Multi-Layer NDL

Freek Dijkstra – SARA

with help of:

Jeroen van der Ham – Universiteit van Amsterdam,

Fernando Kuipers – Technische Universiteit Delft,

Bert Andree – Universiteit van Amsterdam,

Paola Grosso – Universiteit van Amsterdam,

Karst Koymans – Universiteit van Amsterdam,

Cees de Laat – Universiteit van Amsterdam

This work was funded by the Gigaport Project

donderdag 28 mei 2009

Multi-layer path finding is a problem. I will not give an answer to that, but I will show that there is a problem, and give you both an information and a data model to use.

Introduction to G.805

http://tinyurl.com/rb7w9u

http://ties.itu.int/ftp/public/itu-t/ahtmpls/readandwrite/doc_exchange/general-documents/G805intro.ppt

"OSI is dead", says OSI

("ITU-T X.200 is deprecated" says ITU SG 13)

The OSI model is of no use anymore (if it ever was).

- Few networks actually work that way
- Highly inflexible (always need more layers!)
- Some features only in one place (security, mux)
- Missing features (OAM)
- Doesn't help to design transport networks

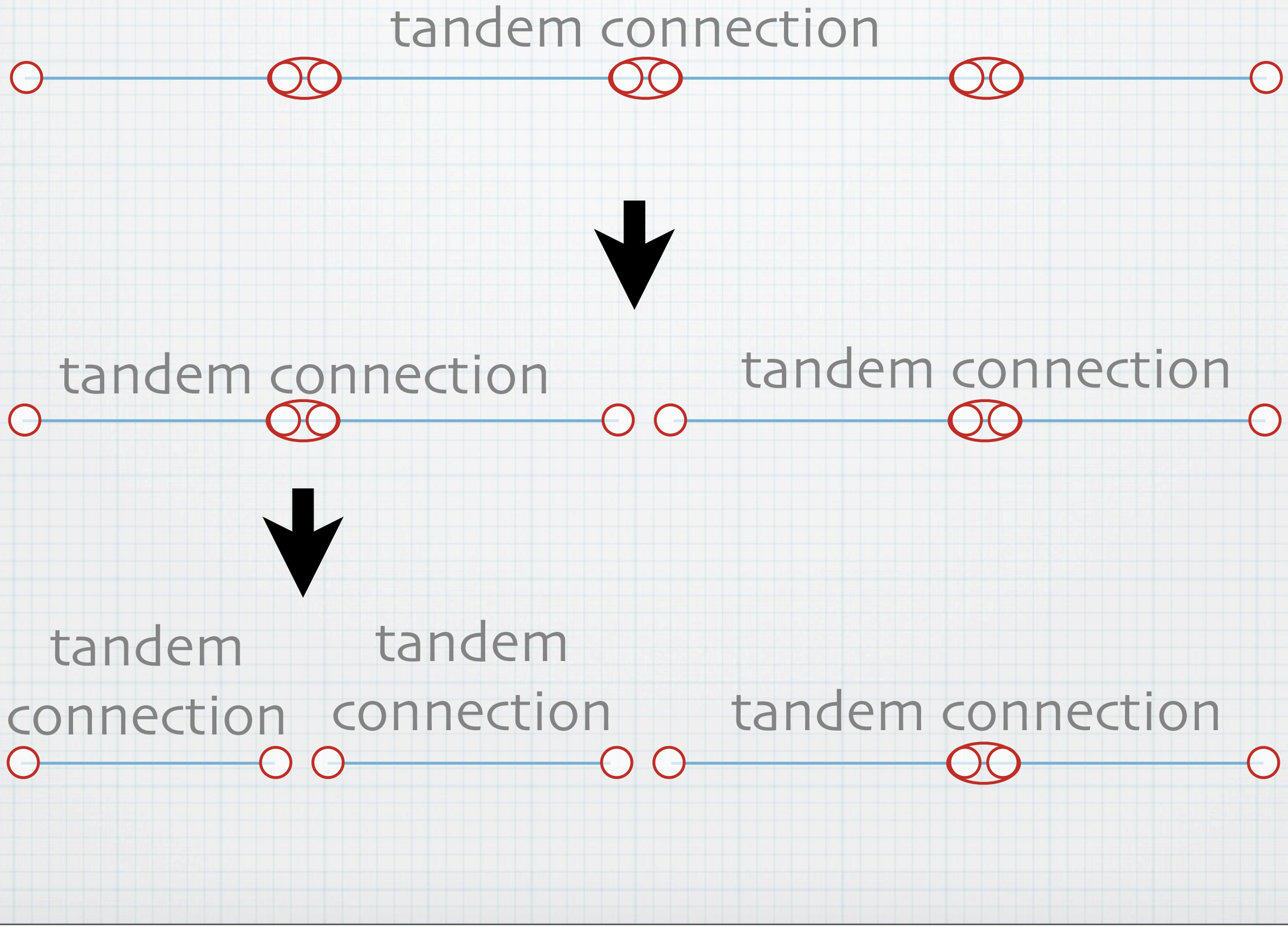
Solution:

