COPS and RAP Overview

Raj Yavatkar, Intel

(on behalf of the RAP working group)

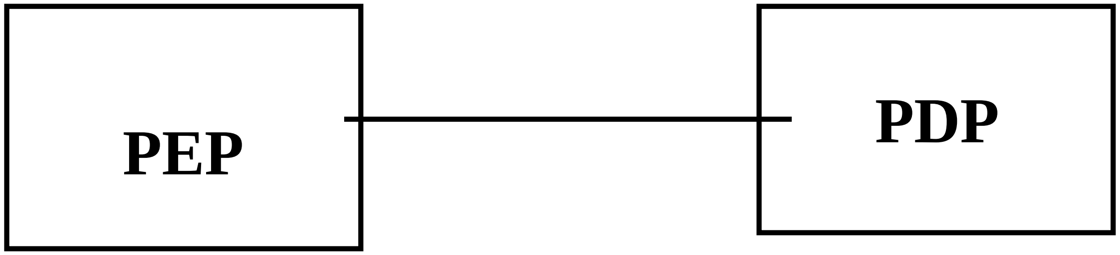
Outline

- Background on RAP working group
- COPS Overview
- Use of COPS with RSVP
- Use of COPS with Diff-Serv

Background

- RAP working group
  - Specify a framework for providing policy-based control over QoS admission control decisions
  - *focus on policy-based control over admission control using RSVP*
  - allow for policy-based admission control in other QoS contexts, whenever possible
  - support for monitoring and accounting
  - drafts
    - draft-ietf-rap-framework-01.txt
    - draft-ietf-rap-cops-02.txt, draft-ietf-rap-cops-ds-00.txt, draft-ietf-rap-user-identity-00.txt

