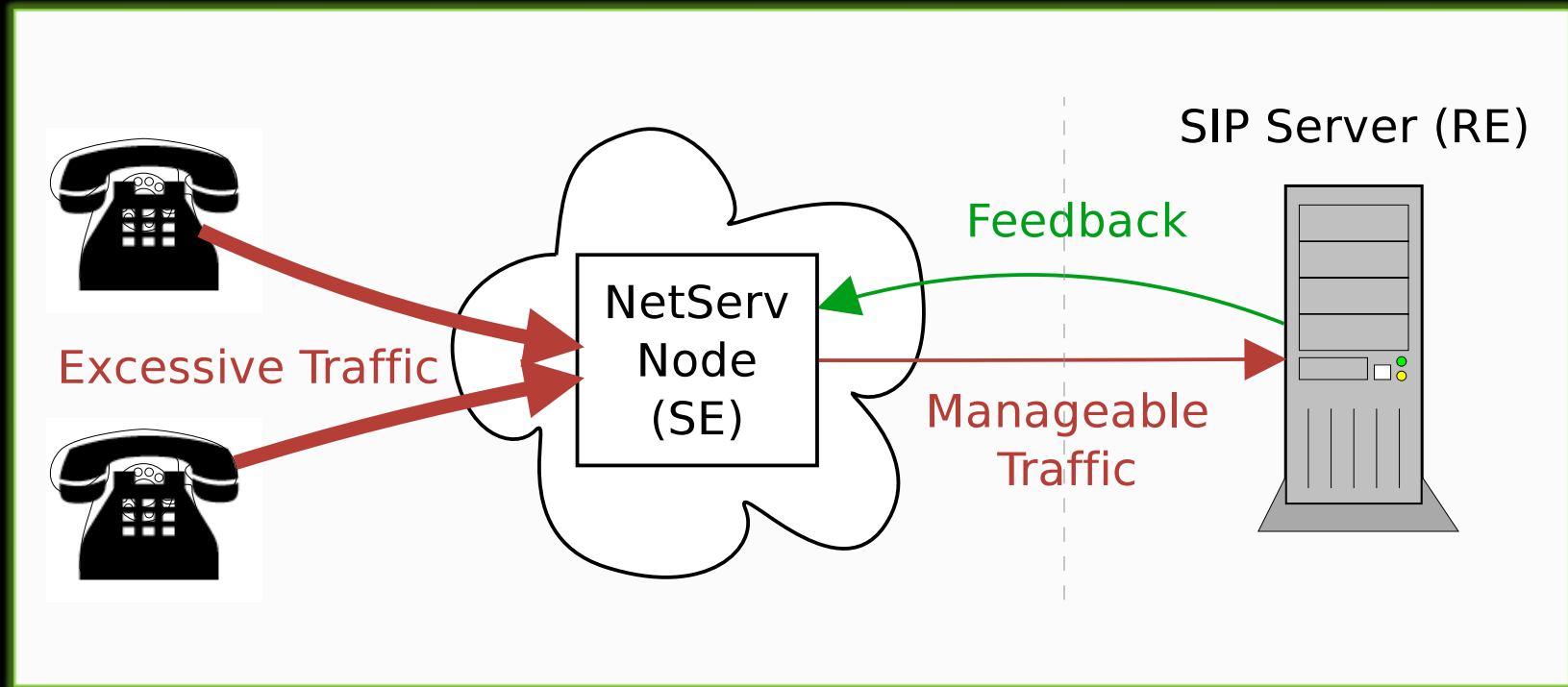
- Standard media relay
 - Required due to NAT
 - Out-of-path
 - Inefficient and Costly
- NetServ media relay
 - Closer to users
 - Improved call quality
 - Reduced cost for ITSP

Application: Keep-alive responder



- NAT Keep-alive responder off
 - UA behind NAT must send keep-alive messages
 - Major bottleneck for SIP server
- NAT Keep-alive responder on
 - Module responds on behalf of SIP server
 - No traffic to server

