Institut für Kommunikations-Technik

MIDAS: Middlebox Discovery and Selection for On-Path Flow Processing

NFV workshop 2015

Published on Comsnets 2015

TNOVA

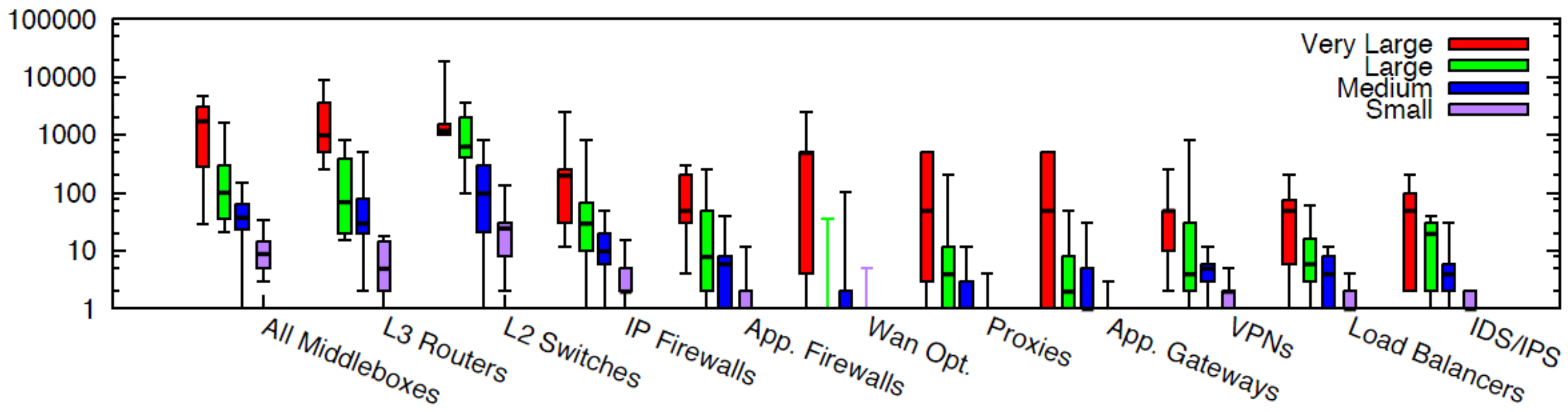
Ahmed Abujoda, **Panagiotis Papadimitriou**

Leibniz Universität Hannover, Germany

Leibniz Universität Hannover

Introduction

- Today’s enterprise relies on wide range of middleboxes :
  - packet filtering
  - proxies
  - load balancing
  - redundancy elimination
  - encryption
  - …..



J. Sherry et al., **Making Middleboxes Someone Elses Problem: Network Processing as a Cloud Service**, SIGCOMM 2012

Introduction

- Today's Middleboxes limitations:
  - Specialized Hardware and functionality
    - High investment cost
  - Standalone device provisioned for peak loads
    - Inefficient resource utilization
  - Diverse management and configuration interfaces
    - High operation cost
  - Deployed closed to the edge
    - Single point of failure
    - Concentrate traffic at the edge

- NFV, replaces middleboxes with software NFs on:
  - To the cloud [APLOMB] (off-path)
  - To the network (on-path)

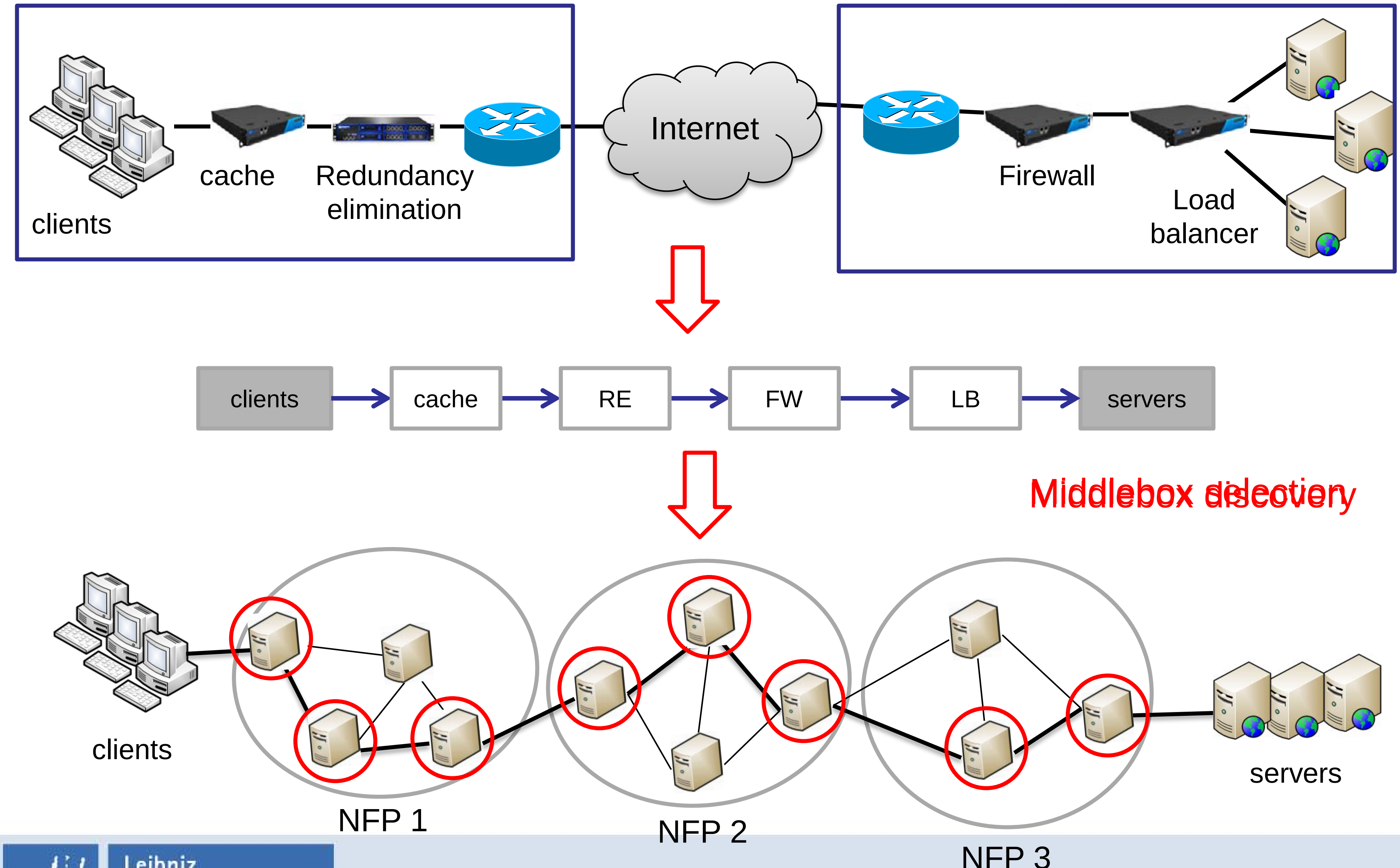
J. Sherry et al., **Making Middleboxes Someone Elses Problem: Network Processing as a Cloud Service**, SIGCOMM 2012