Architectural Elements

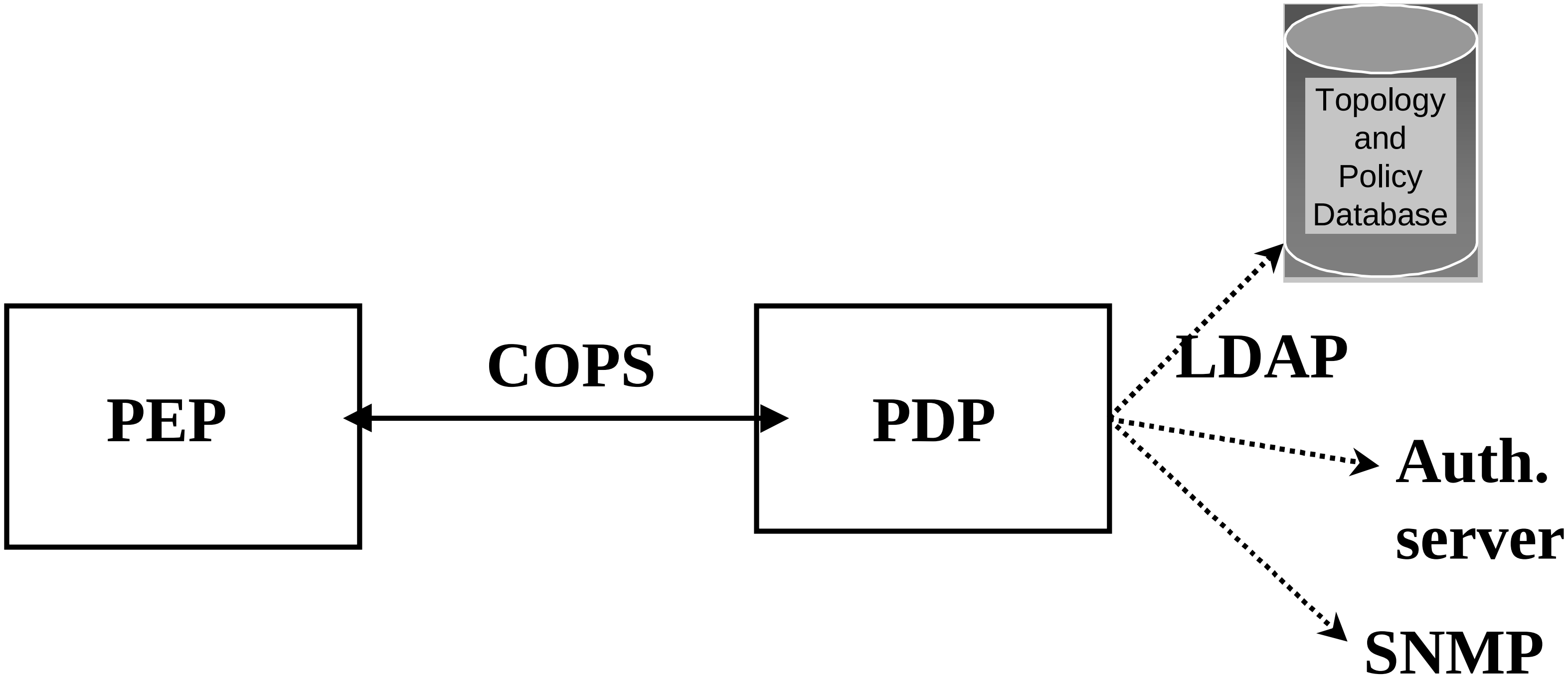


PEP -- Policy Enforcement Point; decisions are enforced here

PDP -- makes policy decisions/pushes policy configuration

Interaction between PEP, PDP

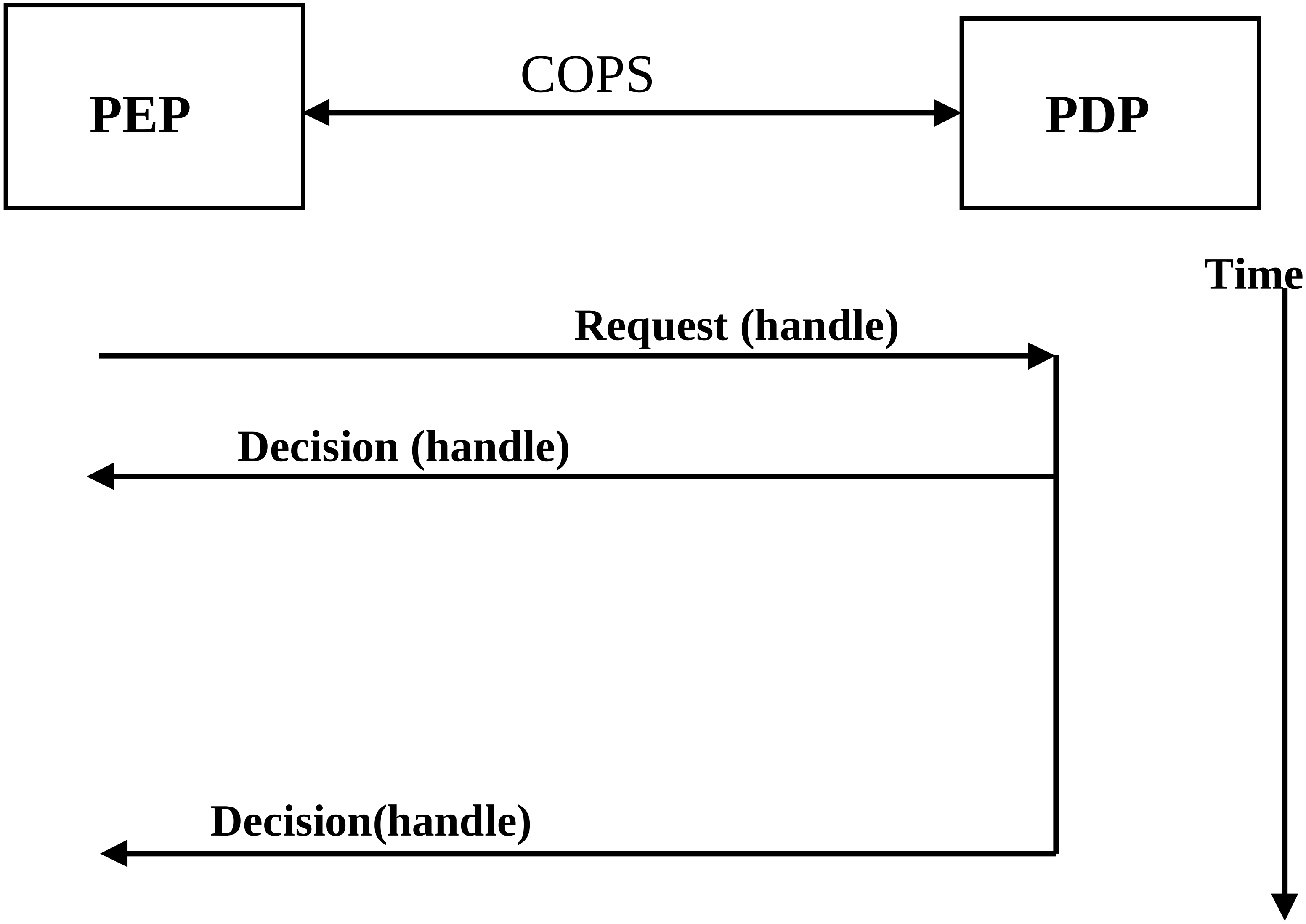
- Two types of operations performed by PEPs
  - Outsource decisions
    - When PEP requires a policy decision, PEP contacts PDP for a policy decision
    - Request contains policy elements and admission control information (e.g., flowspec).
    - PDP returns policy decision and additional info
  - Configuration requests
    - PDP configures PEP with device-specific policy information