Application: Overload control



NetServ Module

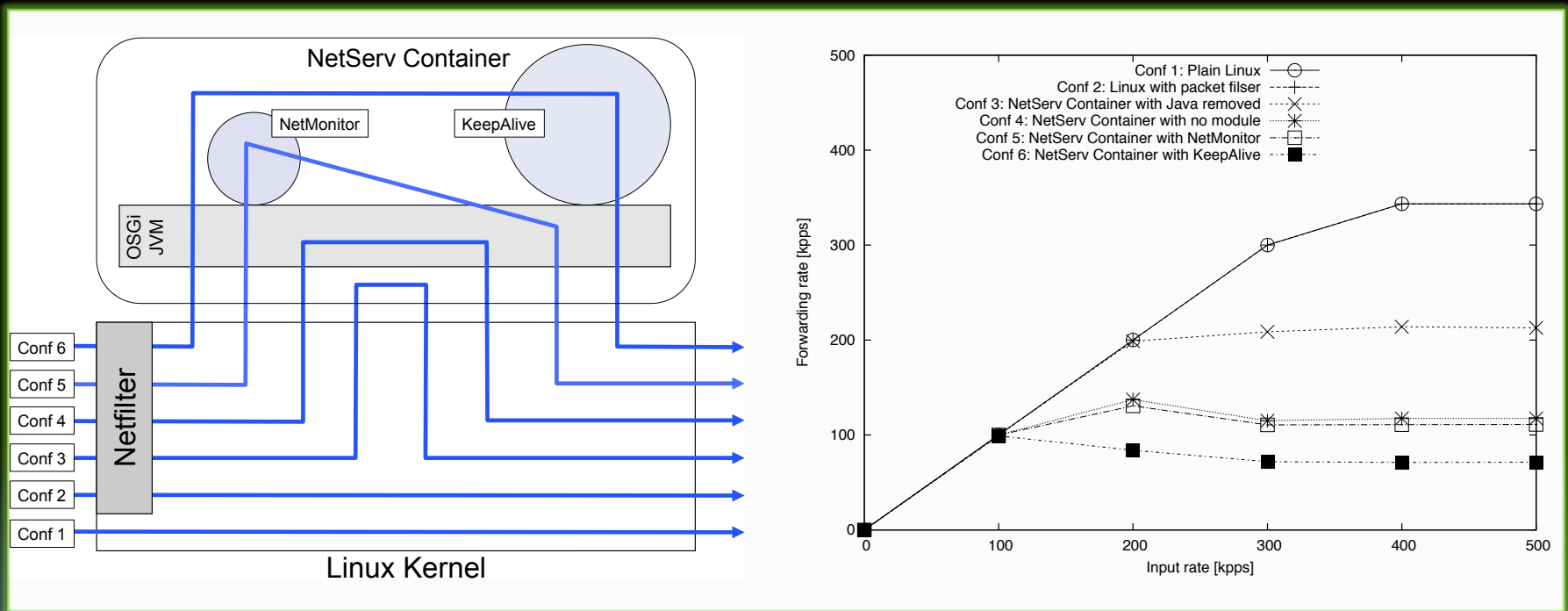
- Controlled by SIP server
- Throttles incoming traffic
- Randomized traffic rejection

SIP Server

- Installs NetServ module on demand
- Controls all NetServ modules
- Real-time feedback to modules

Evaluation

Java packet processing overhead:



- Overhead significant, but not prohibitive
- Handles typical edge router traffic on modest PC hardware

So what's new?

Three design goals:

1. Wide-area deployment
 - Signaling for deployment w/o precise target
2. Multi-user execution environment
 - Packet processing in user space
 - Multiple layers of resource control & isolation
 - Module authentication & filter authorization
3. Clear economic benefit
 - Compelling use cases

Future work

- OpenFlow integration
 - NetServ node as a *side-car*
- NetServ on commercial router
 - JUNOS SDK
- Internet multicast using NetServ
 - Hybrid of ALM and IP multicast
- Authentication and authorization using RPKI

NetServ project roadmap

- Nov 2010: Demo at GECg plenary session
 - Video available at <http://vimeo.com/16474575>
- June 2011: Presentation at FutureNet IV
- Full conference paper in the works
 - Current status available in tech report: <http://www.cs.columbia.edu/~jae/papers/netserv-tech-report-1.0.pdf>
- July 2011: NetServ tutorial at GEC11
 - NetServ as GENI Lite
- Open-source release shortly afterwards