New Model: G.805 (connection oriented), G.800 (generic)

Connections

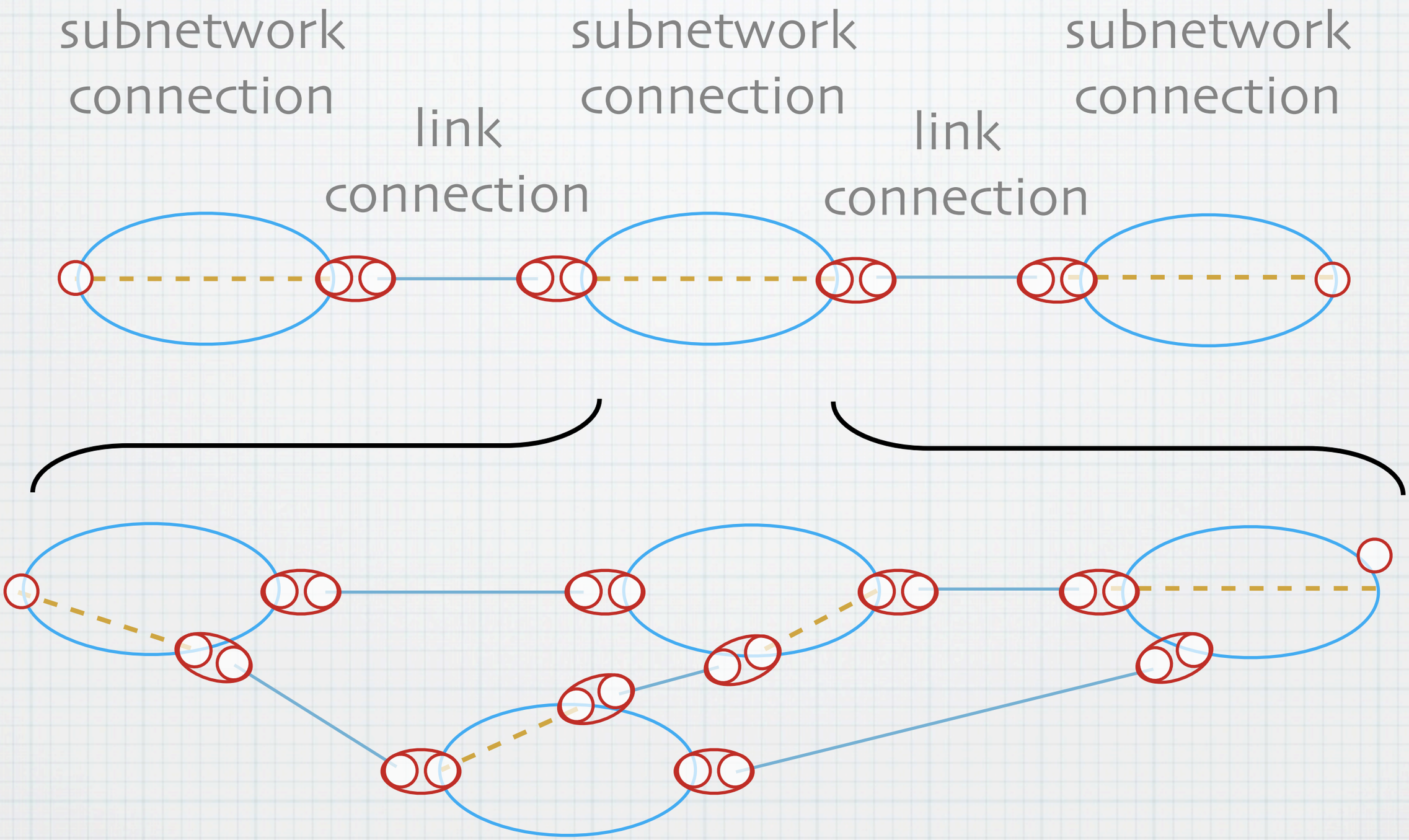
- Link
  Physical Link
- Link Connection
  Logical Link ("Transport function" over a link or lower layer)
- Tandem Connection
  Sequence of Link Connections
- Network Connection
  Tandem Connection between termination points
- Trail
  Terminated Network Connection (retransmission, protection, ...)
- Subnetwork Connection
  Reconfigurable connection in a Subnetwork