- PDP itself may use other services/protocols such as LDAP for accessing policy database,
  an authentication server for user authentication,
  SNMP for configuration/mgmt, etc.
- PEP always runs on a policy-aware node

COPS (Common Open Policy Service)

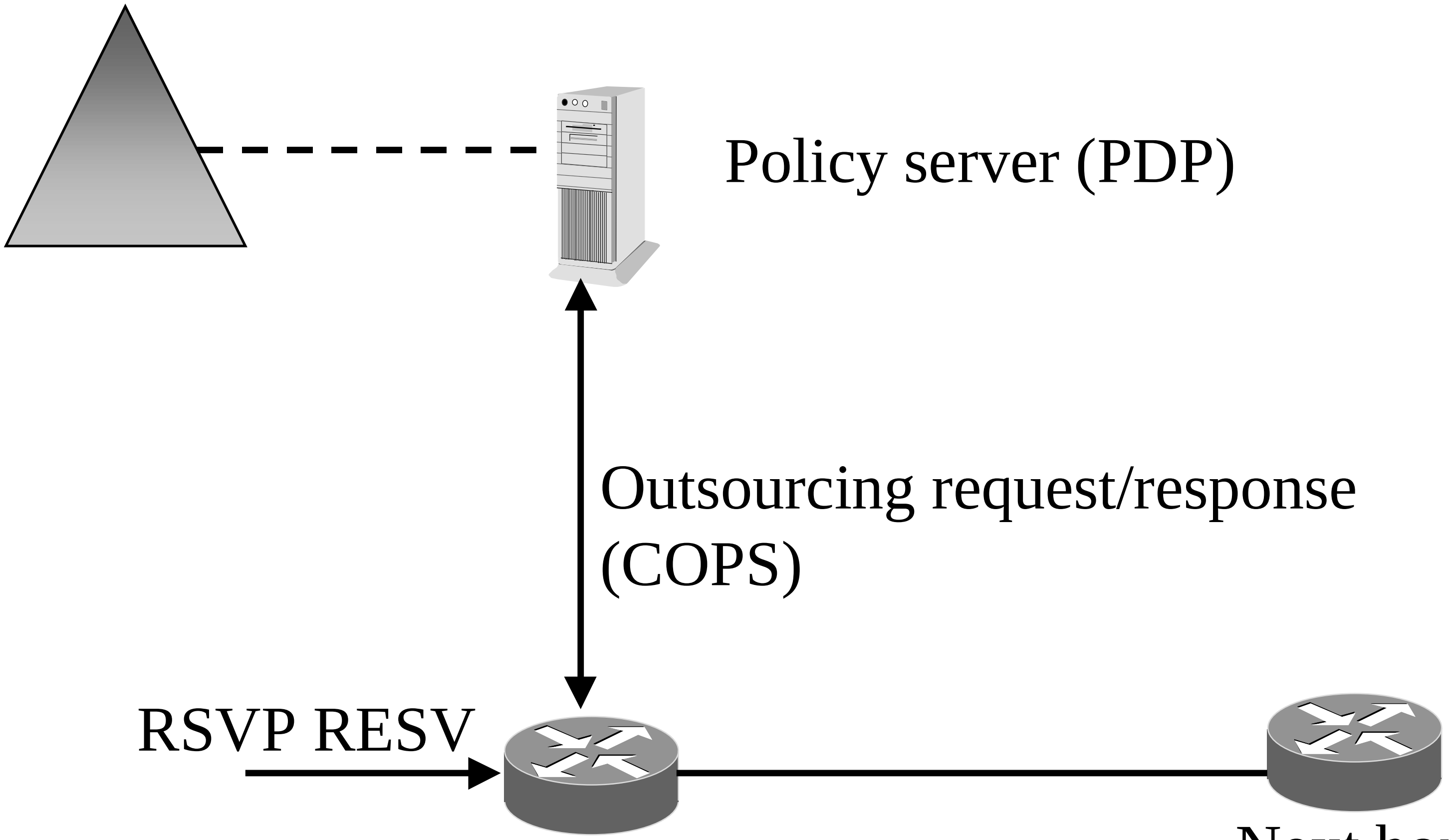
- A request-response protocol for PEP-PDP interaction
  - uses TCP for transport
  - its own Keep-Alives to detect failures
  - includes a state synchronization mechanism to handle recovery from failures, etc.
  - PDP can send an asynchronous notification to PEP when policy decision or configured information changes (e.g., preemption)
  - facilities to report status, stats, monitoring info


PEP
COPS
PDP
Time
Request (handle)
Decision (handle)
Decision(handle)

Example of a COPS Session

- PEP opens a COPS session
  - specifies ClientType
- PEP sends requests and receives responses/decisions
  - a handle associated with each request
- PDP can send *Unsolicited Decisions* any time to change previously installed state(s) at PEP
- PEP sends back *report* messages to report resource usage and accounting info
- KeepAlive messages sent when no activity

Use of COPS with RSVP/Intserv



Policy server (PDP)

Outsourcing request/response (COPS)

