

SPEEDUP Workshop: Programmable Networks

Richard Gold
Dr. Georg Carle

gold@fokus.gmd.de

<http://www.fokus.gmd.de/usr/richard.gold/>

<http://www.fokus.gmd.de/glone/>

**GMD FOKUS - Competence Center GloNe
and Technical University Berlin**



Overview

- **Past**
 - BANG Project: collaboration with Hitachi Japan
 - Programmable Router (GR2000)
 - Mobile Agent Platform
- **Present**
 - Policy-based Routing
 - Making intelligent routing decisions
- **Future**
 - Overlay Networks
 - Self-deploying Networks

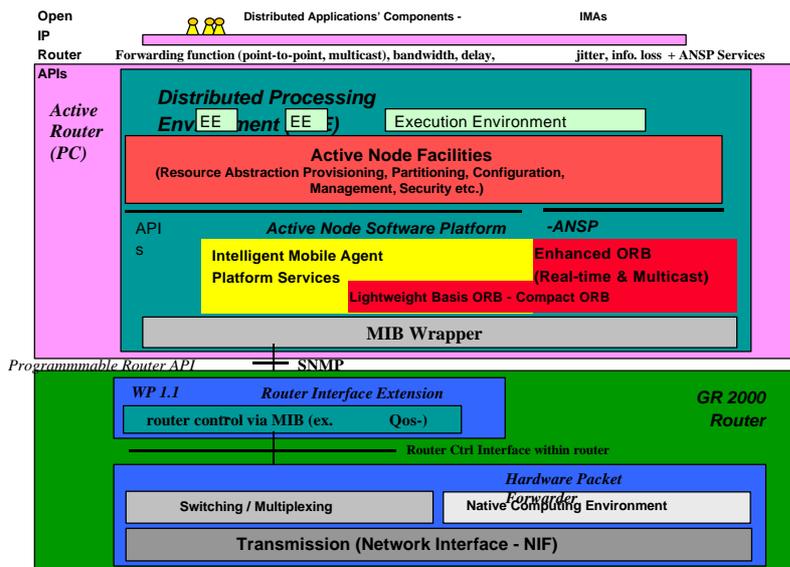


FOKUS/Hitachi Key Technologies

- Advanced Active Network Platform
 - Intelligent Mobile Agent Platform
 - Enhanced ORB Platform with Resource Control Framework (RCF)
 - Enhanced Network Support: Active QoS Control Modules (DiffServ, Meter access, Multicast Mediation)
- Applications
 - Video streaming (partial 'hard' reservation plus filtering)
 - VPN (policy, QoS, metering)
 - Multimedia Conferencing (QoS, multicast)
 - VoIP: association of call-setup signaling, QoS [DiffServ, adaptivity], multicast, metering
 - End-to-End QoS using IntServ and DiffServ
 - Active Firewalling

