INSTITUTE
OF COMMUNICATION,
INFORMATION
AND PERCEPTION
TECHNOLOGIES

Scuola Superiore
Sant'Anna

Telecom Italia | TIM

cnit

Monitoring plane architecture for modern cloud-based networks

N. Sambo[1], A. Di Giglio[2], A. Pagano[2], F. Cugini[3], P. Castoldi[1]

1: Scuola Superiore Sant'Anna, Pisa, Italy
2: TIM
3: CNIT, Pisa, Italy

ORCHESTRA

Workshop "Big Data for networking: opportunities, challenges, and requirements", ONDM 2016

Introduction

- Toward flexible, agile, and programmable cloud-based networks:

Introduction

- Toward flexible, agile, and programmable cloud-based networks:
  - configurable transmission parameters in transponders at the edge of optical backbone [a,b]

[a] A. Napoli et al., ComMag vol. 53 n. 2, 2015
[b] N. Sambo et al., ComMag vol. 53 n. 2, 2015

Introduction

- Toward flexible, agile, and programmable cloud-based networks:
  - configurable transmission parameters in transponders at the edge of optical backbone [a,b]
  - reduction of margins: possibility to have not-considered degradations, e.g. aging [c,d] → soft failures (BER degradation) more frequent

[a] A. Napoli et al., ComMag vol. 53 n. 2, 2015
[b] N. Sambo et al., ComMag vol. 53 n. 2, 2015

Introduction

- Toward flexible, agile, and programmable cloud-based networks:
  - configurable transmission parameters in transponders at the edge of optical backbone [a,b]
  - reduction of margins: possibility to have not-considered degradations, e.g. aging [c,d] → soft failures (BER degradation) more frequent
  - e.g., alien wavelengths from data centers injected in optical backbone

[a] A. Napoli et al., ComMag vol. 53 n. 2, 2015

[b] N. Sambo et al., ComMag vol. 53 n. 2, 2015

[c] J.-L. Auge, paper OTu2A.1, OFC 2013

[d] Y. Pointurier, invited talk, OFC 2016

Introduction

- Toward flexible, agile, and programmable cloud-based networks:
  - configurable transmission parameters in transponders at the edge of optical backbone [a,b]
  - reduction of margins: possibility to have not-considered degradations, e.g. aging [c,d] → soft failures (BER degradation) more frequent
  - e.g., alien wavelengths from data centers injected in optical backbone
- Operation, Administration, and Maintainance (OAM) are key functionalities to verify Quality of Transmission (QoT) and Quality of Service (QoS)

[a] A. Napoli et al., ComMag vol. 53 n. 2, 2015
[b] N. Sambo et al., ComMag vol. 53 n. 2, 2015
[c] J.-L. Auge, paper OTu2A.1, OFC 2013
[d] Y. Pointurier, invited talk, OFC 2016

Introduction

- Toward flexible, agile, and programmable cloud-based networks:
  - configurable transmission parameters in transponders at the edge of optical backbone [a,b]
  - reduction of margins: possibility to have not-considered degradations, e.g. aging [c,d] → soft failures (BER degradation) more frequent
  - e.g., alien wavelengths from data centers injected in optical backbone
- Operation, Administration, and Maintainance (OAM) are key functionalities to verify Quality of Transmission (QoT) and Quality of Service (QoS)
- Control/management: ABNO architecture includes OAM functionalities

[a] A. Napoli et al., ComMag vol. 53 n. 2, 2015
[b] N. Sambo et al., ComMag vol. 53 n. 2, 2015
[c] J.-L. Auge, paper OTu2A.1, OFC 2013
[d] Y. Pointurier, invited talk, OFC 2016

Introduction

- Toward flexible, agile, and programmable cloud-based networks:
  - configurable transmission parameters in transponders at the edge of optical backbone [a,b]
  - reduction of margins: possibility to have not-considered degradations, e.g. aging [c,d] → soft failures (BER degradation) more frequent
  - e.g., alien wavelengths from data centers injected in optical backbone
- Operation, Administration, and Maintainance (OAM) are key functionalities to verify Quality of Transmission (QoT) and Quality of Service (QoS)
- Control/management: ABNO architecture includes OAM functionalities

This talk will mainly focus on **resiliency**

[a] A. Napoli et al., ComMag vol. 53 n. 2, 2015
[b] N. Sambo et al., ComMag vol. 53 n. 2, 2015
[c] J.-L. Auge, paper OTu2A.1, OFC 2013
[d] Y. Pointurier, invited talk, OFC 2016

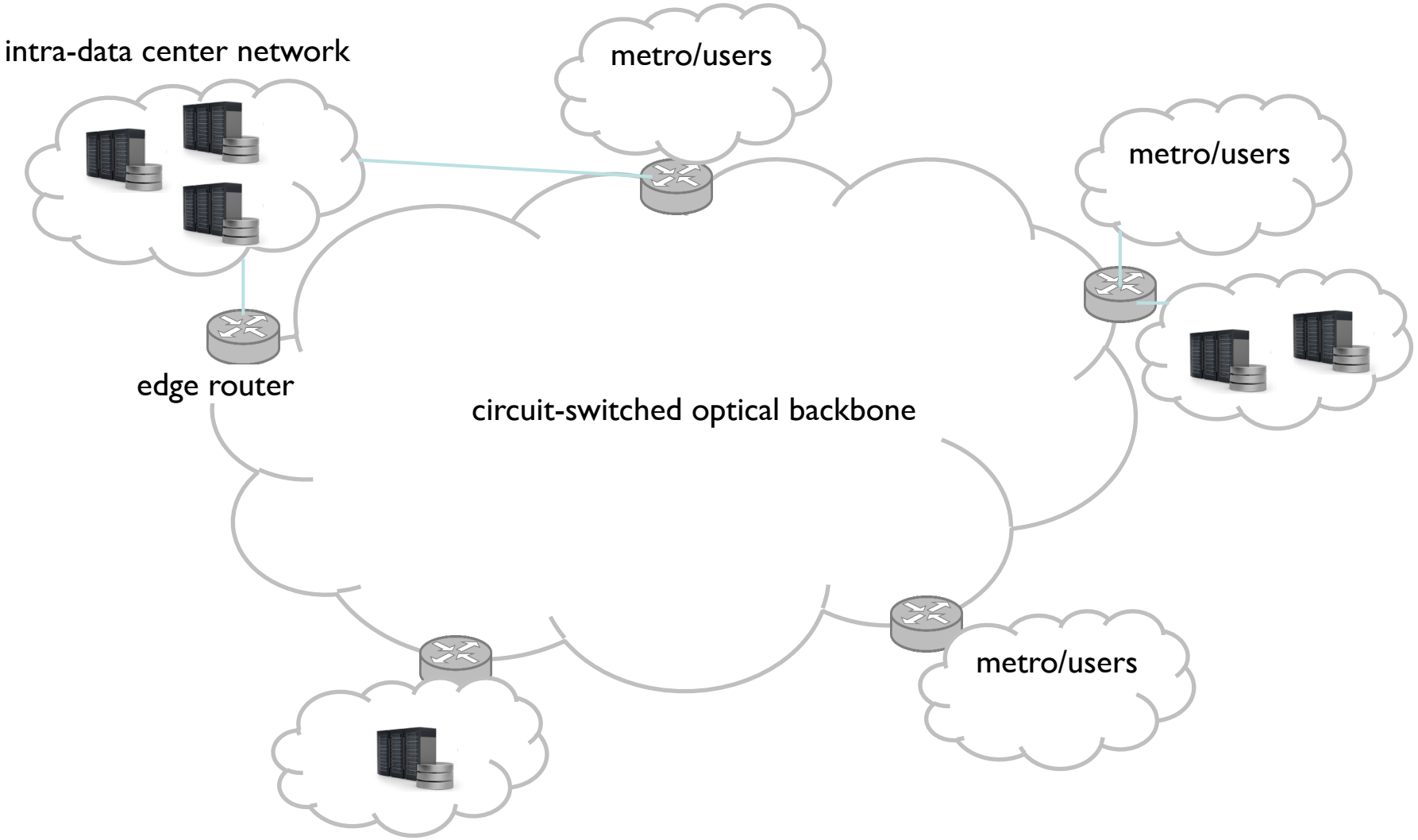
Introduction

- Toward flexible, agile, and programmable cloud-based networks:
  - configurable transmission parameters in transponders at the edge of optical backbone [a,b]
  - reduction of margins: possibility to have not-considered degradations, e.g. aging [c,d] → soft failures (BER degradation) more frequent
  - e.g., alien wavelengths from data centers injected in optical backbone
- Operation, Administration, and Maintainance (OAM) are key functionalities to verify Quality of Transmission (QoT) and Quality of Service (QoS)
- Control/management: ABNO architecture includes OAM functionalities

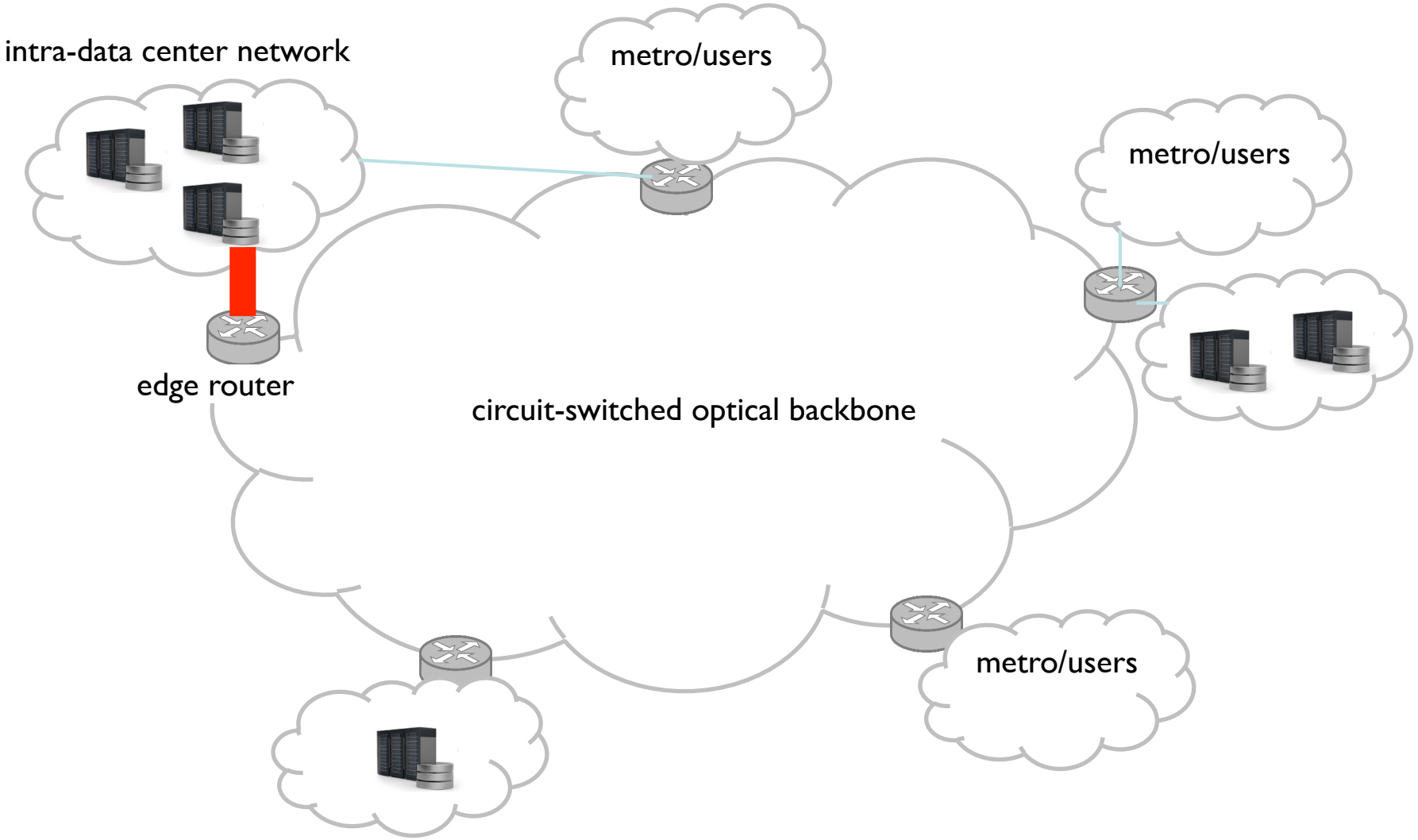
This talk will mainly focus on **resiliency**