Introduction

- Migration of middleboxes to the network:
  - Enable NFaaS
    - Pay-per-use model
    - Reduced CAPEX/OPEX and high flexibility and scalability
  - Empowering the “middle”
  - Bandwidth conservation
    - Redundancy elimination
    - Packet filtering for DoS mitigation
- Recent trends
  - Routers with programmable processors
  - Packet processing on commodity servers
    - Consolidated SW middleboxes [Flowstream, CoMB]
  - Micro-datacenter deployment by ISPs

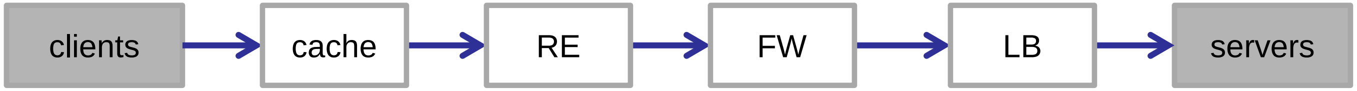
A. Greenhalgh et al, **Flow Processing and the Rise of Commodity Network Hardware**, CCR 2009
V. Sekar et al., **The Design and Implementation of a Consolidated Middlebox Architecture**, NSDI 2012

On-Path Flow Processing



Network Processing Requirements

- Performance
  - High packet forwarding rates [RouteBricks, ClickOS]
  - Low processing setup delay
- Resource utilization efficiency
  - Load balancing
- Correctness
  - Network functions (NFs) should be embedded in the correct order
- Proximity
  - Some NFs should be closed to the source/destination



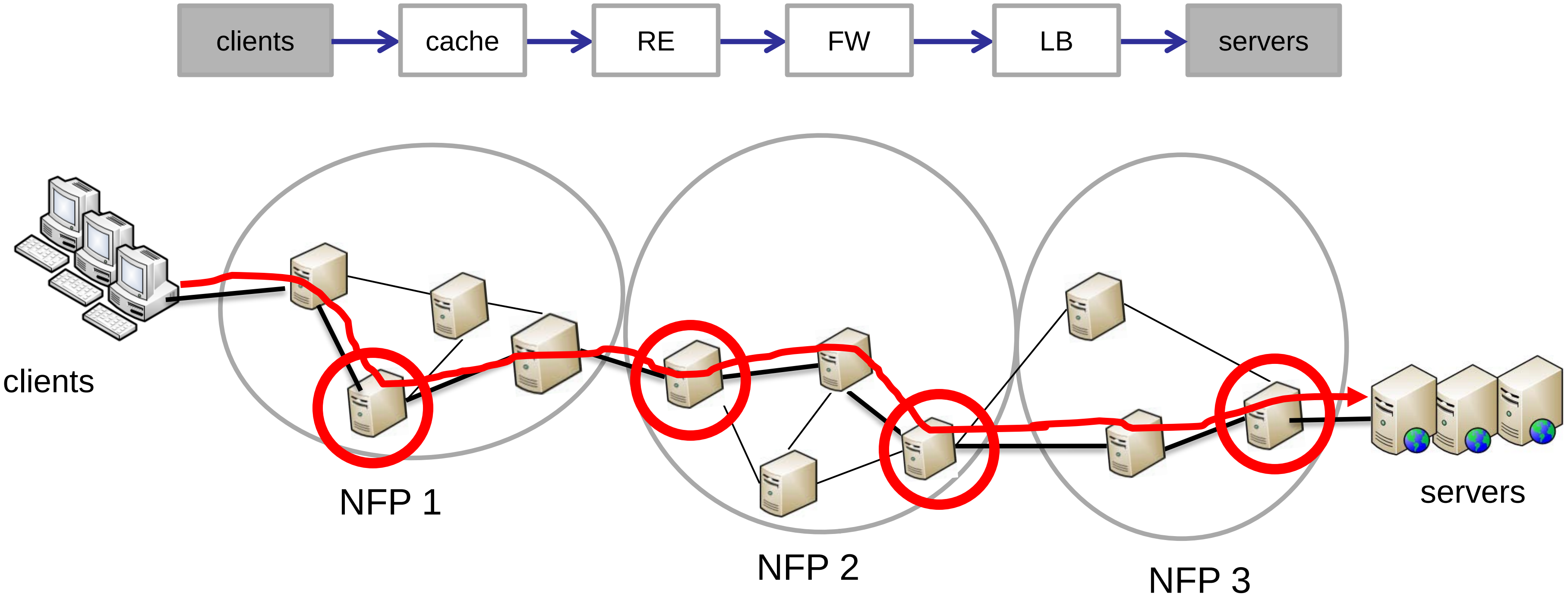
M. Dobrescu et al., **RouteBricks: Exploiting Parallelism to Scale Software Routers**, SOSP 2009
J. Martins et al., **ClickOS and the Art of Network Function Virtualization**, NSDI 2014

Middlebox Discovery and Selection

- Middlebox discovery
  - Path discovery and middlebox detection techniques (e.g., traceroute, tracebox) incur high delays
  - Signaling protocols (e.g., SIMCO) are designed for middlebox configuration

- Middlebox selection
  - NF location dependencies require large provider footprint (i.e., multiple NFPs)
    - NFP resource information disclosure policies

On-the-fly Processing Setup



- Middleboxes pick up flows as they arrive
  - ✓ Performance
  - ✘ Trade-off between correctness and efficiency
- Need for processing setup coordination within and across NFPs

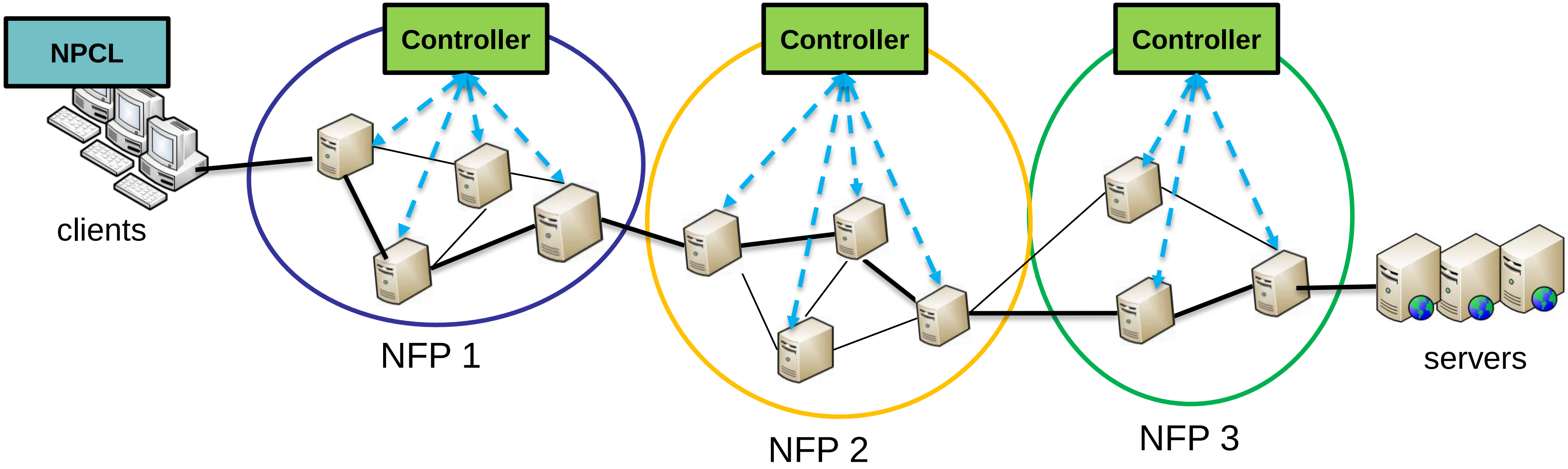
Outline

- MIDAS overview
- Middlebox discovery
  - Middlebox signaling
  - Middlebox controller chaining
- Middlebox selection
  - Privacy-preserving NFP assignment
  - Intra-provider middlebox selection
- Implementation
- Evaluation
- Conclusions