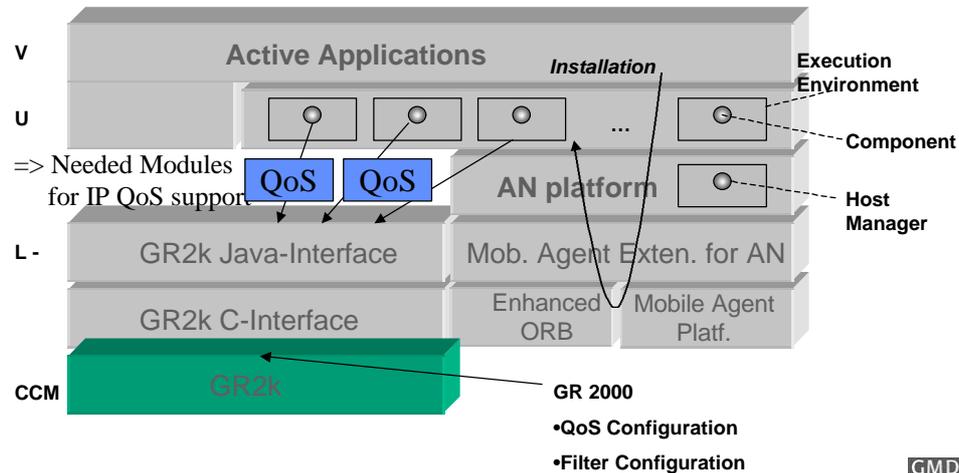


Active Node Architecture



Active Node Architecture

P1520
Interfaces



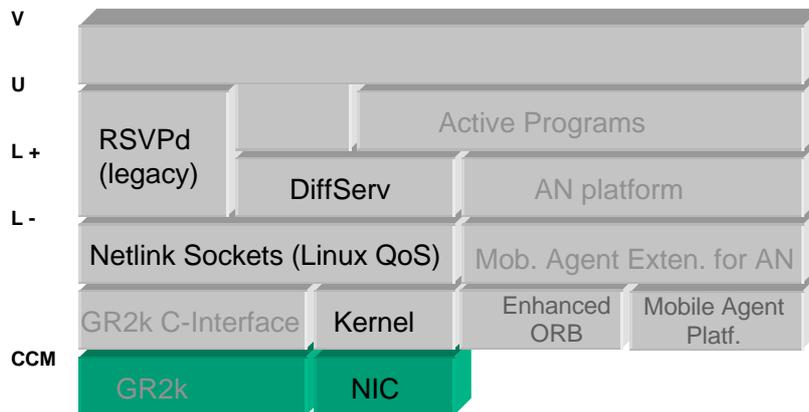
MA Active Node Architecture

- **AN Platform**
 - active host software
 - code download
 - runs the host manager
- **Components**
 - small programs managing the GR2000
 - installed by mobile agents
- **Host Manager**
 - manages the installation, execution and de-installation of components
- **Execution Environments**
 - Sandbox for components



Active Node Architecture

Extended Active Networking Platform:
Needed Modules for Active IP QoS Control



Programmability

- Active code programs using standardized interfaces (additionally to direct access to the router via the GR2k Java/C interfaces)
- Interface modules:
 - Multicast: Membership management (IETF IDMR WG)
Mediation (IETF MALLOC WG)
 - QoS: RSVP (RSVP-API, Traffic Control Interface),
DiffServ functionality
(Traffic Control Interface, Bandwidth Brokers)
 - Metering: Access to IP Meter for active applications



BANG Key Features

- Value-Added IP Active Network Nodes (Active Router, Active Communication Server)
- Value-Added IP Service Creation: high-level services (IP Telephony, VPN) require association of several basic IP services
- Three level architecture (active [U], programmable [L+], fixed part [L-]) reflects performance vs. flexibility tradeoff
- Code distribution: Mobile Agent and DPE platforms
- Distinction between node-local and network-wide functionality => Extended network view for active application development and deployment
- Transition to a programmable/active network: partial AN deployment with full end-to-end QoS support
- Architecture supports legacy software architectures: RSVP implementations, DiffServ on Linux implementation



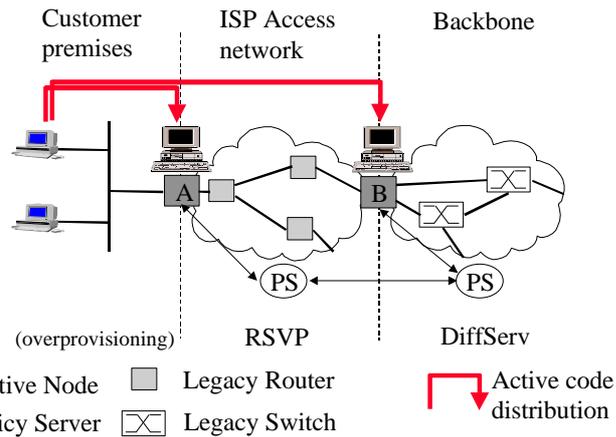
Service Creation Examples

- **Video on Demand** (partial 'hard' reservation plus filtering)
- **VPN** (policy, QoS [reservation], metering)
- **Multimedia Conferencing** (QoS [reservation], multicast)
- **VoIP / IP Telephony**: association of call-setup signaling, QoS [DiffServ, adaptivity], multicast, metering
- **End-to-End QoS** using IntServ and DiffServ
- **Active Firewalling**: Fast reactions against attacks