[a] A. Napoli et al., ComMag vol. 53 n. 2, 2015
[b] N. Sambo et al., ComMag vol. 53 n. 2, 2015
[c] J.-L. Auge, paper OTu2A.1, OFC 2013
[d] Y. Pointurier, invited talk, OFC 2016

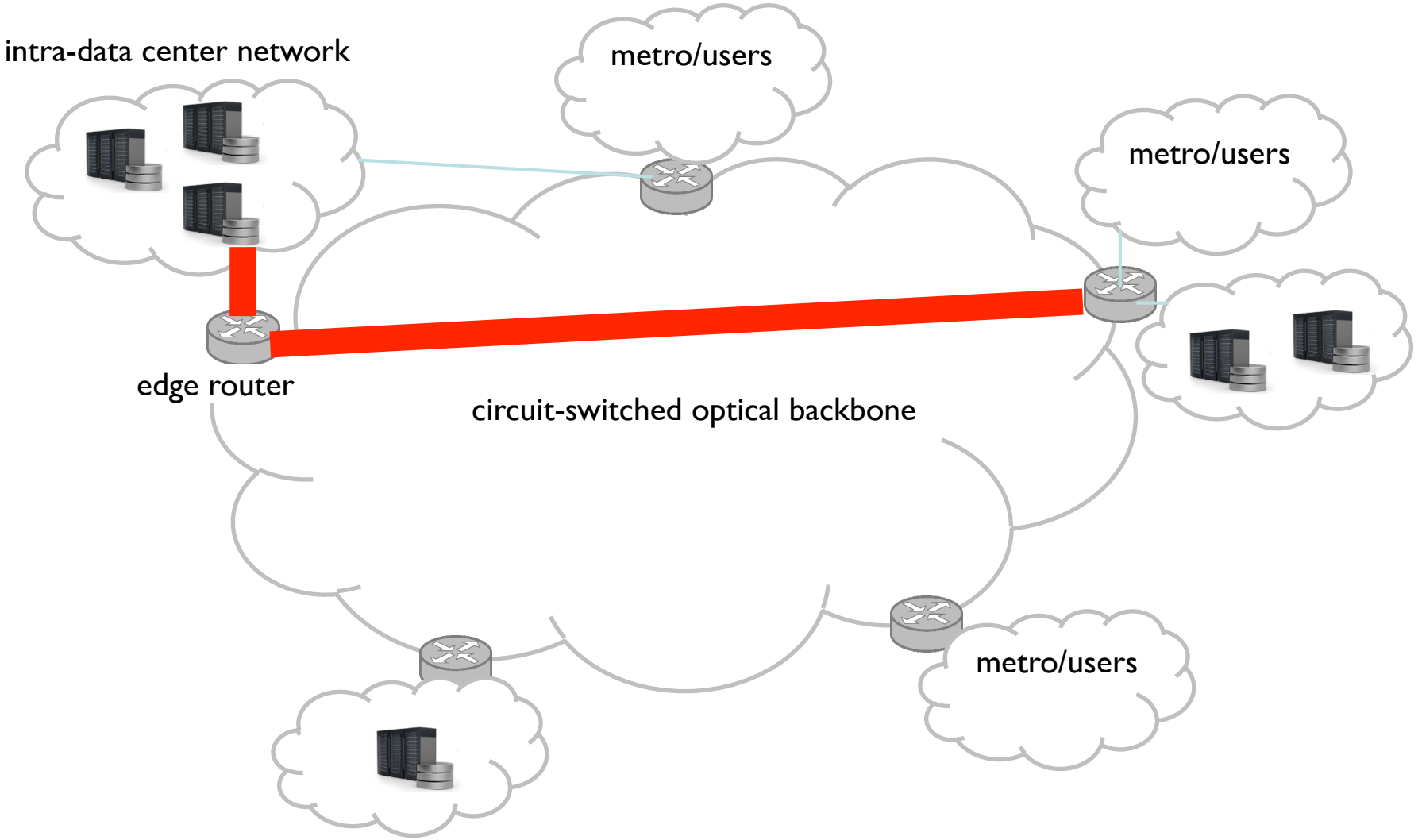
Network architecture and scenario



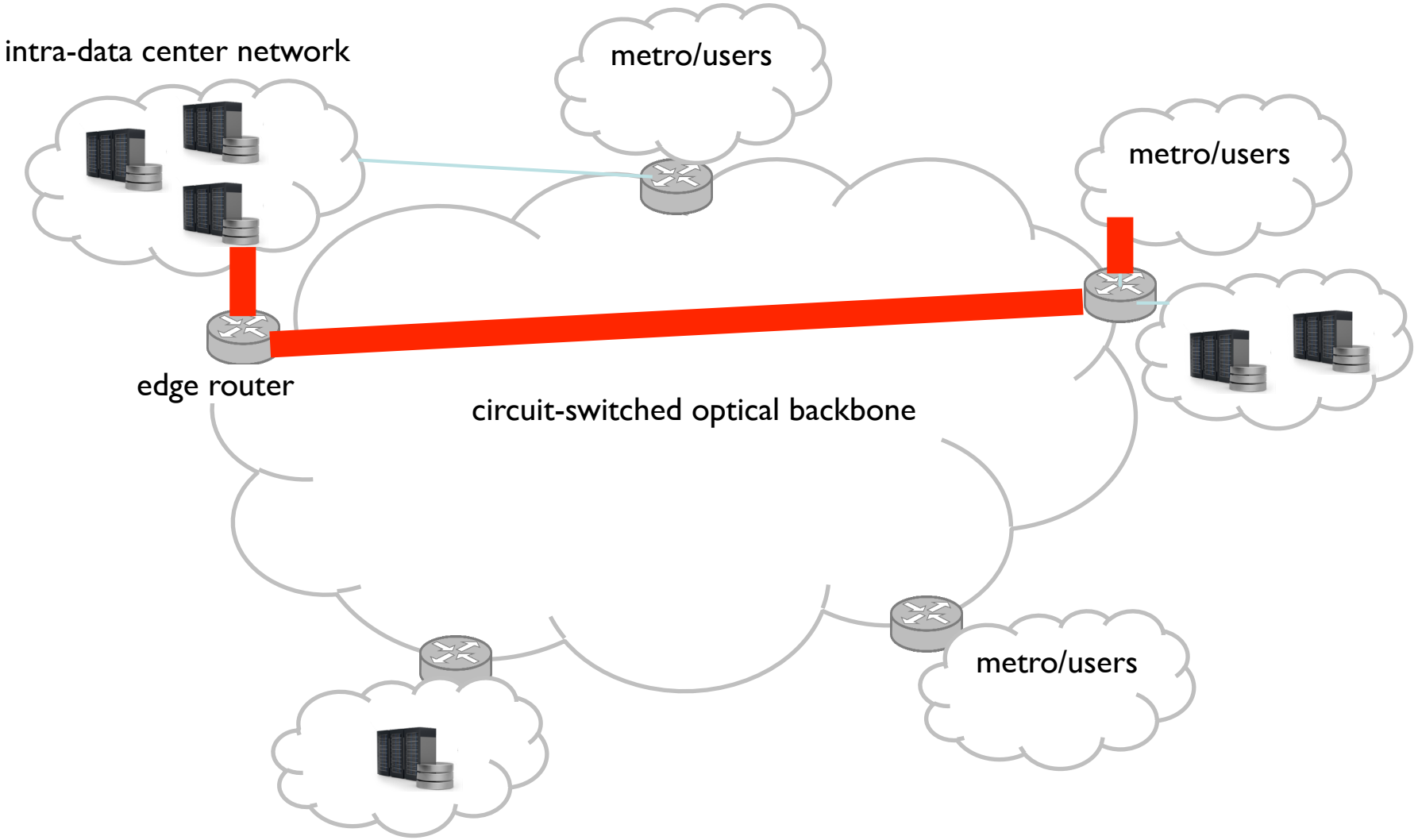
Network architecture and scenario



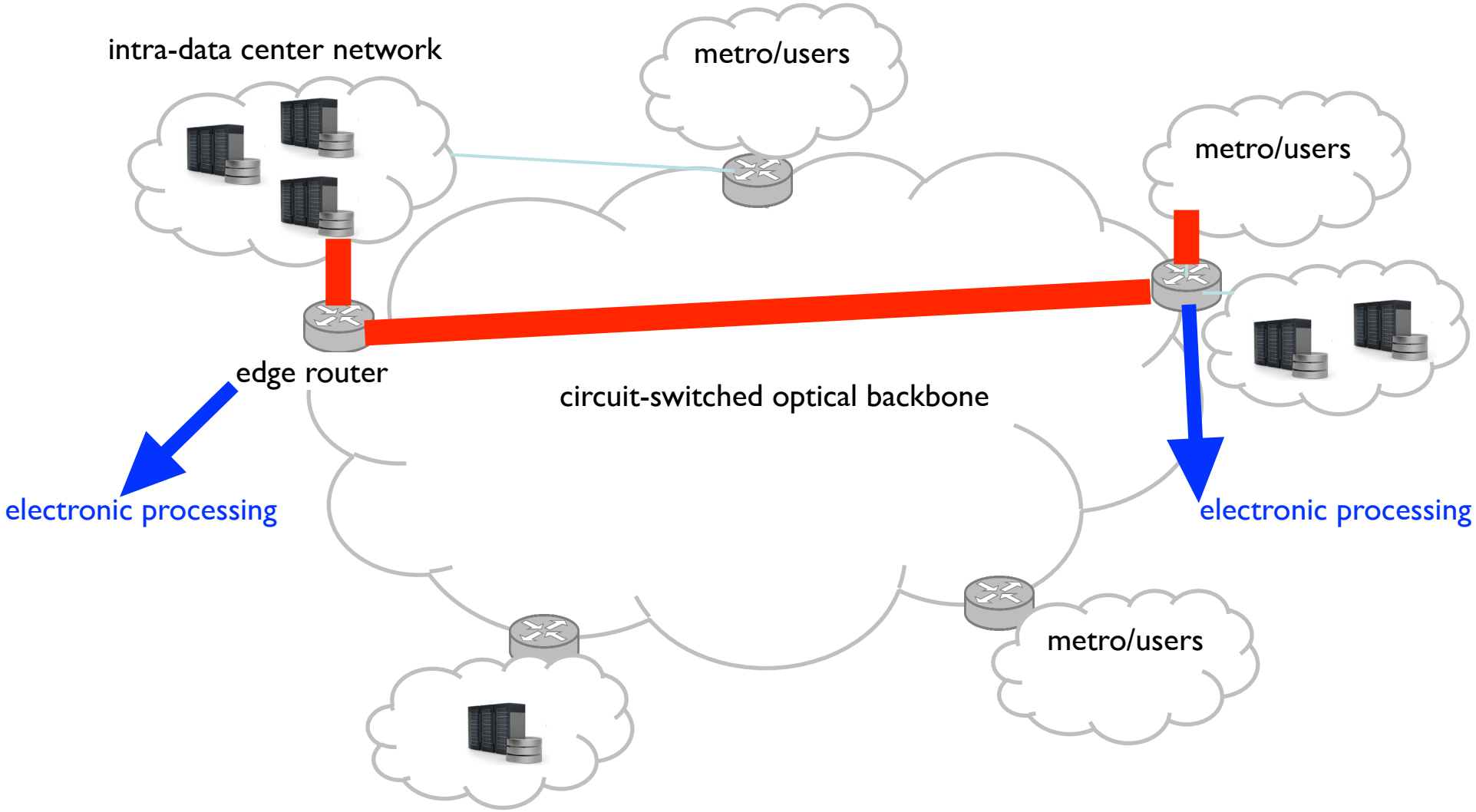
Network architecture and scenario



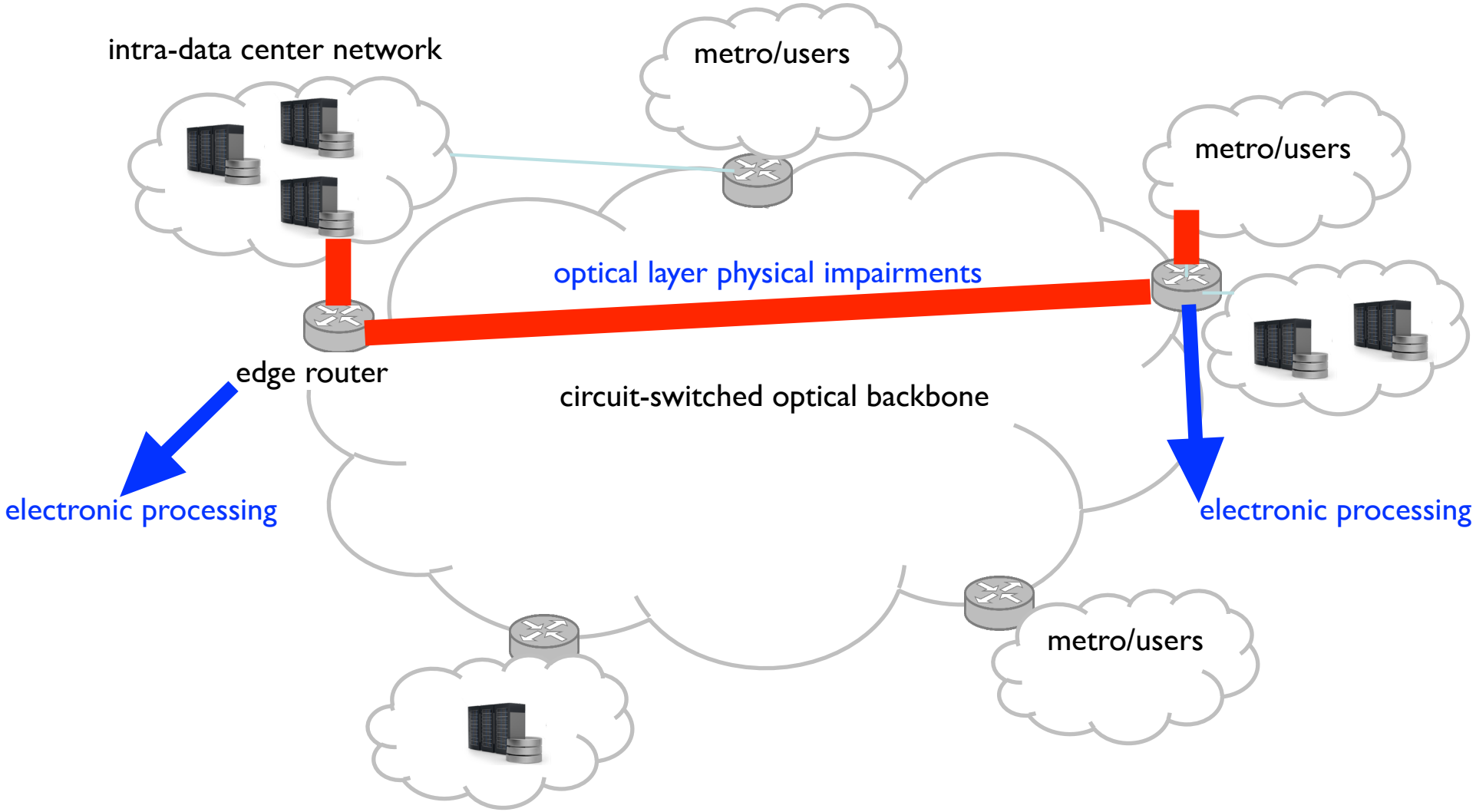
Network architecture and scenario



Network architecture and scenario



Network architecture and scenario



Some numbers on data centers

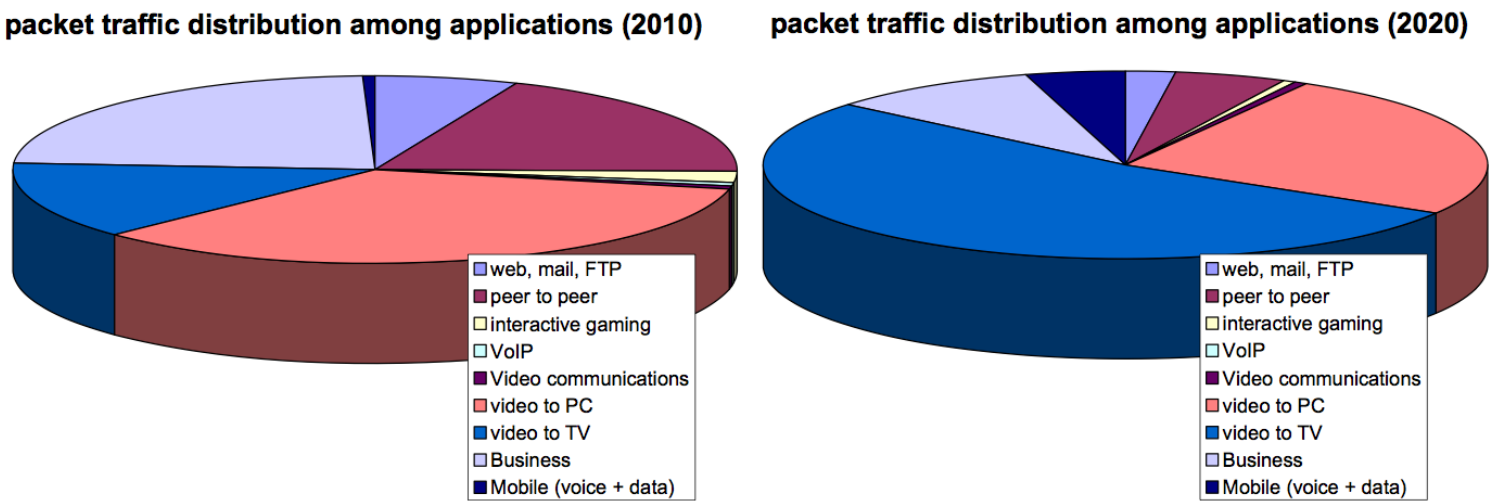


FIG. by STRONGEST D2.1

Specifics of the services, which can be classified as:

- **Interactive:** delay in RTT below 150ms and jitter below 10ms
- **Guaranteed:** delay below 400 ms, no specific requirements on jitter (buffering is enough)
- **Best effort**

Considerations on QoS and QoT

- QoS affected by QoT: e.g., packet loss rate (PLR) — thus delay and jitter because of packet retransmission — influenced by bit error rate (BER)

Considerations on QoS and QoT

- QoS affected by QoT: e.g., packet loss rate (PLR) — thus delay and jitter because of packet retransmission — influenced by bit error rate (BER)
- QoS affected by electronic layer: e.g., delay, jitter influenced by queuing time at the edge router

Considerations on QoS and QoT

- QoS affected by QoT: e.g., packet loss rate (PLR) — thus delay and jitter because of packet retransmission — influenced by bit error rate (BER)
- QoS affected by electronic layer: e.g., delay, jitter influenced by queuing time at the edge router

- Observation of the optical physical layer (e.g., BER or correlated parameters such as OSNR) is key to prevent PLR increase → an increase of BER should trigger some reaction before PLR increase

Considerations on QoS and QoT

- QoS affected by QoT: e.g., packet loss rate (PLR) — thus delay and jitter because of packet retransmission — influenced by bit error rate (BER)
- QoS affected by electronic layer: e.g., delay, jitter influenced by queuing time at the edge router

- Observation of the optical physical layer (e.g., BER or correlated parameters such as OSNR) is key to prevent PLR increase → an increase of BER should trigger some reaction before PLR increase
- Observation of the physical layer is not enough: service level parameters should be monitored (delay, PLR, etc.). A worsening of the service performance could be not due to the optical physical layer: e.g., could be due to edge routers

Considerations on QoS and QoT