MIDAS Overview

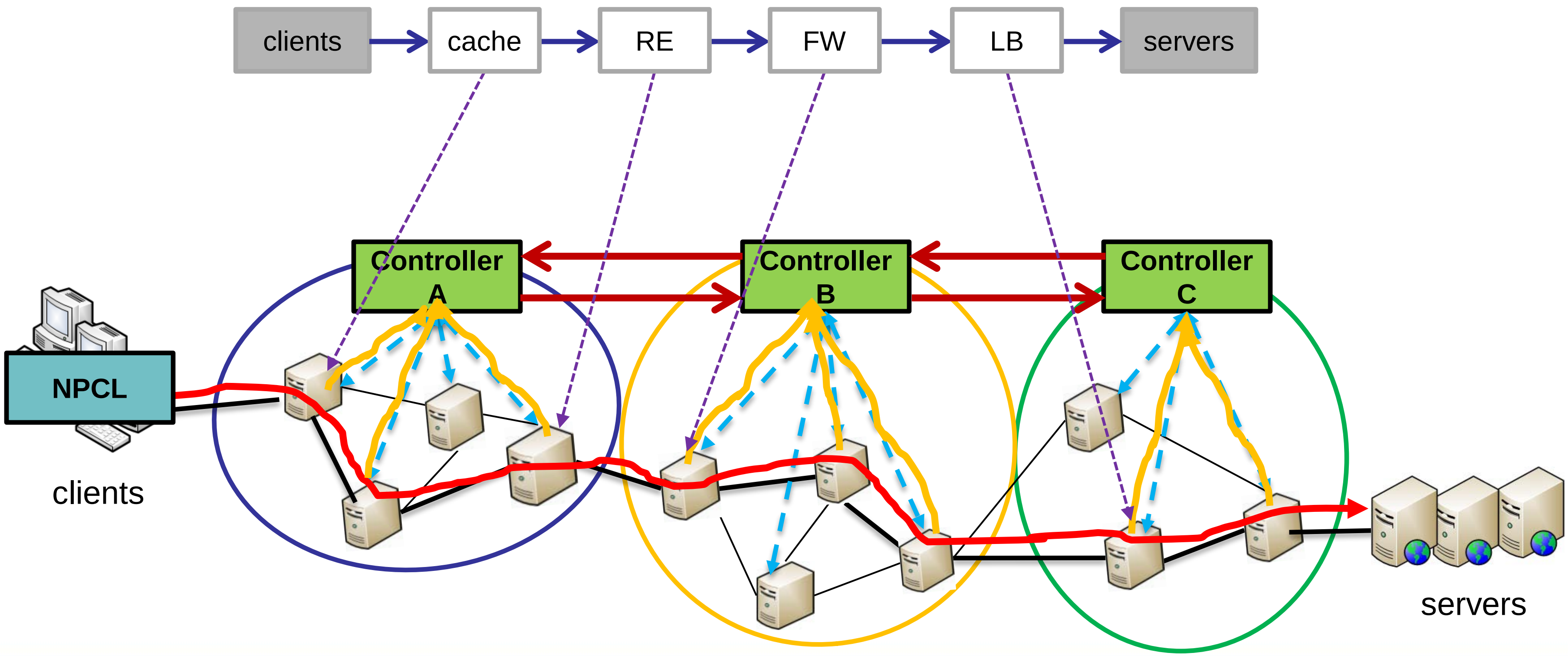
MIDAS Components

- Main components:
  - Consolidated middlebox (CoMB)
  - Centralized CoMB controller in each NFP
  - Network processing client (NPCL)



MIDAS Approach

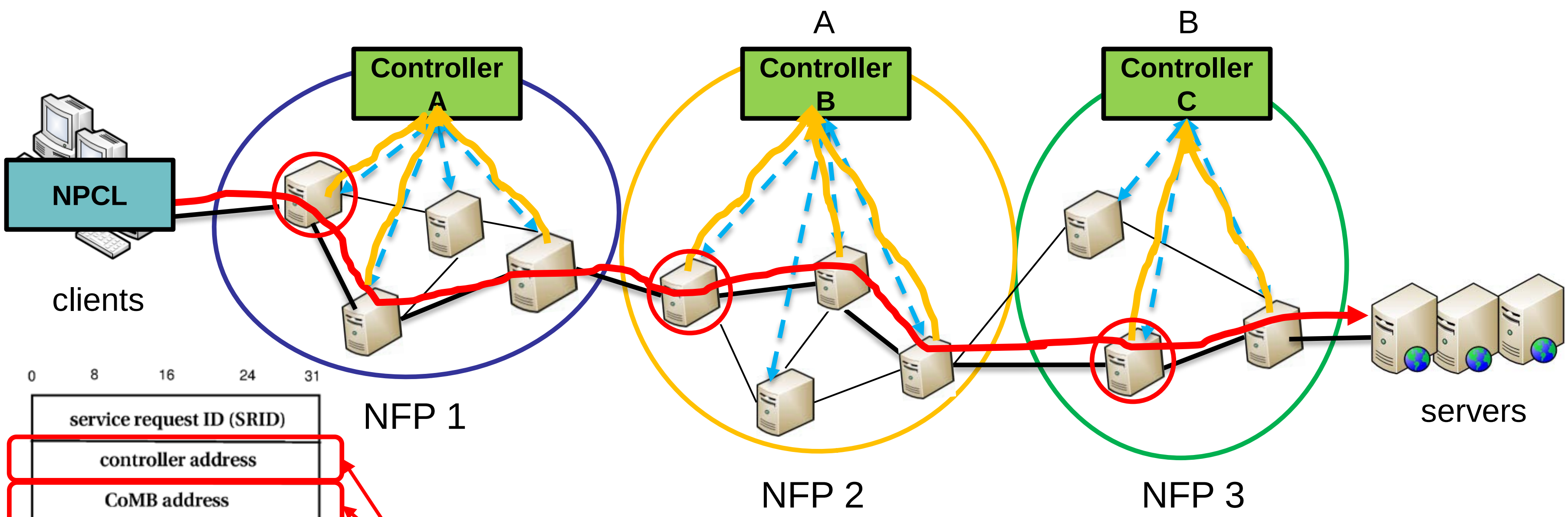
Middlebox Signaling → Controller Notification → Controller Chaining → Middlebox Selection → NF Setup