Partitioning (Links)



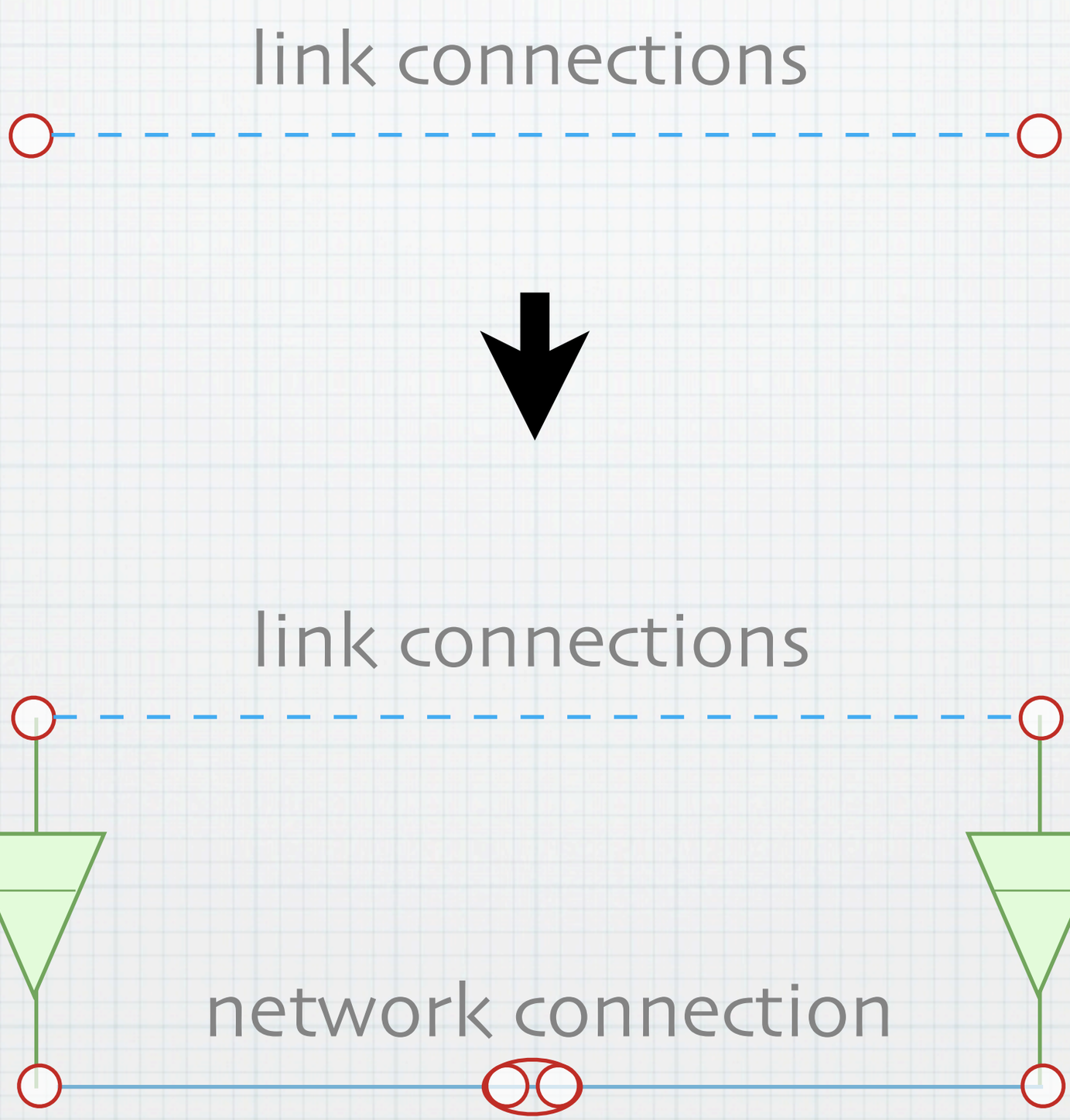
The top tandem connection is a network connection
The bottom tandem connection is a link connection
Recursive partitioning of links. (horizontal)