- QoS affected by QoT: e.g., packet loss rate (PLR) — thus delay and jitter because of packet retransmission — influenced by bit error rate (BER)
- QoS affected by electronic layer: e.g., delay, jitter influenced by queuing time at the edge router

- Observation of the optical physical layer (e.g., BER or correlated parameters such as OSNR) is key to prevent PLR increase → an increase of BER should trigger some reaction before PLR increase
- Observation of the physical layer is not enough: service level parameters should be monitored (delay, PLR, etc.). A worsening of the service performance could be not due to the optical physical layer: e.g., could be due to edge routers

ABNO architecture includes functional modules **controlling and managing networks and services**

Application-based Network Operations (ABNO)

IETF RFC 7491

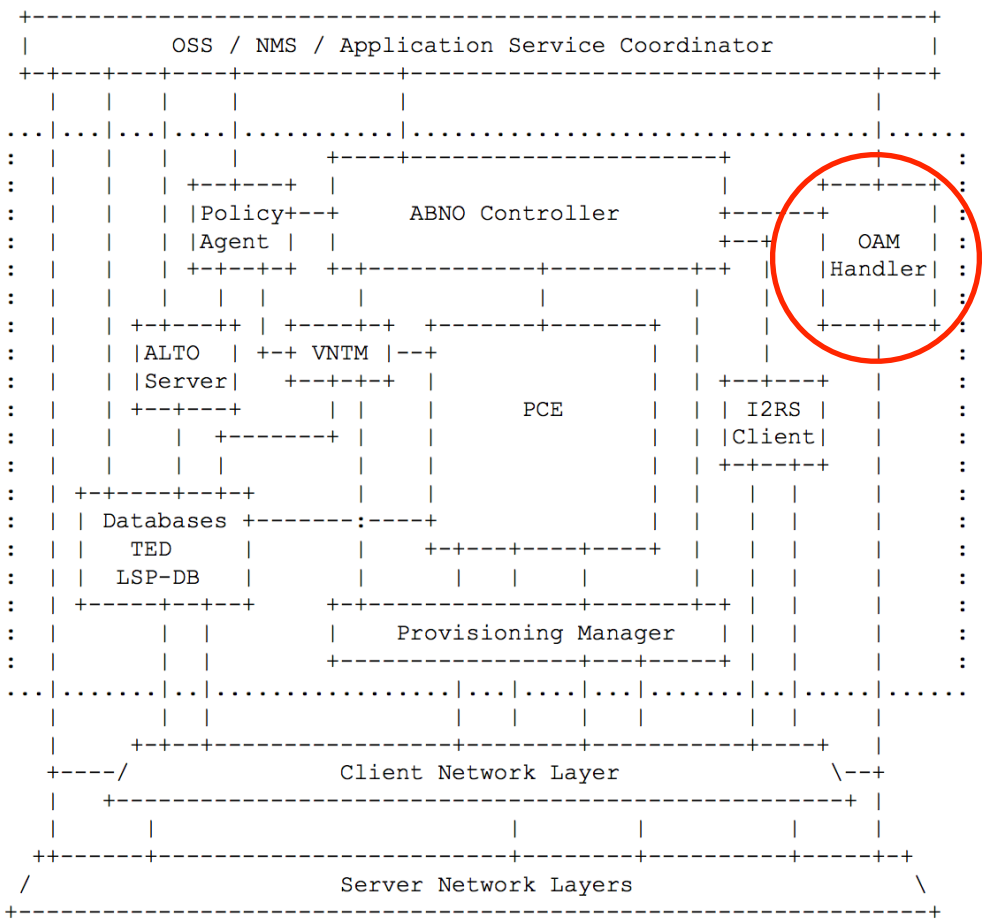


Figure 1 : Generic ABNO Architecture

- OAM receiving **alerts** about potential problems
- **correlating** them
- **triggering other components** of the ABNO system to take action to preserve or recover the services

Open issues

- Alarms are typically managed considering hard failures; soft failures are not considered
- Lack of cross-layer quality parameter correlation
- Several alarms can be generated at different levels in the presence of soft failure

Open issues

- Alarms are typically managed considering hard failures; soft failures are not considered
- Lack of cross-layer quality parameter correlation
- Several alarms can be generated at different levels in the presence of soft failure

→ how to handle a huge amount of alarms especially considering soft-failures that will be more frequent?

Open issues

- Alarms are typically managed considering hard failures; soft failures are not considered
- Lack of cross-layer quality parameter correlation
- Several alarms can be generated at different levels in the presence of soft failure

→ how to handle a huge amount of alarms especially considering soft-failures that will be more frequent?

→ are common management planes scalable?

Open issues

- Alarms are typically managed considering hard failures; soft failures are not considered
- Lack of cross-layer quality parameter correlation
- Several alarms can be generated at different levels in the presence of soft failure

→ how to handle a huge amount of alarms especially considering soft-failures that will be more frequent?
→ are common management planes scalable?
→ can soft/hard-failures or problems determining a service degradation easily localized and identified?