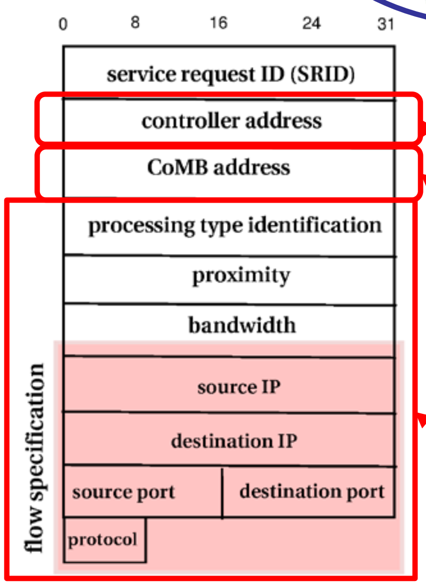


Middlebox Discovery

Middlebox Signaling

Institut für Kommunikations-Technik





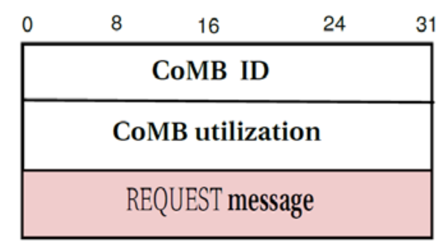
REQUEST message

Extracted by each ingress CoMB and inserted into the DISCOVERY message

Each CoMB inserts its IP address

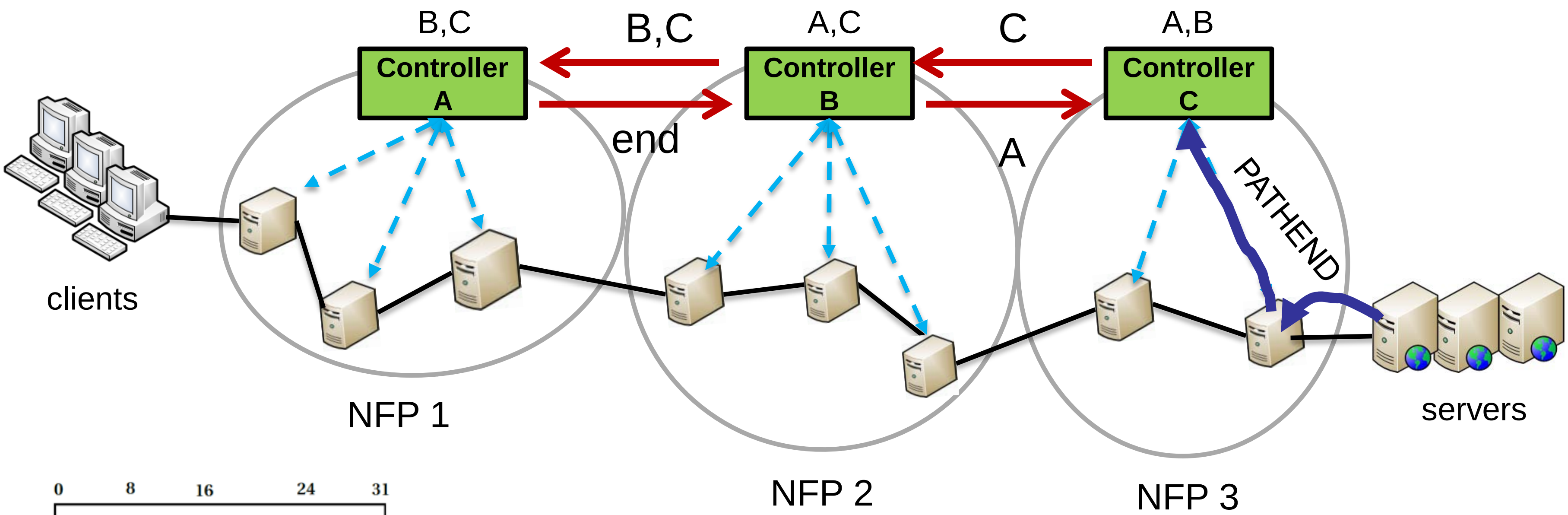
- Downstream controllers know about upstream ones
- Middleboxes identified
- Processing request spec. delivered

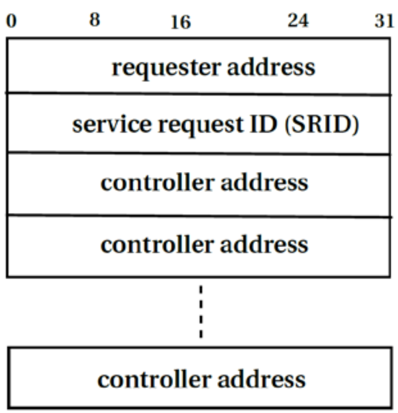
NFs chain specifications



DISCOVERY message

Controller Chaining





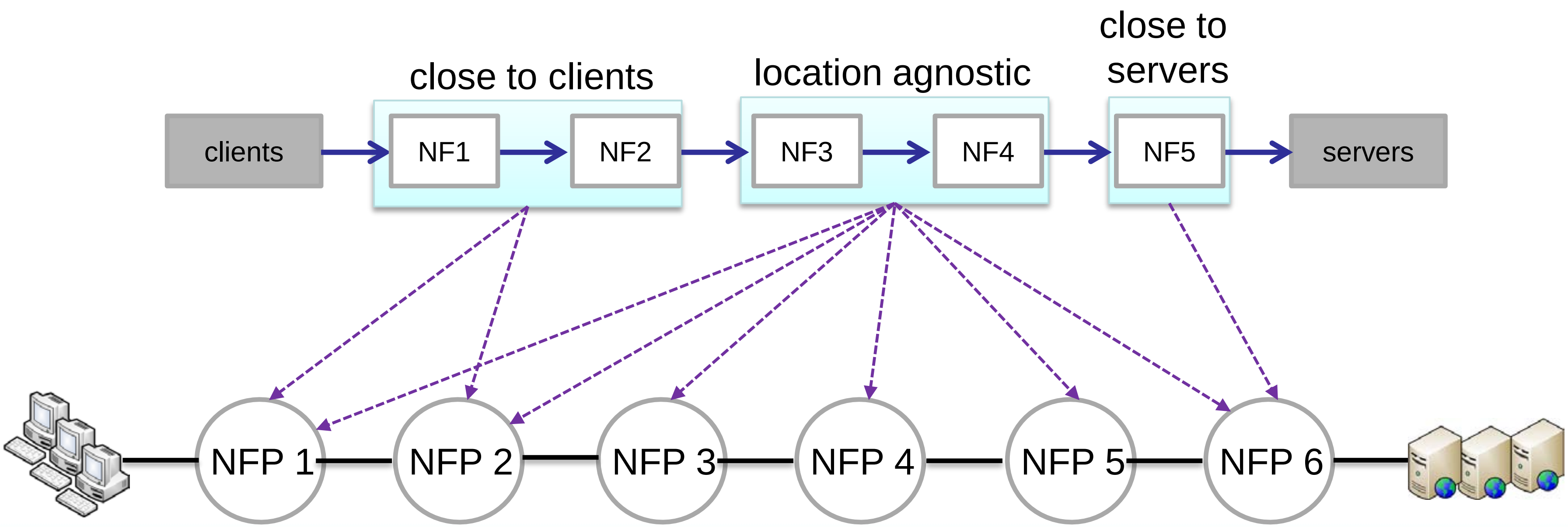
CONTROLLER message

Middlebox Selection

NFP Assignment

Institut für Kommunikations-Technik

- Objective:
  - Minimize number of assigned NFPs
  - Maintain providers privacy
- Distributed approach:
  - Each NFP partitions the service chain based on NF location dependencies
  - Each NFP announces which segment it is willing to host
  - Privacy-preserving protocol (using MPC) to assign chain segments to the NFP with the lowest utilization