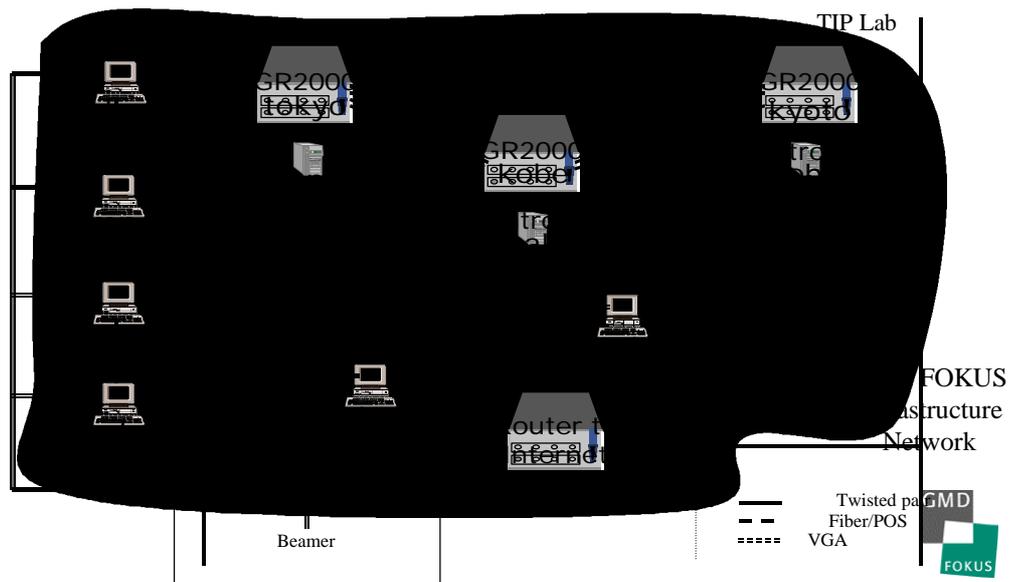


Active Applications: Service Creation

- Creation of high-level services (Telephony) require association of several basic IP services
- Significant performance gains can be achieved when basic IP services are employed in a *coordinated* way



FOKUS/Hitachi Testbed at GMD Fokus “Active Internet Lab”



Present: Policy-based Routing

- Detour Routing + Peterson's work on end-to-end media streams
- Use Active Nets to deploy intelligence onto edge devices
- Make decisions based on metrics/policies concerning how to route a flow through the network
- Multiple metrics/policies allow for increased intelligence in the routing decisions that we can take
- Programmable Networks provide substrate for introducing informed routing decisions into the network



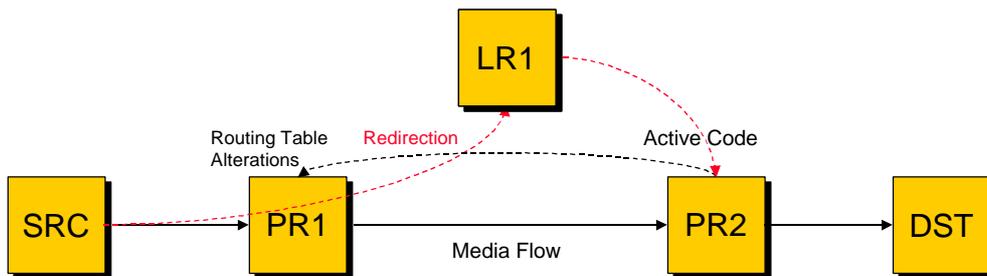
Present: Policy-based Routing

- **Current Node Architecture**
 - ANTS
 - Active Network Execution Environment
 - Openet
 - Programmable Router interface
 - ABONE
 - Provides virtual topology and packet demultiplexing
 - Linux v2.4 Kernel & GR2000
 - Provide flexible routing infrastructure



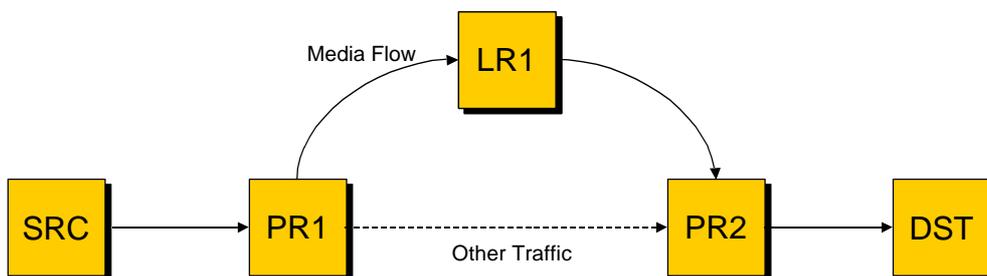
Application scenario

- Re-routing of non-time sensitive application flows



Application Scenario

- Re-routing of non-time sensitive application flows



Future: Overlay Networks

- Decentralized systems like Peer-to-Peer applications (Gnutella, FreeNet etc.) have scalability problems
- Lack of fixed infrastructure means that infrastructure must be created by the nodes of the **overlay network**
- Provide **BGP**-like hierarchies and route aggregation for Ad-Hoc Overlay Networks
- System also **self-organizing**: does not require static **AS** definitions, groups are dynamically created at runtime depending on **metrics** (delay, hops etc.)
- Scales **up** and **down** according to number of nodes in the system
- **Related work**: SOAR at UCL, Rendezvous at Washington Uni



Future: Overlay Networks

- Proposal: Programmable Networks allow us to make intelligent routing decisions
- Problem with tunnelled overlay networks is sub-optimal routing due to routing decision being based on the encapsulating packet header
- With Programmable Networks we can make routing decisions based upon the tunnelled packet header
- Application-awareness in the network is thus easy to deploy as it just consists of injecting new code onto selective active devices in the network



Conclusions

- Programmable Networks consist of intelligent IP router and co-located Active Network platform
- Combines the **flexibility** of Active Networks with the **high-performance** of a hardware-based router
- Past application: Active Loss Concealment (**IWAN2000** paper)
- Currently we are building a policy-based routing system based upon Programmable Networks
- One future direction is the usage of Programmable Networks to steer Overlay Networks (**OpenArch'01** paper)