Open issues

- Alarms are typically managed considering hard failures; soft failures are not considered
- Lack of cross-layer quality parameter correlation
- Several alarms can be generated at different levels in the presence of soft failure

→ how to handle a huge amount of alarms especially considering soft-failures that will be more frequent?
→ are common management planes scalable?
→ can soft/hard-failures or problems determining a service degradation easily localized and identified?
→ which reaction? re-routing, more robust transmission?

Open issues

- Alarms are typically managed considering hard failures; soft failures are not considered
- Lack of cross-layer quality parameter correlation
- Several alarms can be generated at different levels in the presence of soft failure

→ how to handle a huge amount of alarms especially considering soft-failures that will be more frequent?
→ are common management planes scalable?
→ can soft/hard-failures or problems determining a service degradation easily localized and identified?
→ which reaction? re-routing, more robust transmission?

Need of **data analytics** for **alarm correlation** and **suppression**, fault **localization**, **type of failure** identification, and reaction decision

Open issues

- Alarms are typically managed considering hard failures; soft failures are not considered
- Lack of cross-layer quality parameter correlation
- Several alarms can be generated at different levels in the presence of soft failure

→ how to handle a huge amount of alarms especially considering soft-failures that will be more frequent?
→ are common management planes scalable?
→ can soft/hard-failures or problems determining a service degradation easily localized and identified?
→ which reaction? re-routing, more robust transmission?

Need of **data analytics** for **alarm correlation** and **suppression**, fault **localization**, **type of failure** identification, and reaction decision

Proposed hierarchical monitoring architecture aims at providing a way to gather monitoring information coming from different layers and network elements in a scalable way without overloading centralized controllers

Open issues

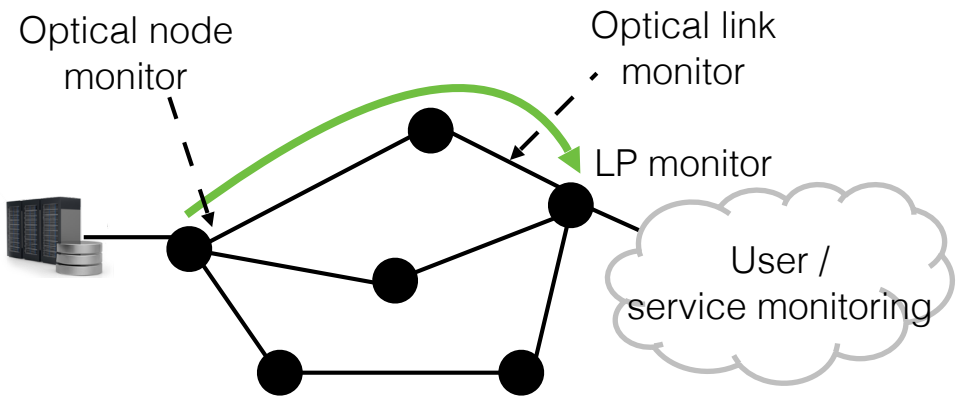
- Alarms are typically managed considering hard failures; soft failures are not considered
- Lack of cross-layer quality parameter correlation
- Several alarms can be generated at different levels in the presence of soft failure

→ how to handle a huge amount of alarms especially considering soft-failures that will be more frequent?
→ are common management planes scalable?
→ can soft/hard-failures or problems determining a service degradation easily localized and identified?
→ which reaction? re-routing, more robust transmission?

Need of **data analytics** for **alarm correlation** and **suppression**, fault **localization**, **type of failure** identification, and reaction decision

Proposed hierarchical monitoring architecture aims at providing a way to gather monitoring information coming from different layers and network elements in a scalable way without overloading centralized controllers

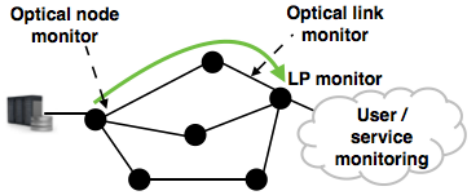
Monitors

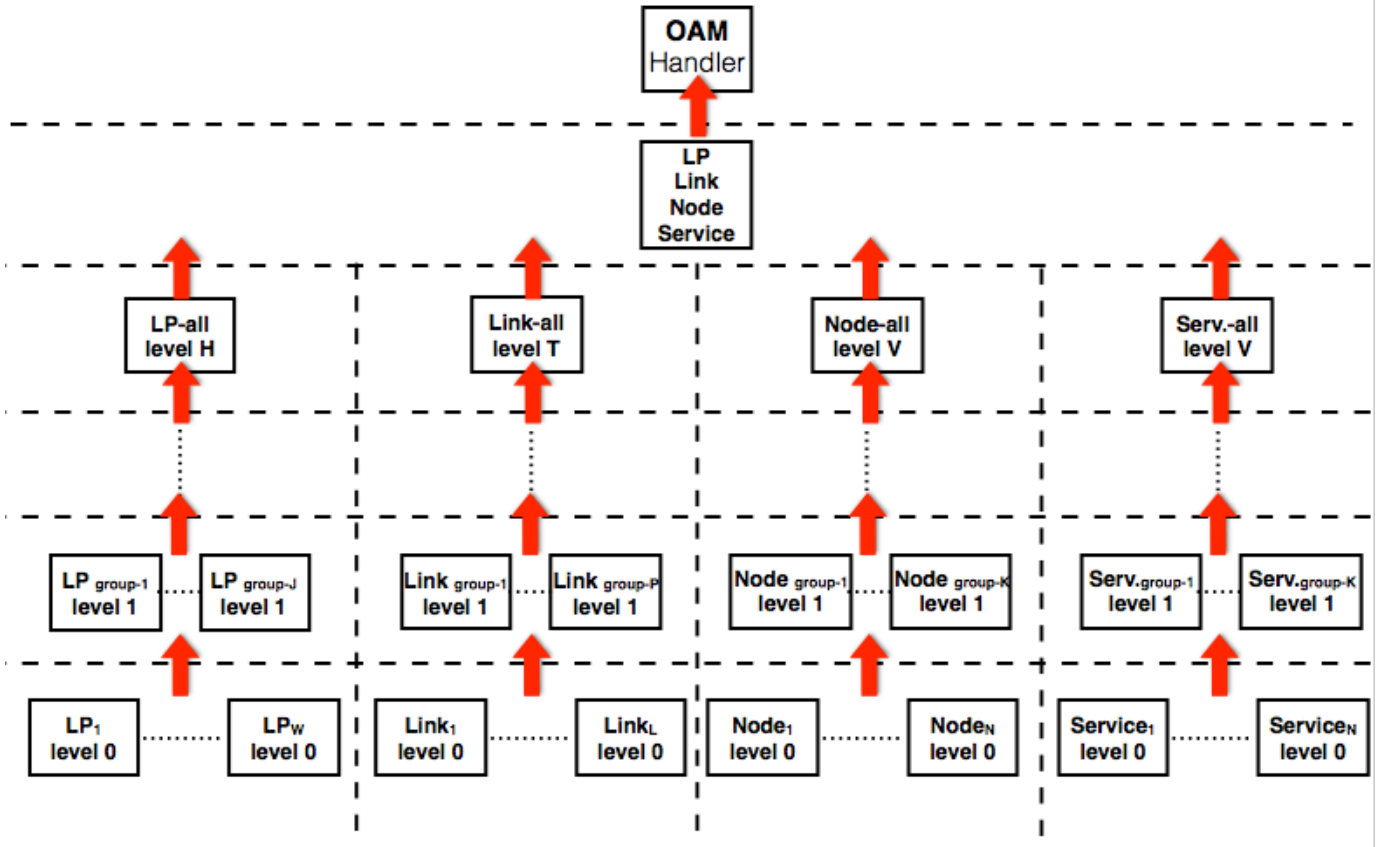


- Lightpath (LP) monitors are assumed integrated in the DSP of each lightpath coherent receiver (e.g., pre-FEC BER monitoring)
- Power monitors can be assumed for links and nodes
- Service monitors for PLR, delay, jitter

Hierarchical monitoring architecture

- Each entity is responsible for specific elements: e.g. a set of lightpaths

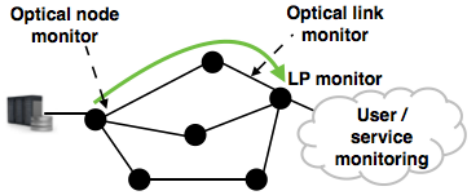


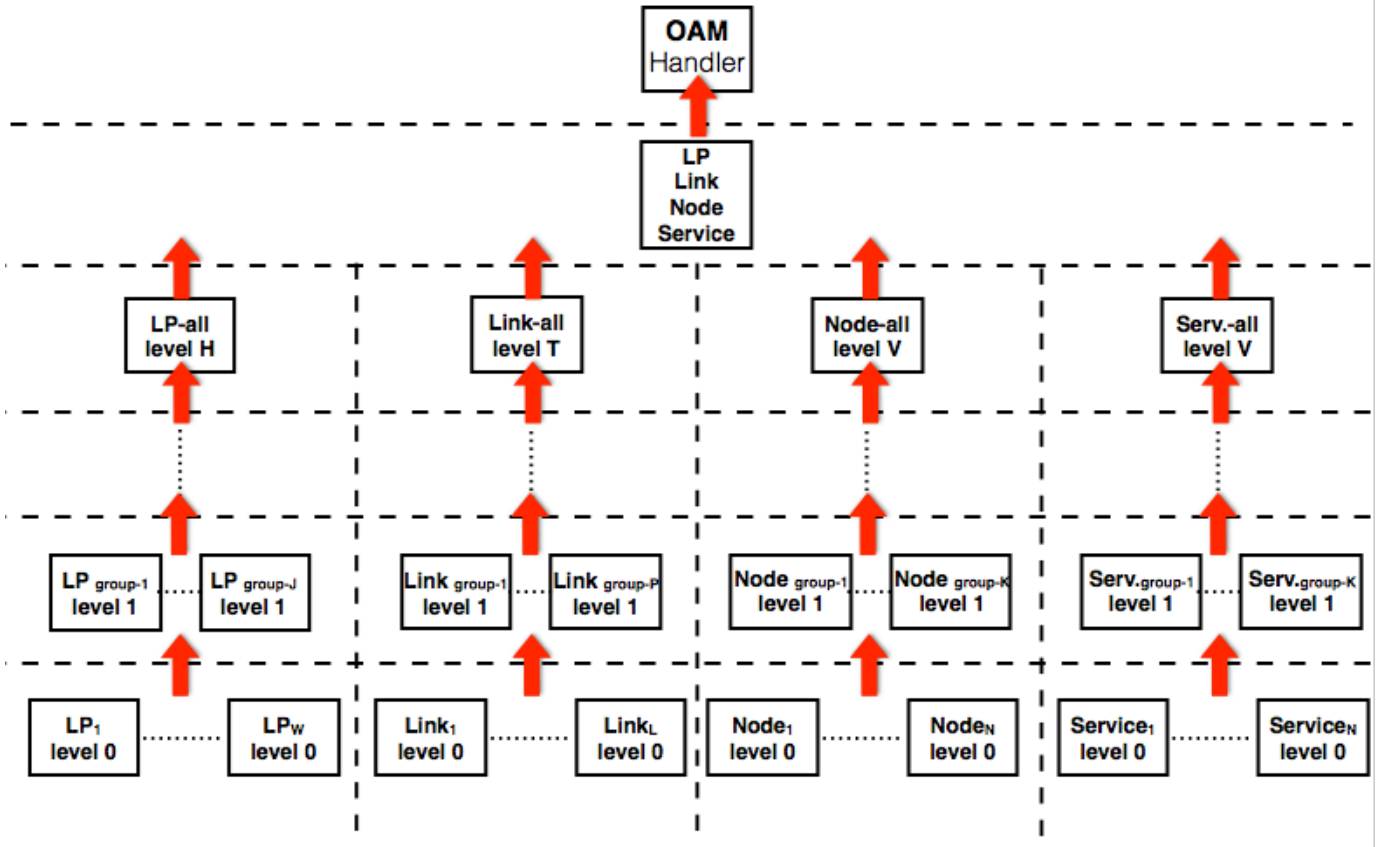


N. Sambo et al., "Monitoring Plane Architecture and OAM Handler", JLT, vol. 34, n. 8, 2016

Hierarchical monitoring architecture

- Each entity is responsible for specific elements: e.g. a set of lightpaths
- each layer receives information from down layers
  - correlation
  - actions
  - notifications to the upper layers