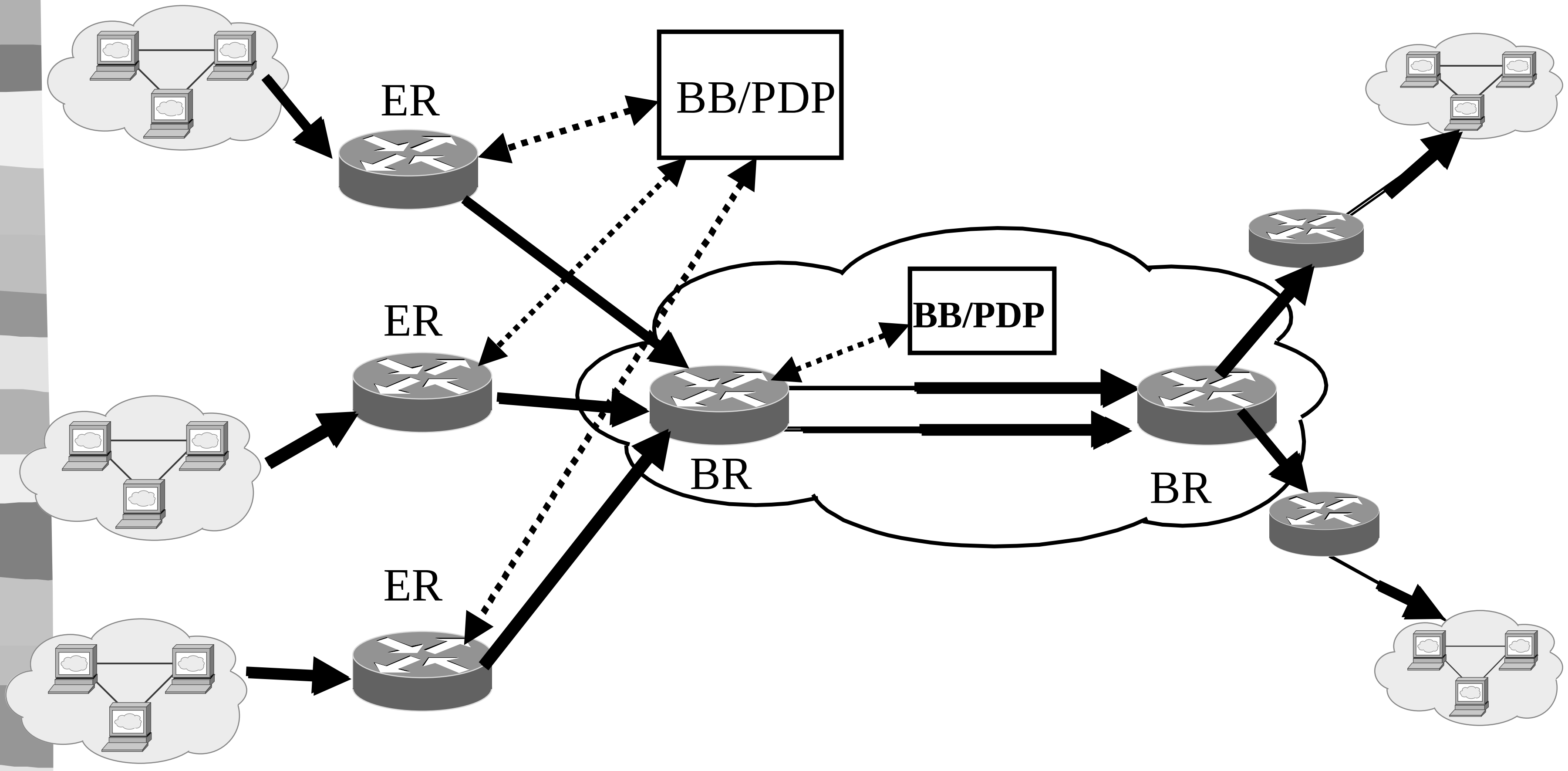
RSVP RESV

RSVP router

Next hop

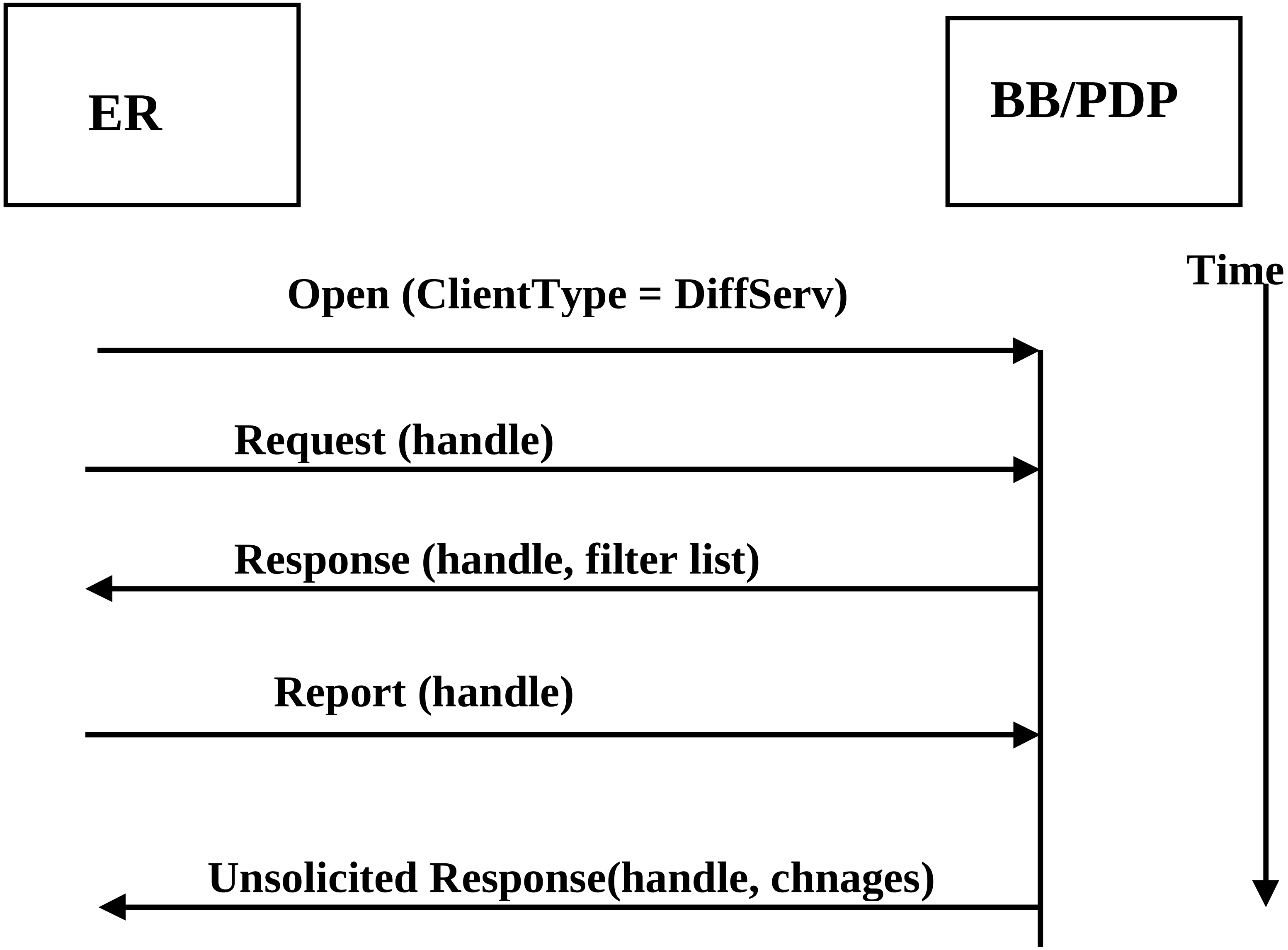
Use of COPS with Diff-Serv

- Edge routers (ERs) rely on BB/PDP to make policy decisions
  - No explicit e2e signaling to ER to trigger policy decision (ex., IP telephony call setup)
  - provides a way to configure ERs with a list of packet filters and accompanying actions
  - provides a way to asynchronously notify ERs about changes to filters/actions
  - allows ERs to log usage/accounting info


ER
BB/PDP
ER
BB/PDP
BR
BR
ER

Use of COPS for Diffserv (contd.)