Partitioning (Subnetworks)



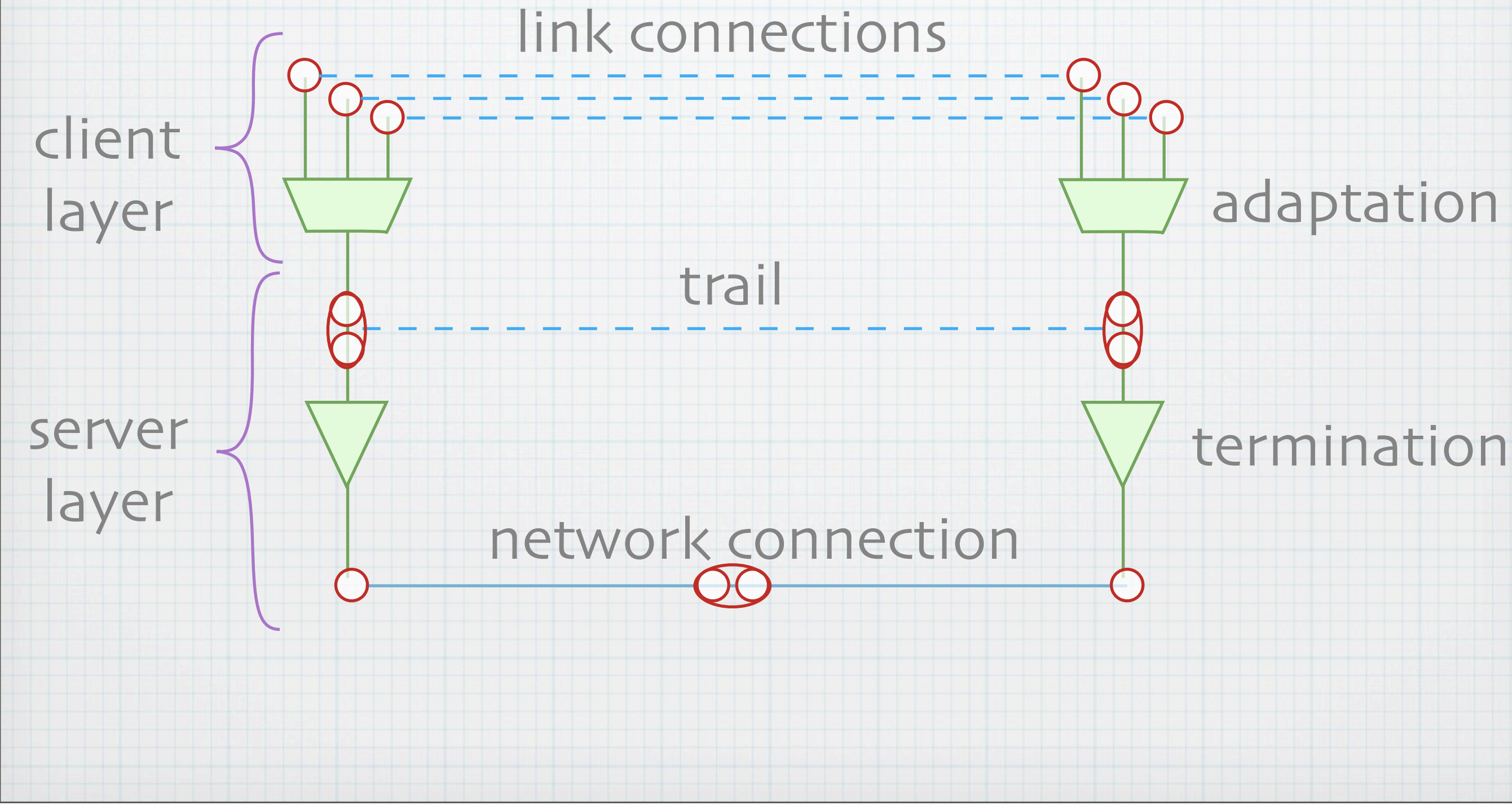
Recursive partition of networks

Layering



Recursive layering (the bottom network connection can in turn be provided by underlying layers)