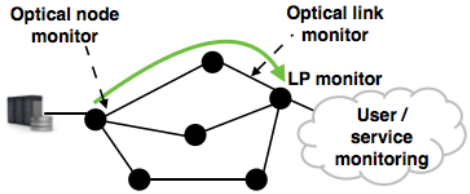


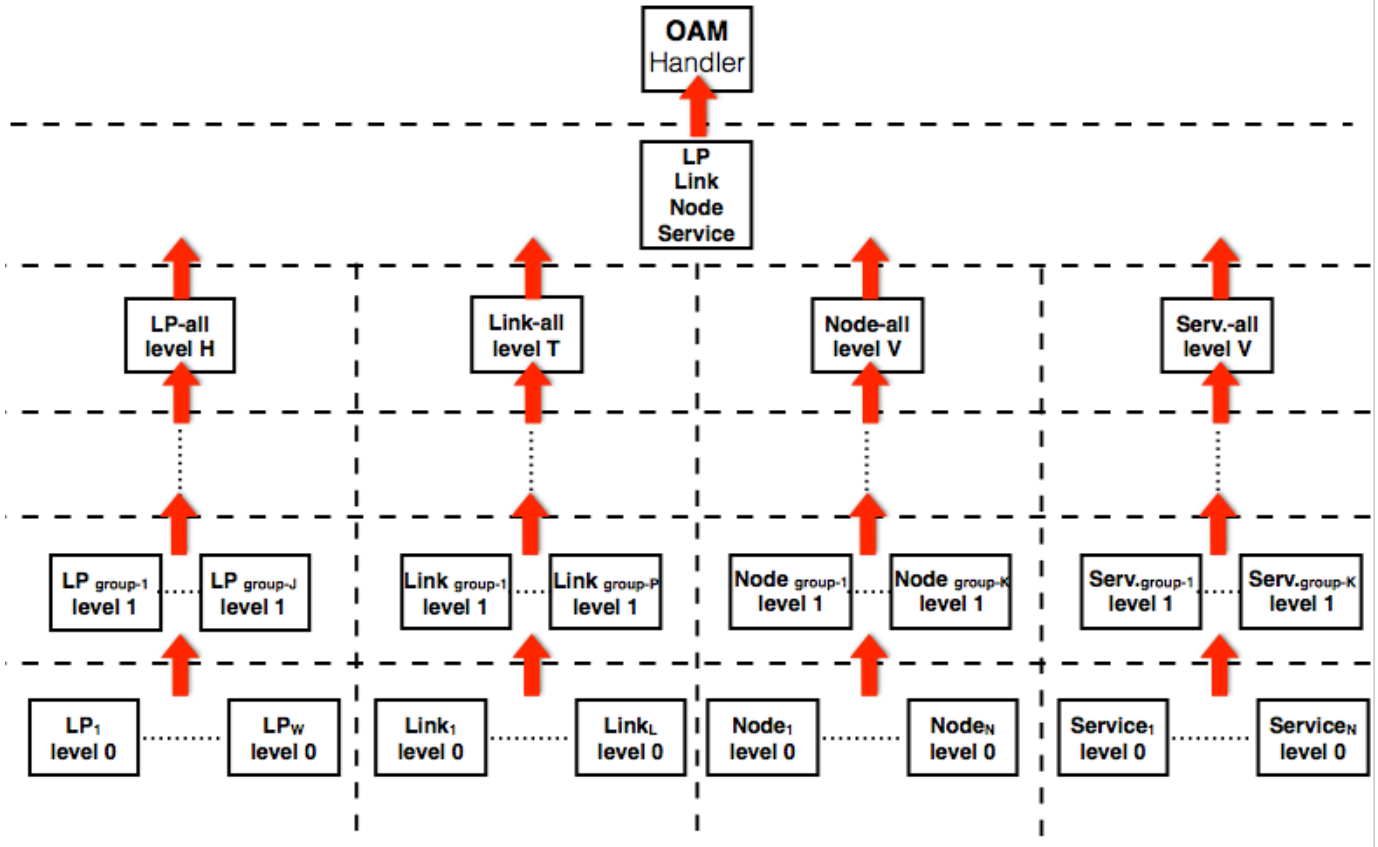


N. Sambo et al., "Monitoring Plane Architecture and OAM Handler", JLT, vol. 34, n. 8, 2016

Hierarchical monitoring architecture

- Each entity is responsible for specific elements: e.g. a set of lightpaths
- each layer receives information from down layers
  - correlation
  - actions
  - notifications to the upper layers
- Going up to higher layers, more responsibility

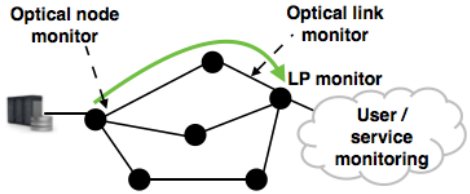


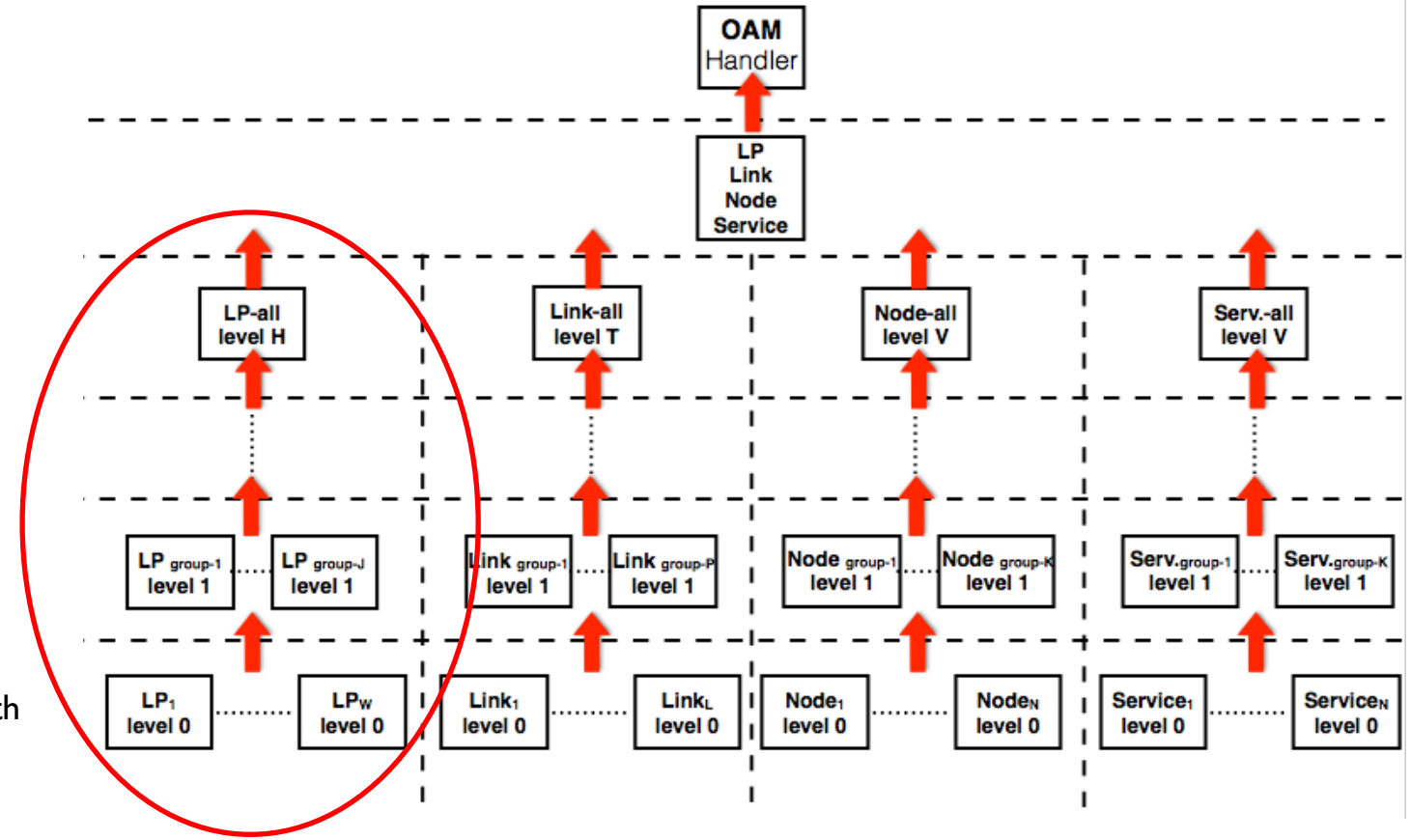


N. Sambo et al., "Monitoring Plane Architecture and OAM Handler", JLT, vol. 34, n. 8, 2016

Hierarchical monitoring architecture

- Each entity is responsible for specific elements: e.g. a set of lightpaths
- each layer receives information from down layers
  - correlation
  - actions
  - notifications to the upper layers
- Going up to higher layers, more responsibility



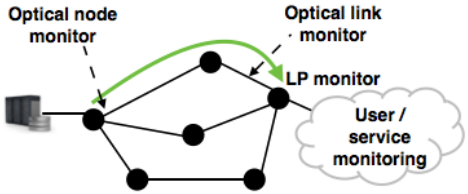


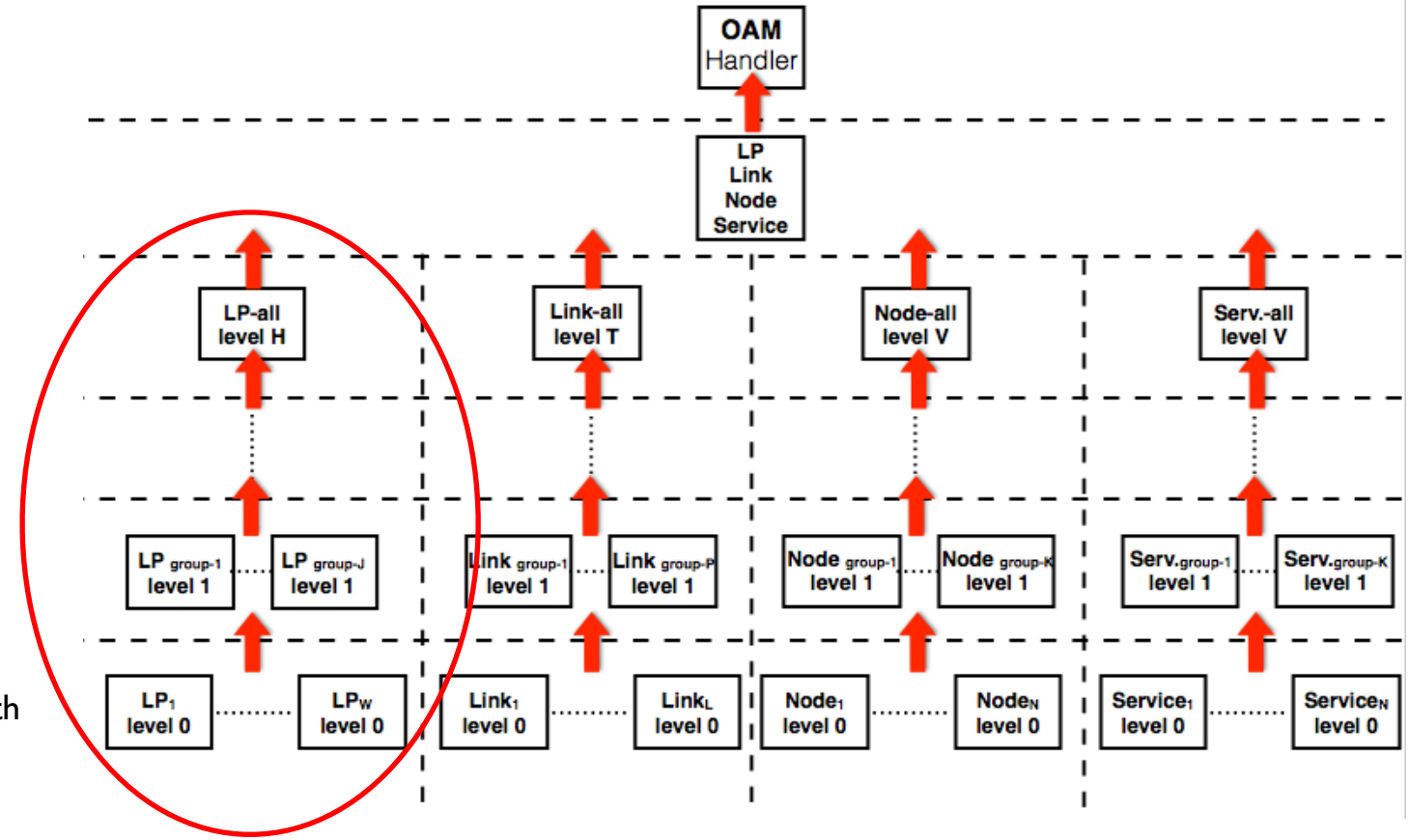
- EX: **LP level 0:** 1 per active lightpath