- Use the configuration operation in COPS
- Example of Interaction:
  - PEP *open*s COPS session with ClientType=DiffServ
  - PEP *request*s a filter list
    - response: Filter list -- <filter criteria, action>+
    - policy tree defined for data format
  - BB/PDP can update the filter list any time using *unsolicited response*
  - PEP notifies BB/PDP of status/usage via *report* messages



ER

BB/PDP

Time

Open (ClientType = DiffServ)

Request (handle)

Response (handle, filter list)

Report (handle)

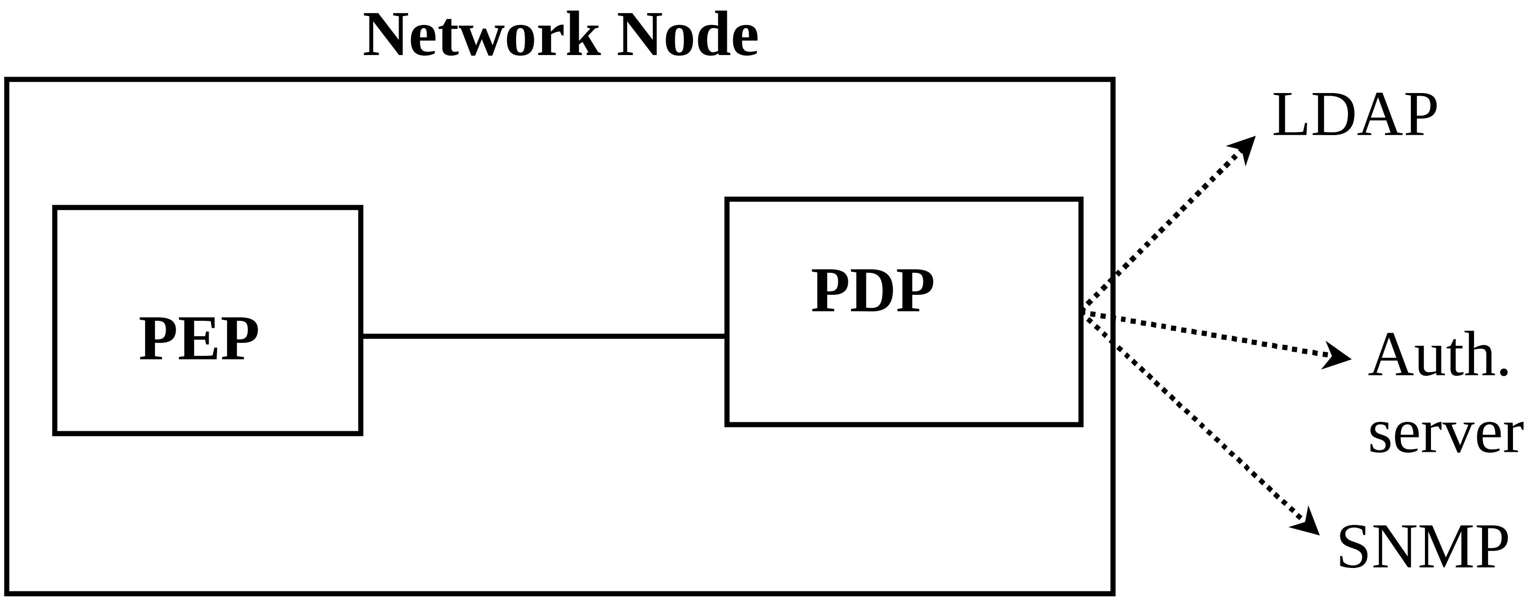
Unsolicited Response(handle, chnages)

Backup

Points

- Support for preemption
  - e.g., remove previously installed reservations
- Support for many styles of policies
  - relative priority, bi-lateral, multi-lateral
  - Scalability:
    - **not necessary to contact PDP at each node**
- Provision for monitoring and accounting
- Fault tolerance/recovery (PDP failures, partitions and merging, etc.)

Possible Configurations


Network Node
PEP
PDP
LDAP
Auth. server
SNMP