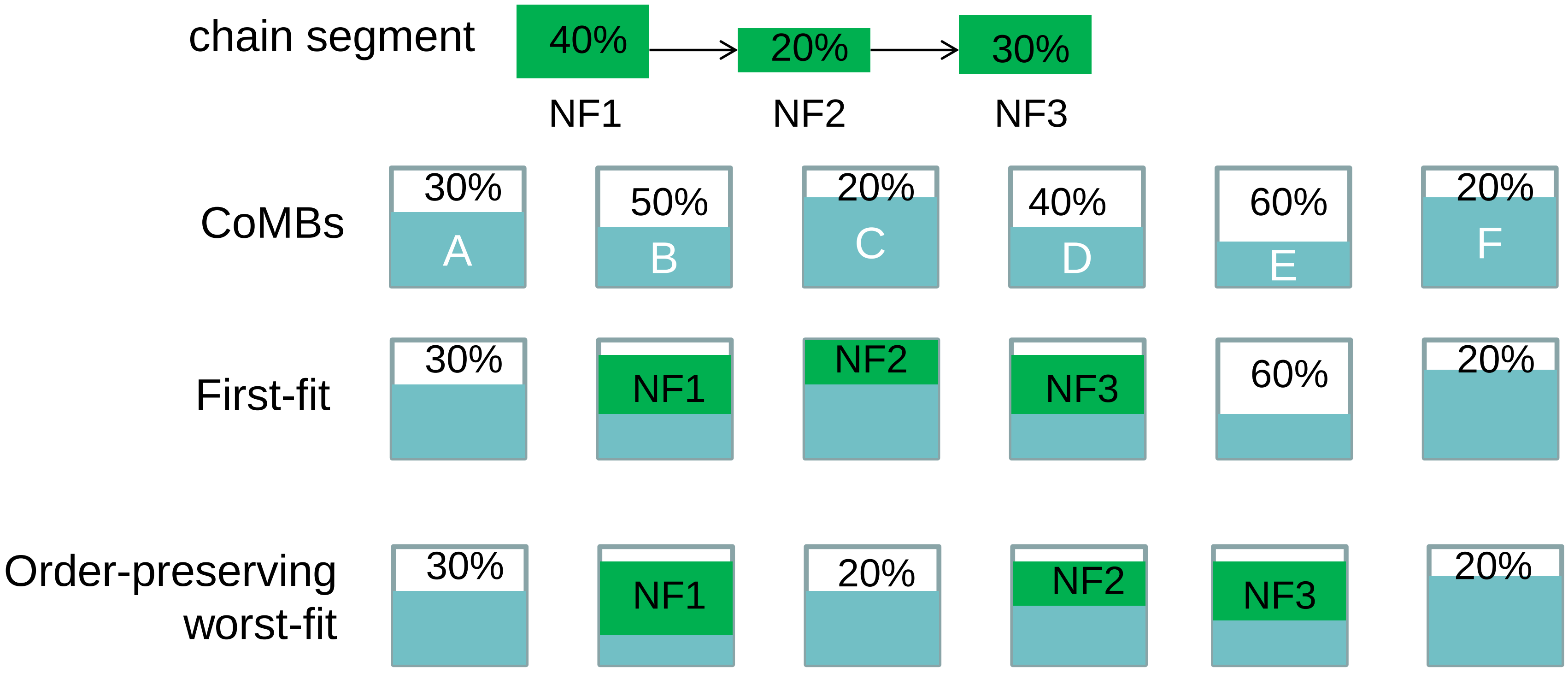


Multi-party Computation (MPC)

- Cryptographic protocol :
  - Different parties with private inputs to compute a function on their inputs:
    1. Input values stays private
       - Utilization of each NFP
    2. Result of the computation is correct
       - Compute the NFP with the lowest utilization
    3. Cheating parties won’t learn information about the honest parties inputs

Intra-Provider Middlebox Selection

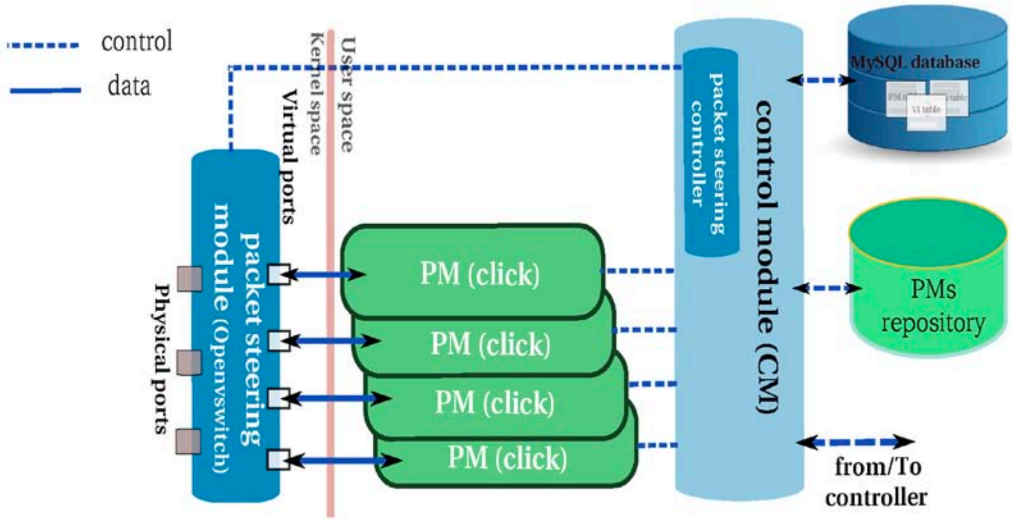
- Objectives:
  - Load balancing
  - Correctness
- Approach:
  - Step 1: Order-preserving First-fit
  - Step 2: Order-preserving worst-fit