N. Sambo et al., "Monitoring Plane Architecture and OAM Handler", JLT, vol. 34, n. 8, 2016

Hierarchical monitoring architecture

- Each entity is responsible for specific elements: e.g. a set of lightpaths
- each layer receives information from down layers
  - correlation
  - actions
  - notifications to the upper layers
- Going up to higher layers, more responsibility

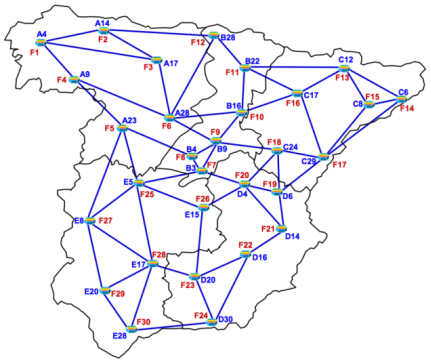




- EX: **LP group level 1:** each box group all the lightpaths starting from the same ingress node
- EX: **LP level 0:** 1 per active lightpath

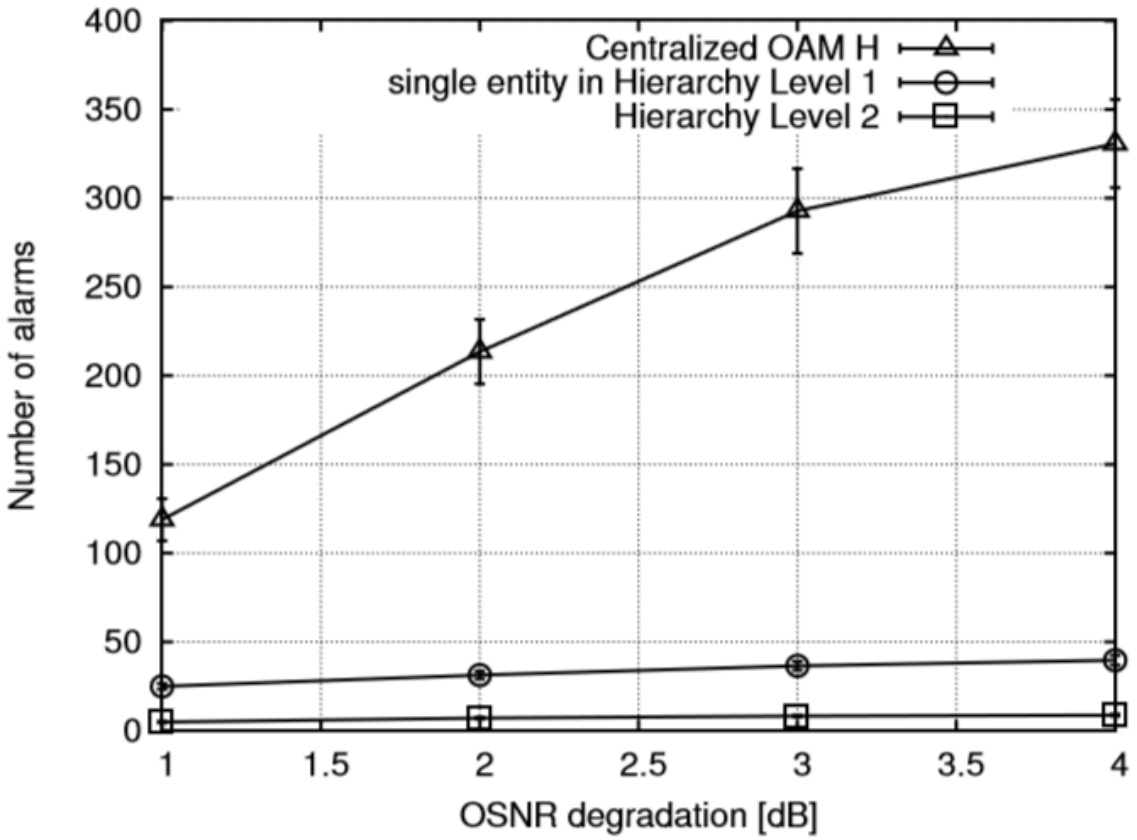
N. Sambo et al., "Monitoring Plane Architecture and OAM Handler", JLT, vol. 34, n. 8, 2016

Simulation scenario



- Comparison of two management architectures:
  - i) the proposed **hierarchical** monitoring architecture;
  - ii) a **centralized OAM** receiving all monitoring information and correlating them.
- **Soft-failure**: performance of a network element – such as an amplifier – are degraded causing the OSNR decrease of traversing lightpaths → some lightpaths suffer, others not: e.g. OSNR degradation may imply a BER increase over the threshold (thus, generating alarms) or not (not generating alarms)

Soft failure



Conclusions

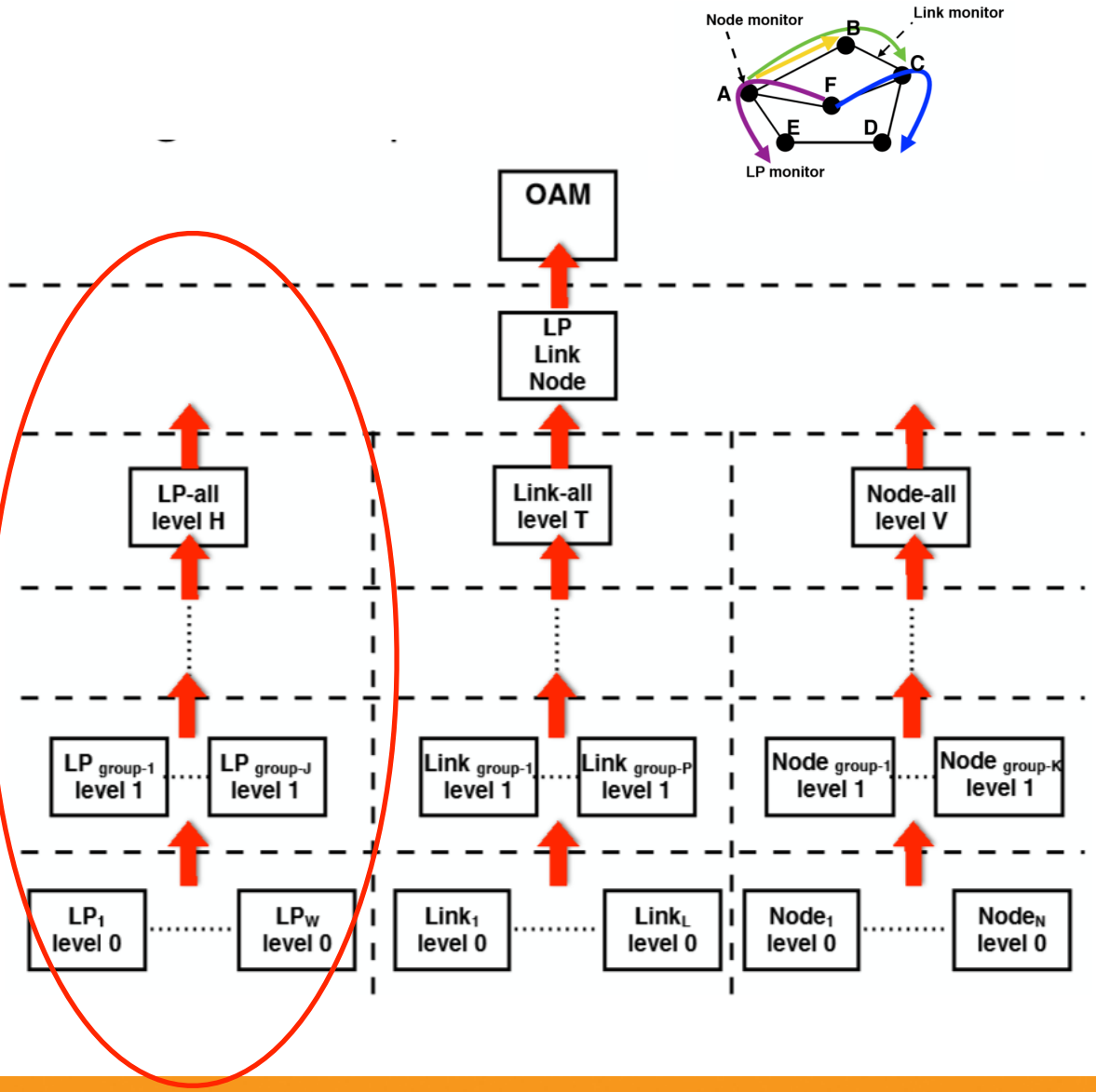
- Improve correlation for monitored parameters coming from different layers
- Management of soft failures: identification of the fault and localization
- Scalable management plane

- This work enhanced the hierarchical monitoring architecture proposed within the EU ORCHESTRA project
- ABNO OAM Handler functionalities are spread into several hierarchical layers, enabling to confine sets of monitored physical parameters within specific levels in the hierarchy:
  - Scalable solution
- Correlation of different-layer monitored parameters is enabled

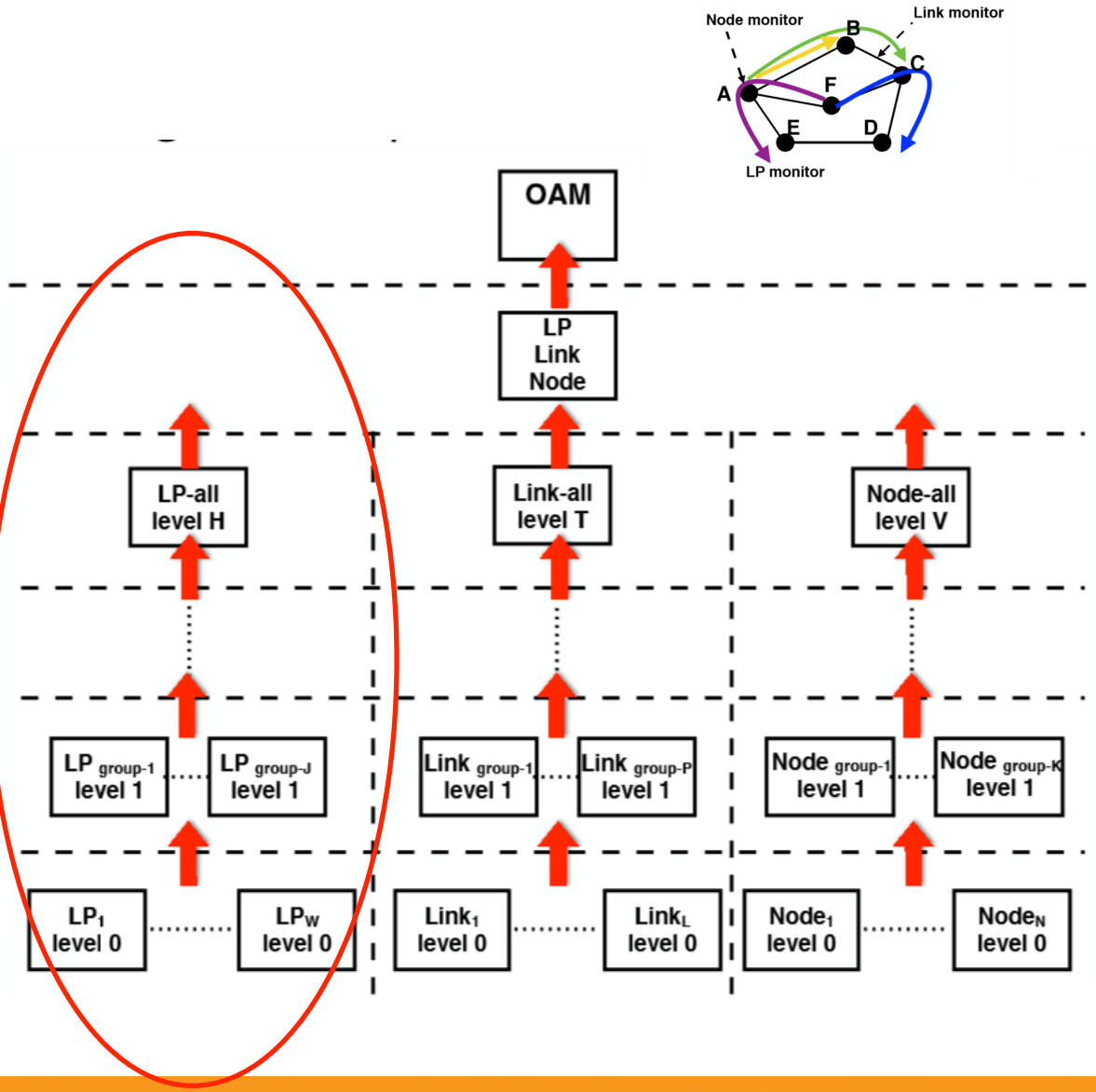
ACK: The work has been supported by the ORCHESTRA project.

email: nicola.sambo@sssup.it

Hierarchical monitoring architecture: example of responsibility

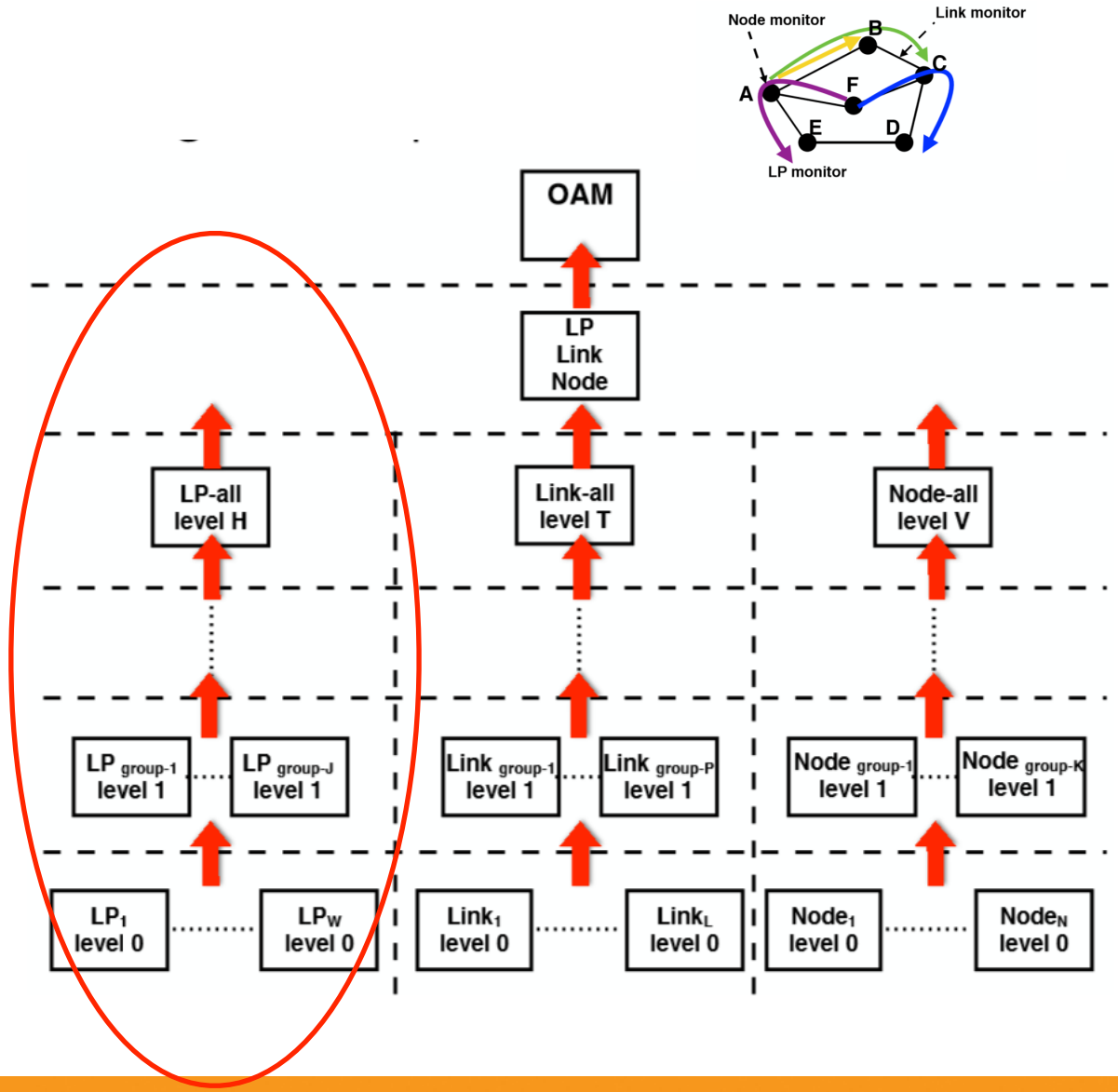


Hierarchical monitoring architecture: example of responsibility



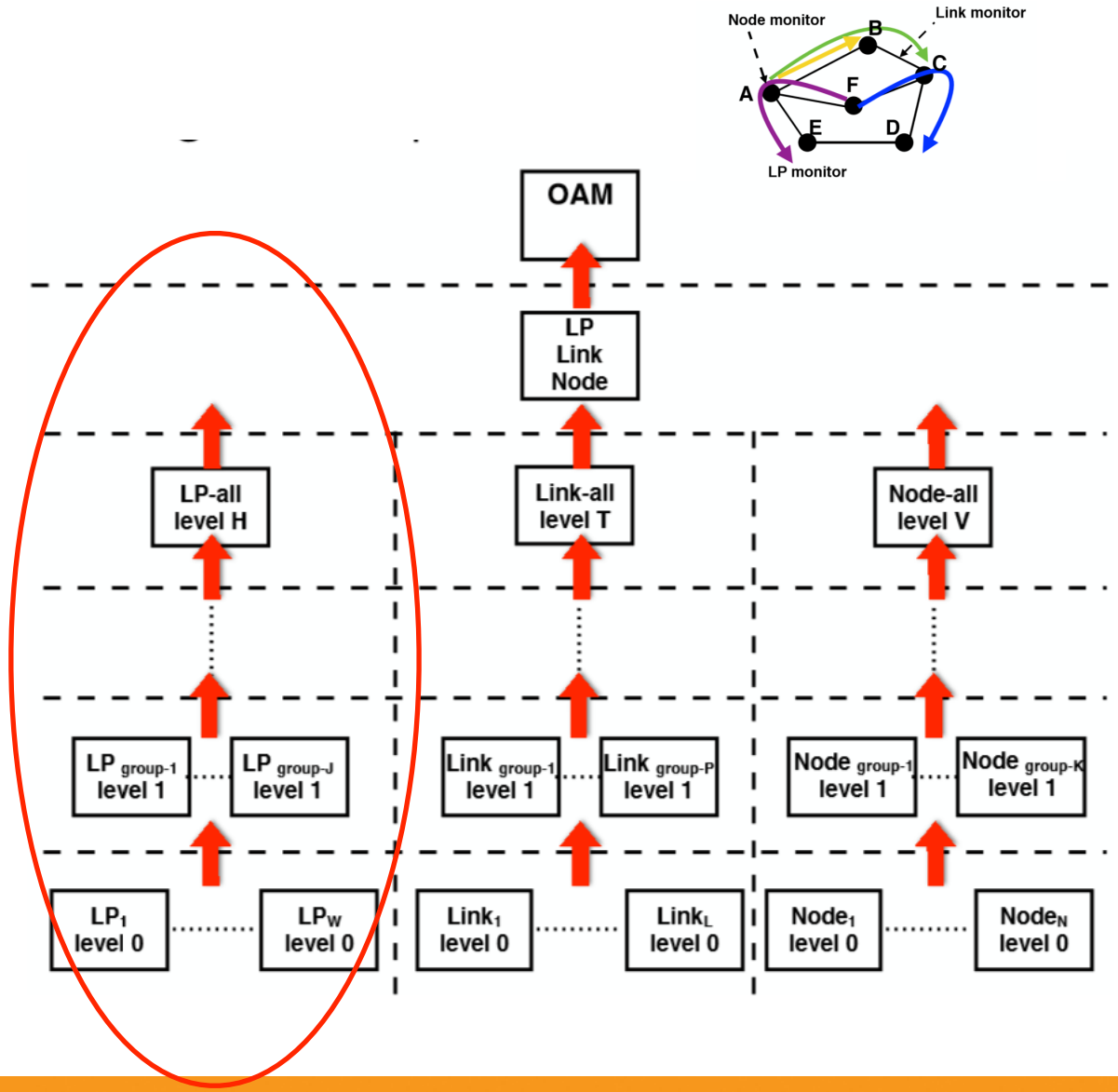
- **LP level 0:** 1 per active lightpath

Hierarchical monitoring architecture: example of responsibility



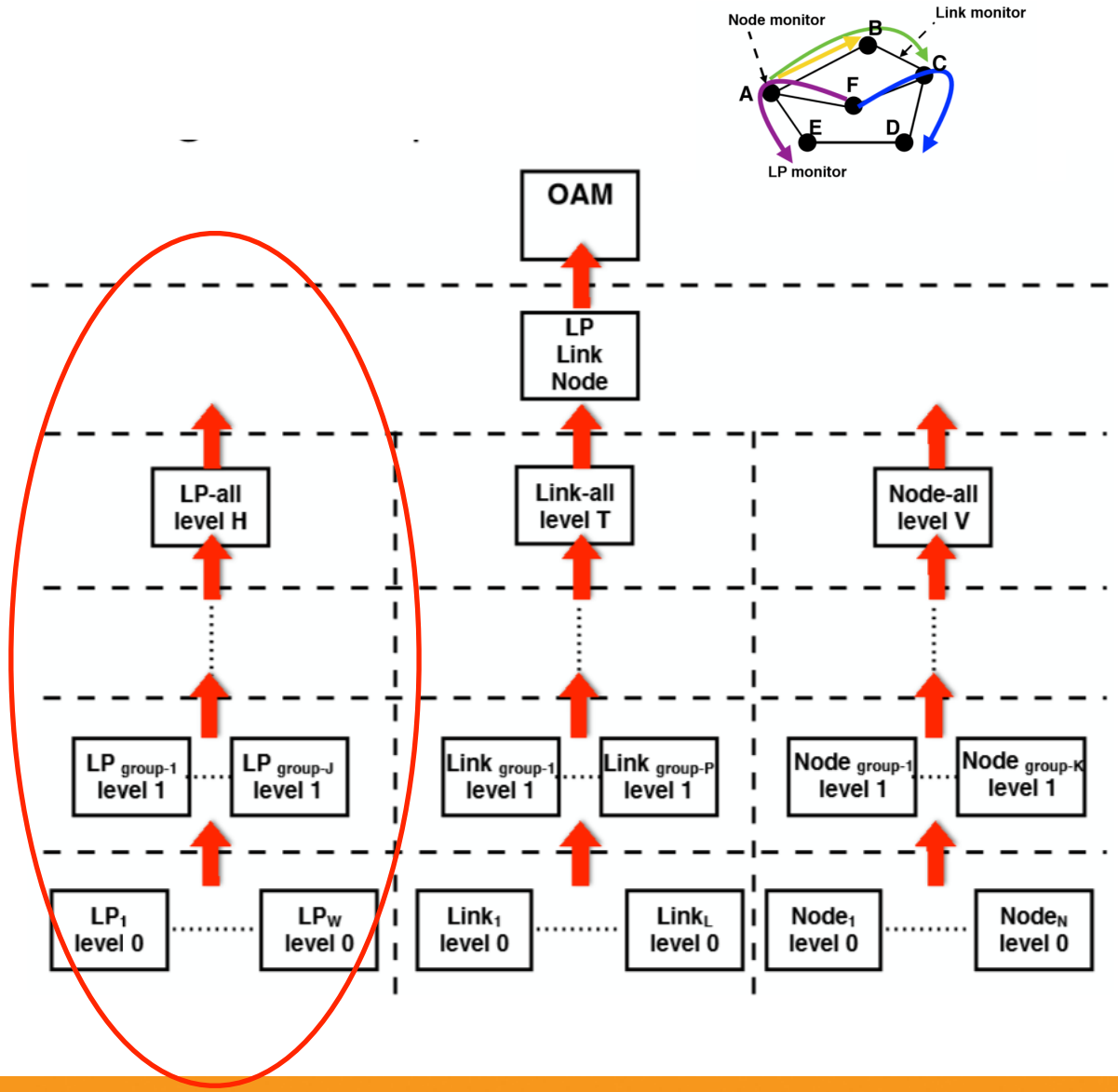
- **LP level 0:** 1 per active lightpath
- **LP group level 1:** each groups group all the lightpaths starting from the same ingress node