Implementation

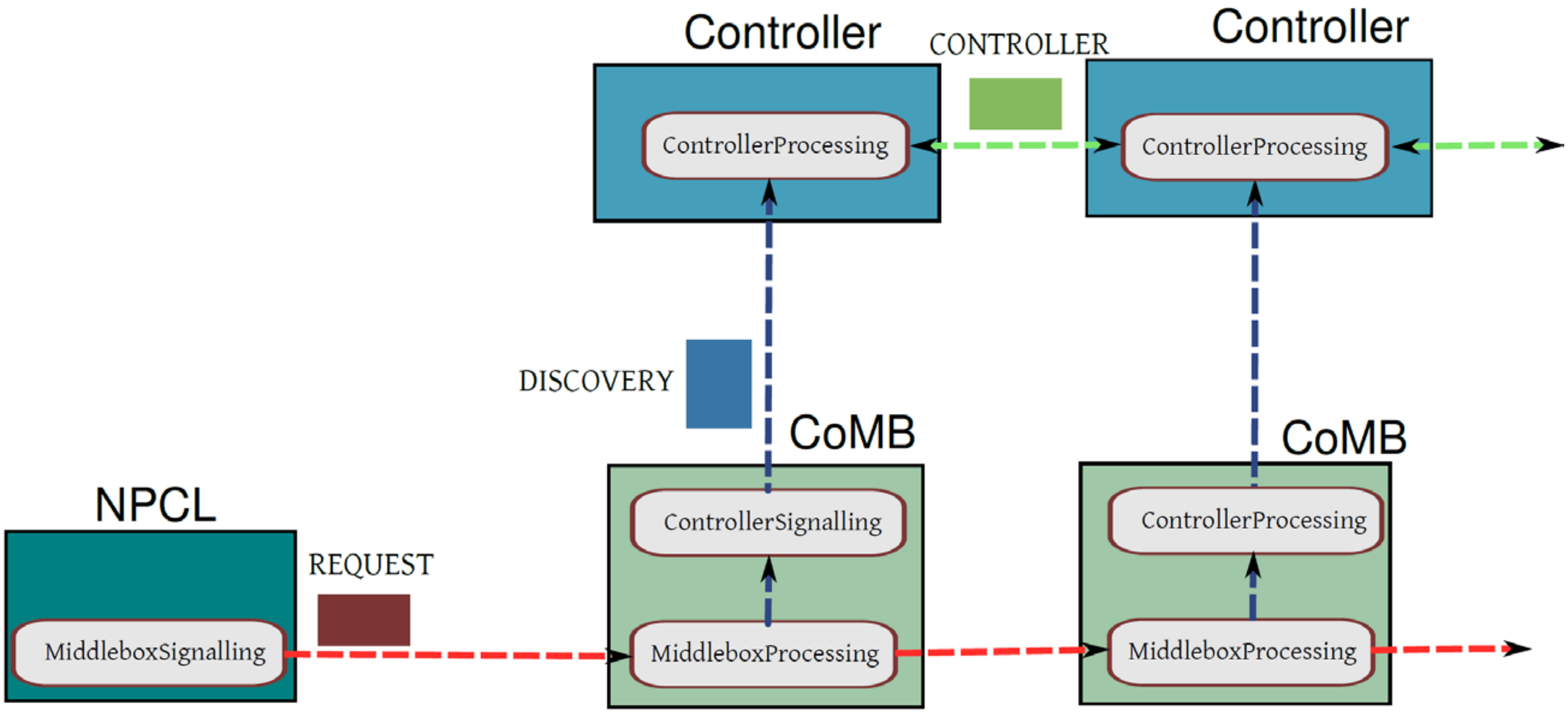
CoMB Implementation

- Processing module (PM):
  - Implements NFs using Click Modular Router
- Packet steering module:
  - Steers traffic between PMs and physical ports using OpenvSwitch
- Repository:
  - Stores PM configuration templates
- Control module:
  - Installs, configures, and terminates PMs
  - API exposed to controller



Discovery implementation

- Implemented 4 Click modular router elements
- Encapsulate messages in UDP packets
- Identify signaling message by the IP Router Alert Option (RAO)



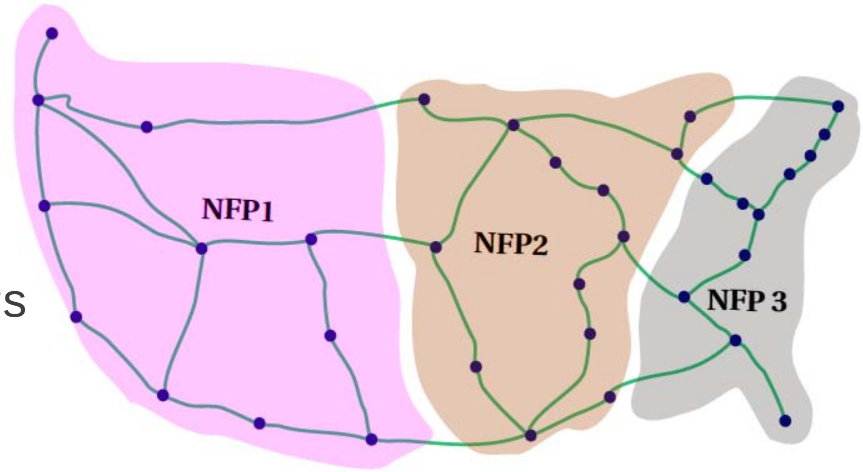
Evaluation

Evaluation Environment

Institut für Kommunikations-Technik

- Experimental evaluation of flow processing setup delay:
  - 22 servers deployed in an Emulab-based testbed (FILAB):
    - quad-core Xeon CPUs @2.27GHz and 6 GB DDR3
  - 2 - 5 NFPs, each with:
    - 1 controller
    - 3 CoMBs (deployed in separate nodes)

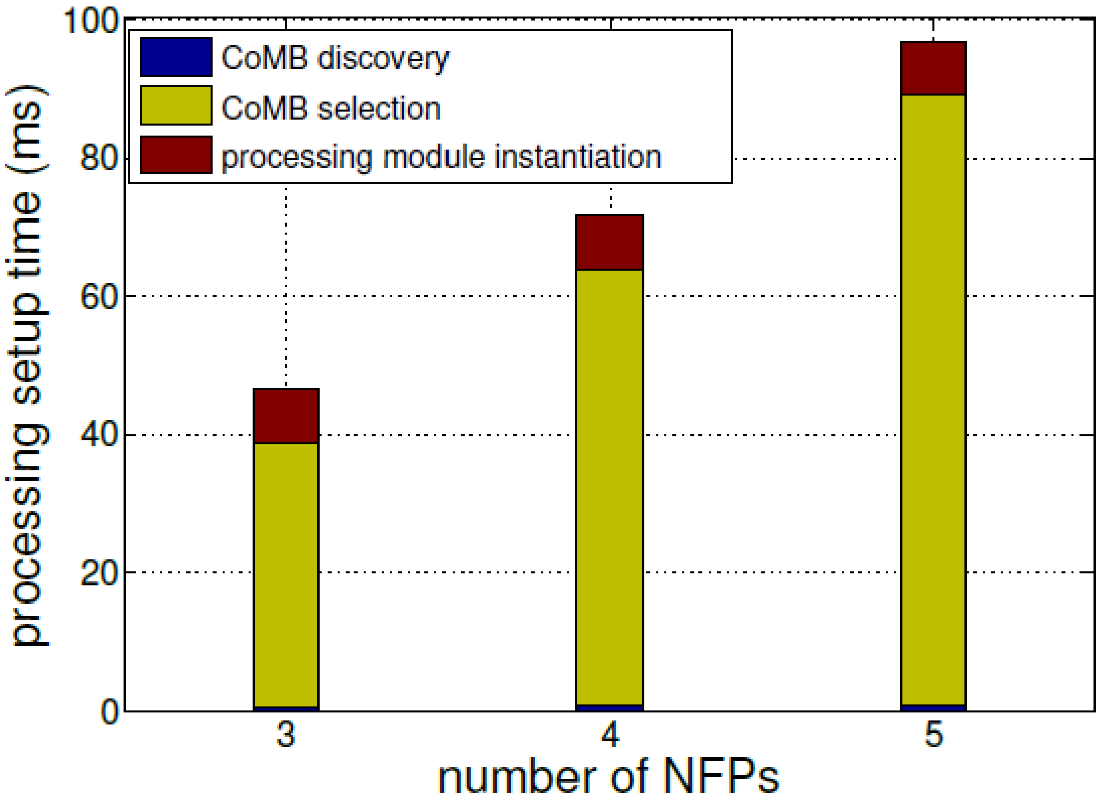
- Evaluation of ComB selection efficiency with simulations:
  - Simulator:
    - Flow-level simulator (Python)
  - Simulation setup:
    - Internet-2 topology
    - 34 CoMBs subdivided into 3 NFPs