Hierarchical monitoring architecture: example of responsibility



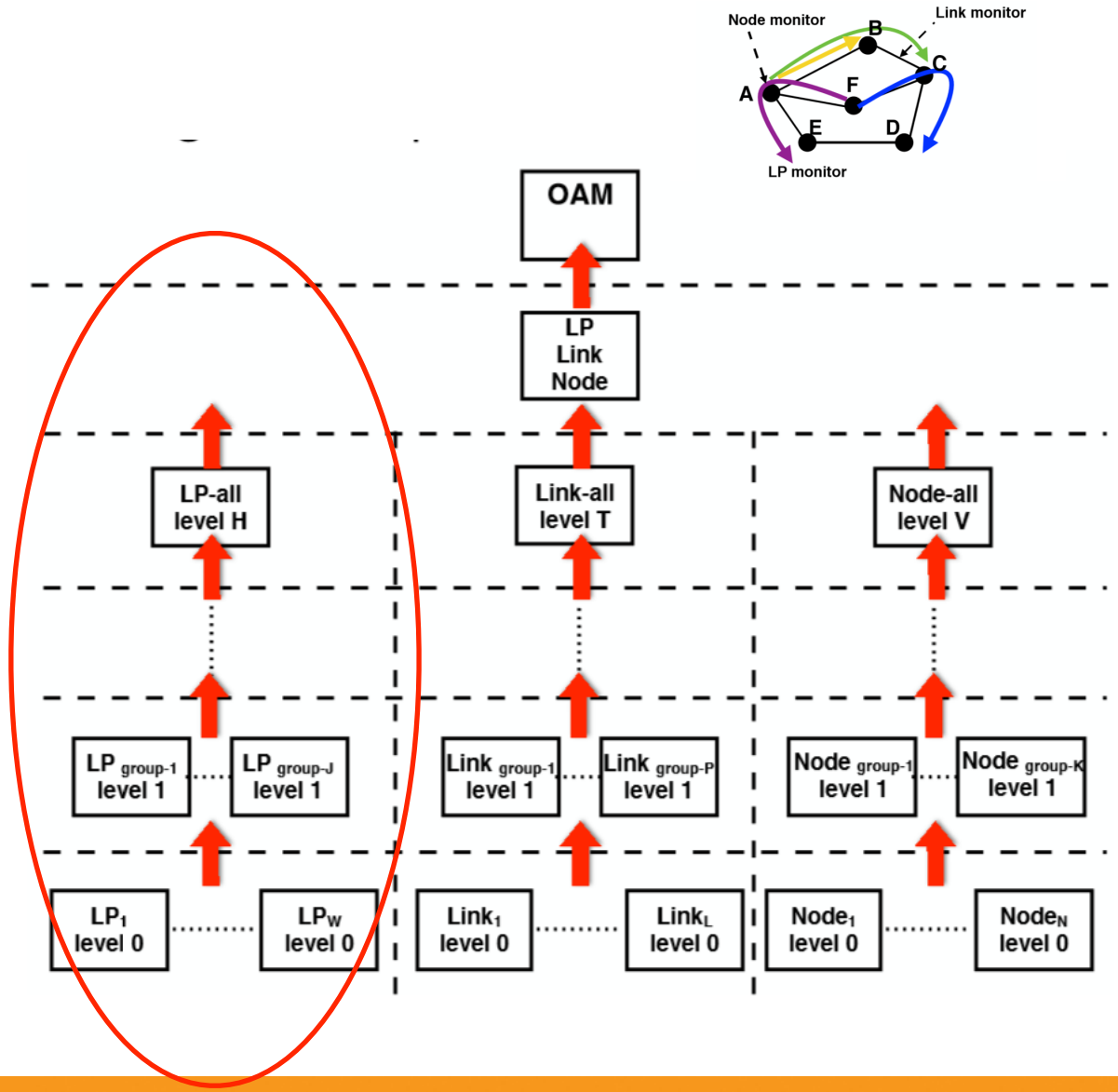
- **LP level 0:** 1 per active lightpath
- **LP group level 1:** each groups group all the lightpaths starting from the same ingress node
- Assuming an amplifier malfunction in link A-B —> alarms generated for the A-B LP and for A-C LP

Hierarchical monitoring architecture: example of responsibility

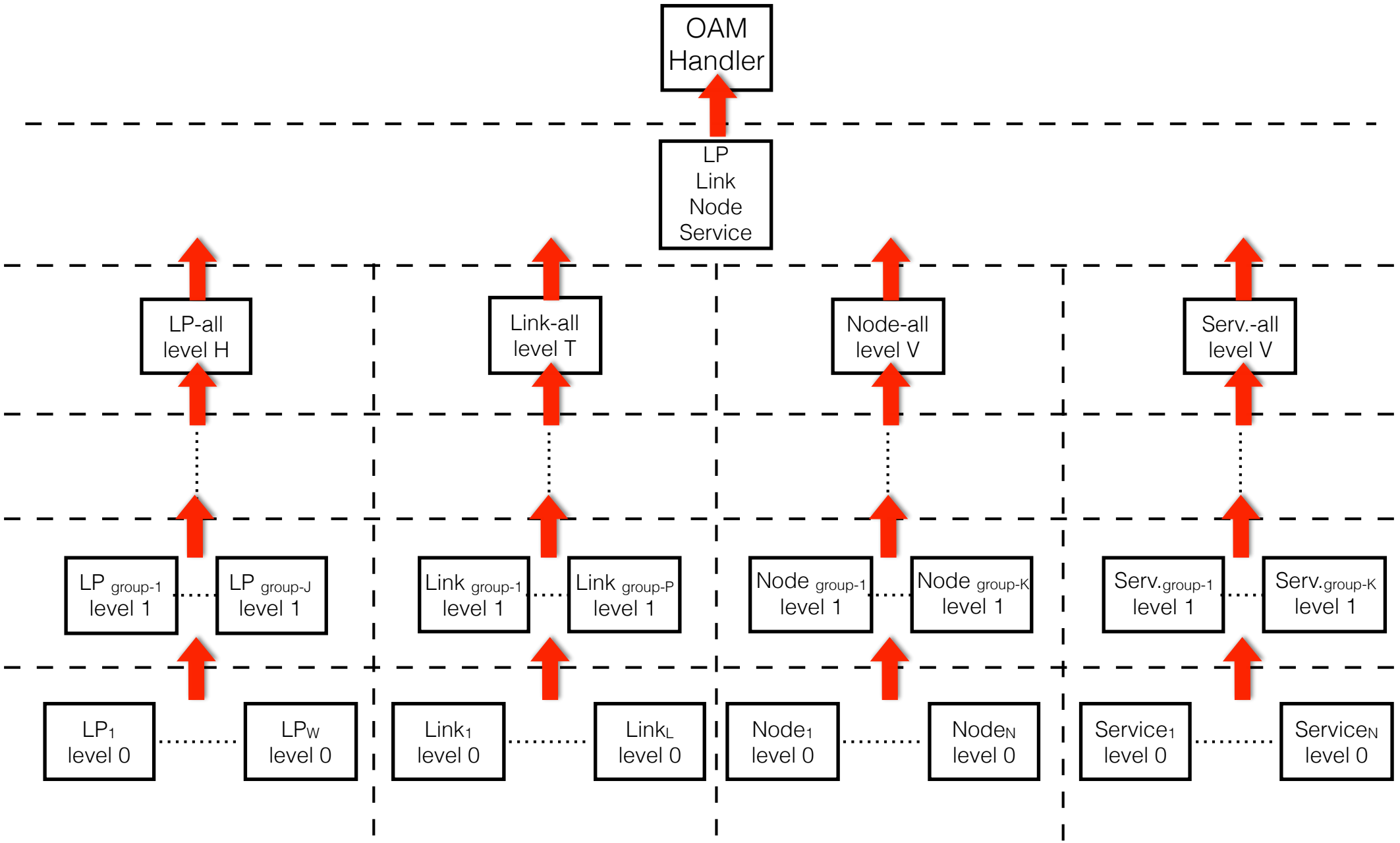


- **LP level 0:** 1 per active lightpath
- **LP group level 1:** each groups group all the lightpaths starting from the same ingress node
- Assuming an amplifier malfunction in link A-B —> alarms generated for the A-B LP and for A-C LP
- Alarms sent to level 1: by correlating this information, a problem can be identified in the segment A-B.

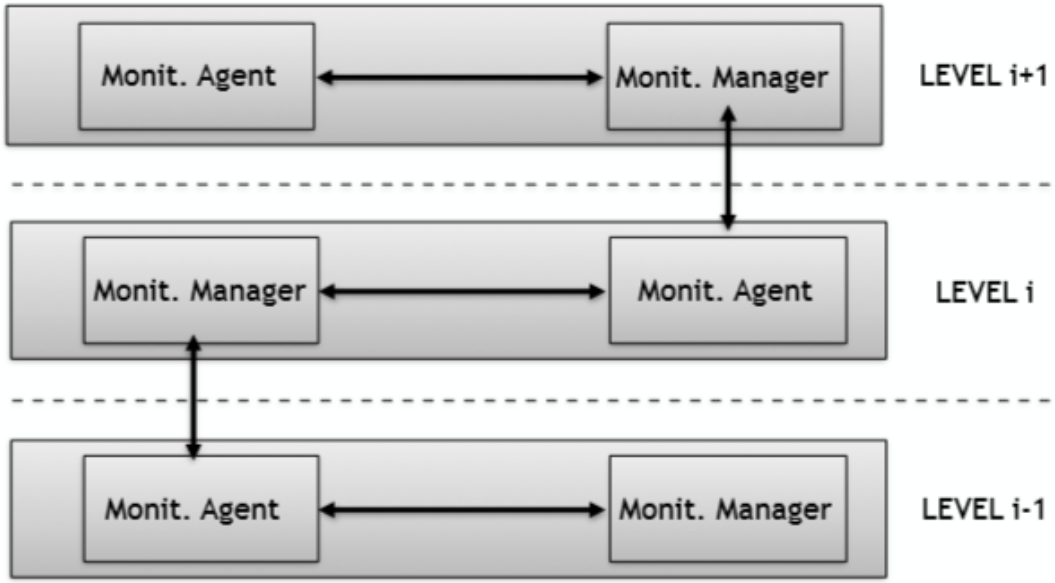
Hierarchical monitoring architecture: example of responsibility



- **LP level 0:** 1 per active lightpath
- **LP group level 1:** each groups group all the lightpaths starting from the same ingress node
- Assuming an amplifier malfunction in link A-B —> alarms generated for the A-B LP and for A-C LP
- Alarms sent to level 1: by correlating this information, a problem can be identified in the segment A-B.
- Then, LP level 2 can group all the lightpaths whose ingress node belongs to a specific region of the network and so on up to a generic level H.


OAM Handler
LP
Link
Node
Service
LP-all level H
Link-all level T
Node-all level V
Serv.-all level V
LP group-1 level 1
LP group-J level 1
Link group-1 level 1
Link group-P level 1
Node group-1 level 1
Node group-K level 1
Serv.group-1 level 1
Serv.group-K level 1
LP1 level 0
LPW level 0
Link1 level 0
LinkL level 0
Node1 level 0
NodeN level 0
Service1 level 0
ServiceN level 0

Monitoring entity



- **Agent** disseminates monitoring information to the upper layer
- Although not shown, the Manager at level *i* is connected to several monitoring entities of the level *i-1*
- **Manager** correlates and processes info coming from agents at the level *i-1*