Flow Processing Setup

Institut für Kommunikations-Technik

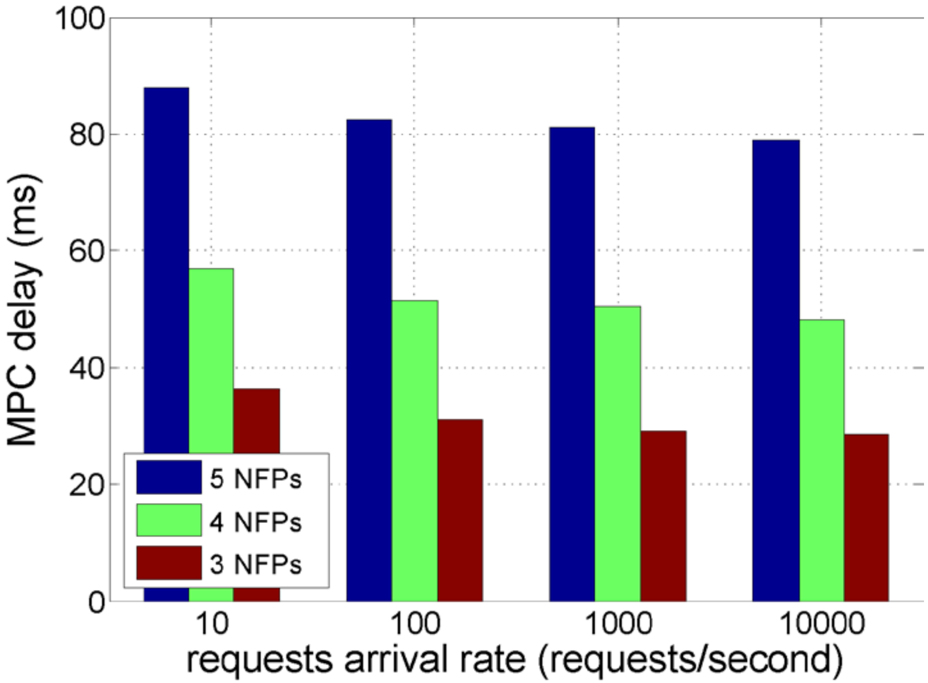
- Flow setup delay < 100 ms
- CoMB selection dominates flow processing setup delay
  - MPC is computationally-intensive (O(n^2), n is the number of NFPs)



processing delay per request (10 requests/sec)

MPC selection

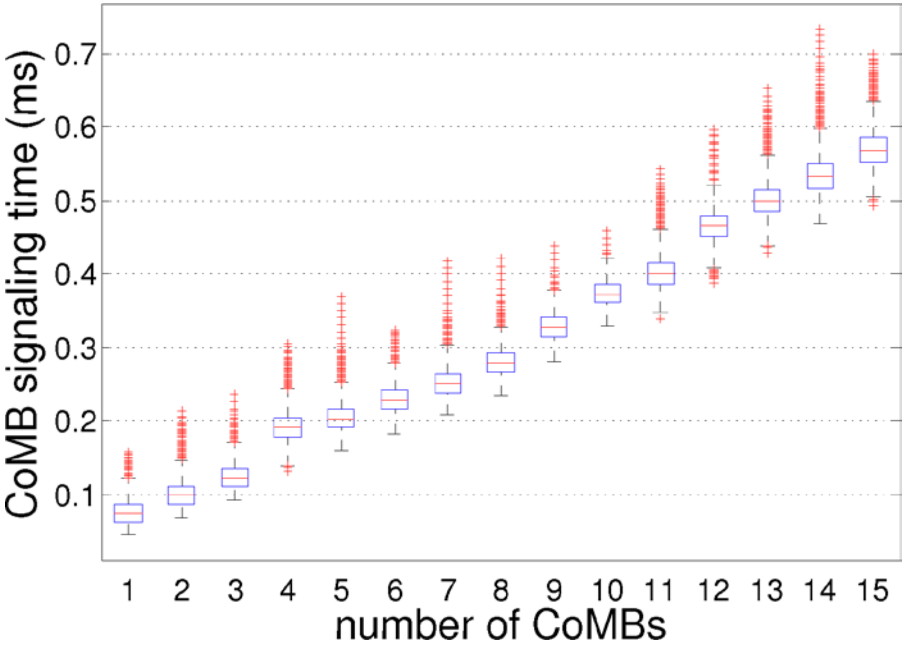
- MPC delay < 100 ms for up to 5 NFPs (i.e., average AS-path length)
  - MPC can be scaled with GPU



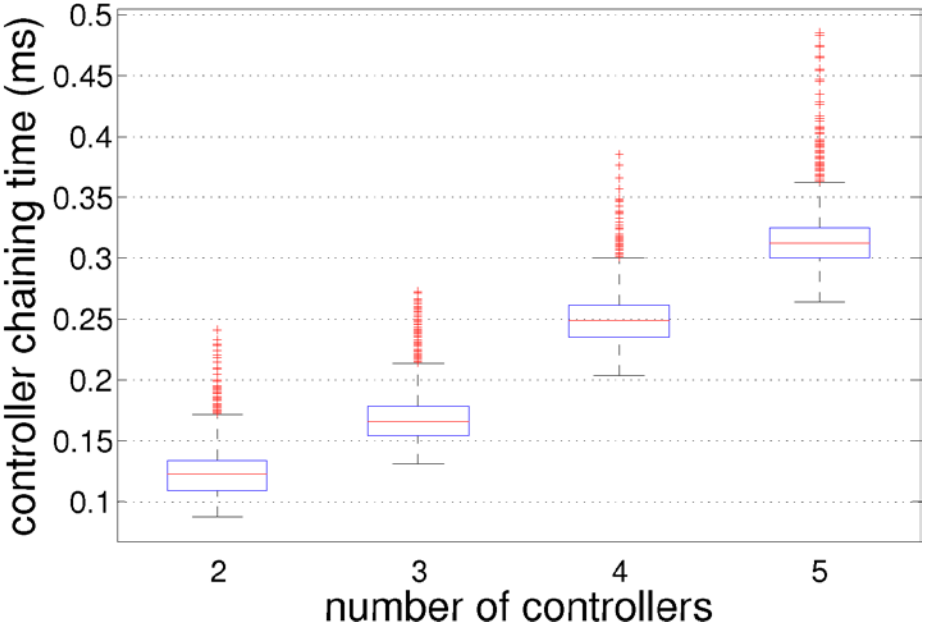
MPC delay per request

Middlebox Discovery

- Minimal delay with CoMB signaling and controller chaining
- Middlebox discovery scales with the number of CoMBs and controllers



CoMB signaling delay per request

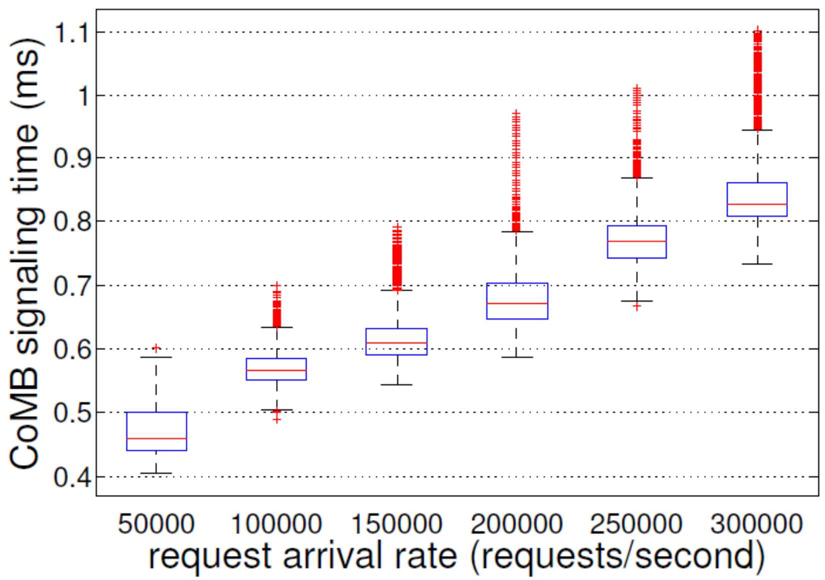


controller chaining delay per request

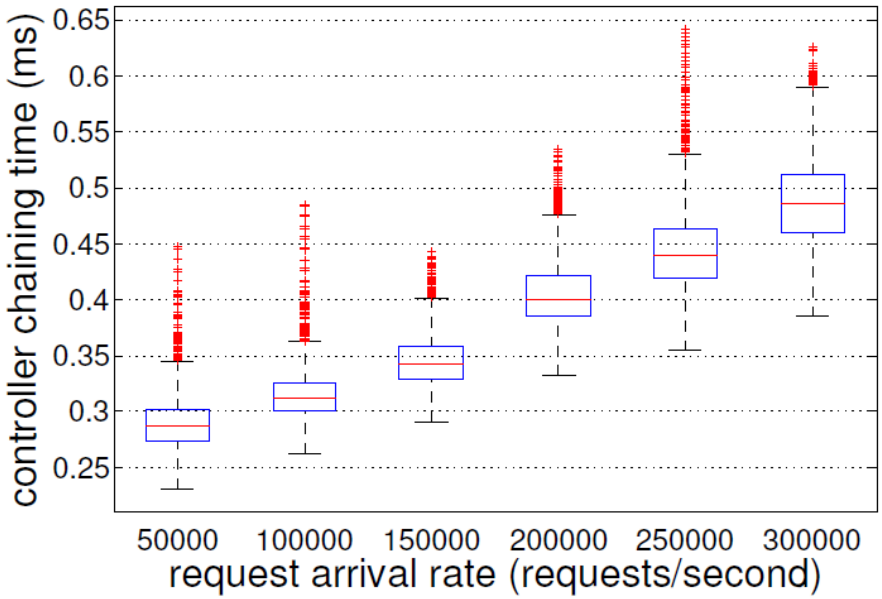
Middlebox Discovery

Institut für Kommunikations-Technik

- Middlebox discovery scales with the number of requests (300K requests/second)



CoMB signaling delay per request

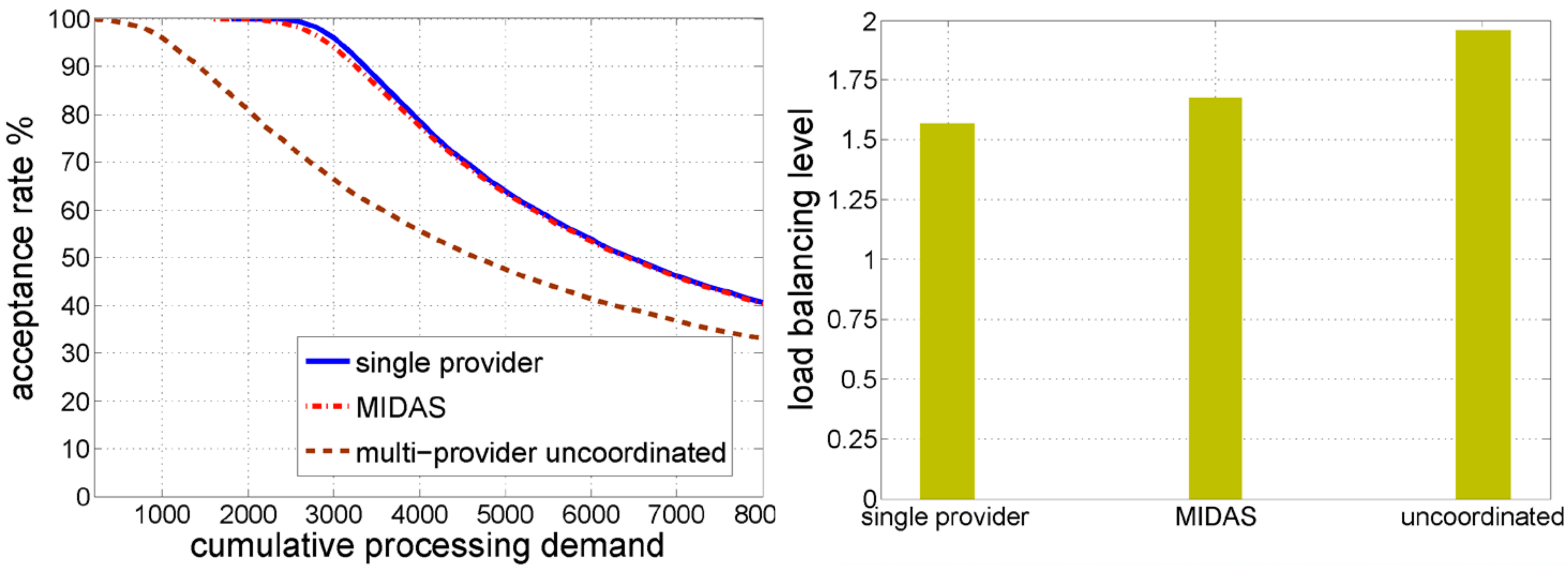


controller chaining delay per request

Simulation Results

Institut für Kommunikations-Technik

- Comparison method:
  - Single provider:
    - All CoMBs managed by a single controller
  - Multi-provider uncoordinated:
    - On-the-fly selection of CoMBs based on the utilization level



Conclusions

Conclusions

- MIDAS enables:
  - Middlebox discovery without prior knowledge of the traffic path
  - Privacy-preserving Interoperability among NFPs for middlebox selection
  - Rapid and order-preserving network service embedding
  - Feasibility of coordinated on-path processing setup

Thank you!

Ahmed Abujoda

E-mail: ahmed.abujoda@ikt.uni-hannover.de
WWW: http://www.ikt.uni-hannover.de/