Layering



donderdag 28 mei 2009

Termination is the process of “adding monitoring information” to the network connection. E.g. error correction, connectivity and continuity check, signal quality monitoring. (I think retransmission too)
Red circles are connection points (logical interfaces)

Also in G.805

(but not in NDL)

- Service Interworking
  Change/terminate technology without adaptation/client layer.
- Protection
  1+1 protection of trails
- OAM
  Monitoring, defect detection, alarms (terminology)
- Access Groups
  Group of similar interfaces at a client
- Traffic Conditioning
  Packet classification
- Unidirectional/Bidirectional
  Bidirectional is shorthand for 2x unidirectional.

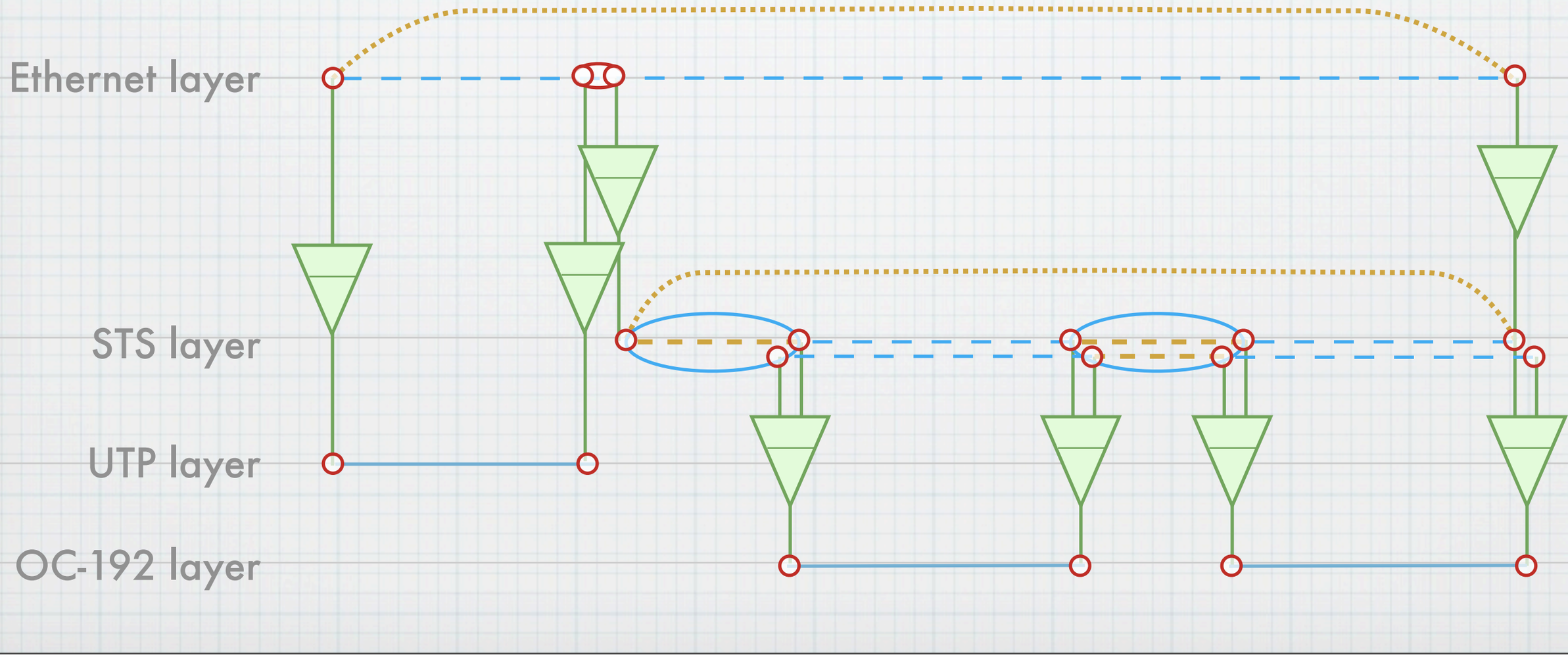
Early Model Development

Computer readable **network description**, which can describe state and capabilities of **multi-layer** networks, using a **technology independent** model.

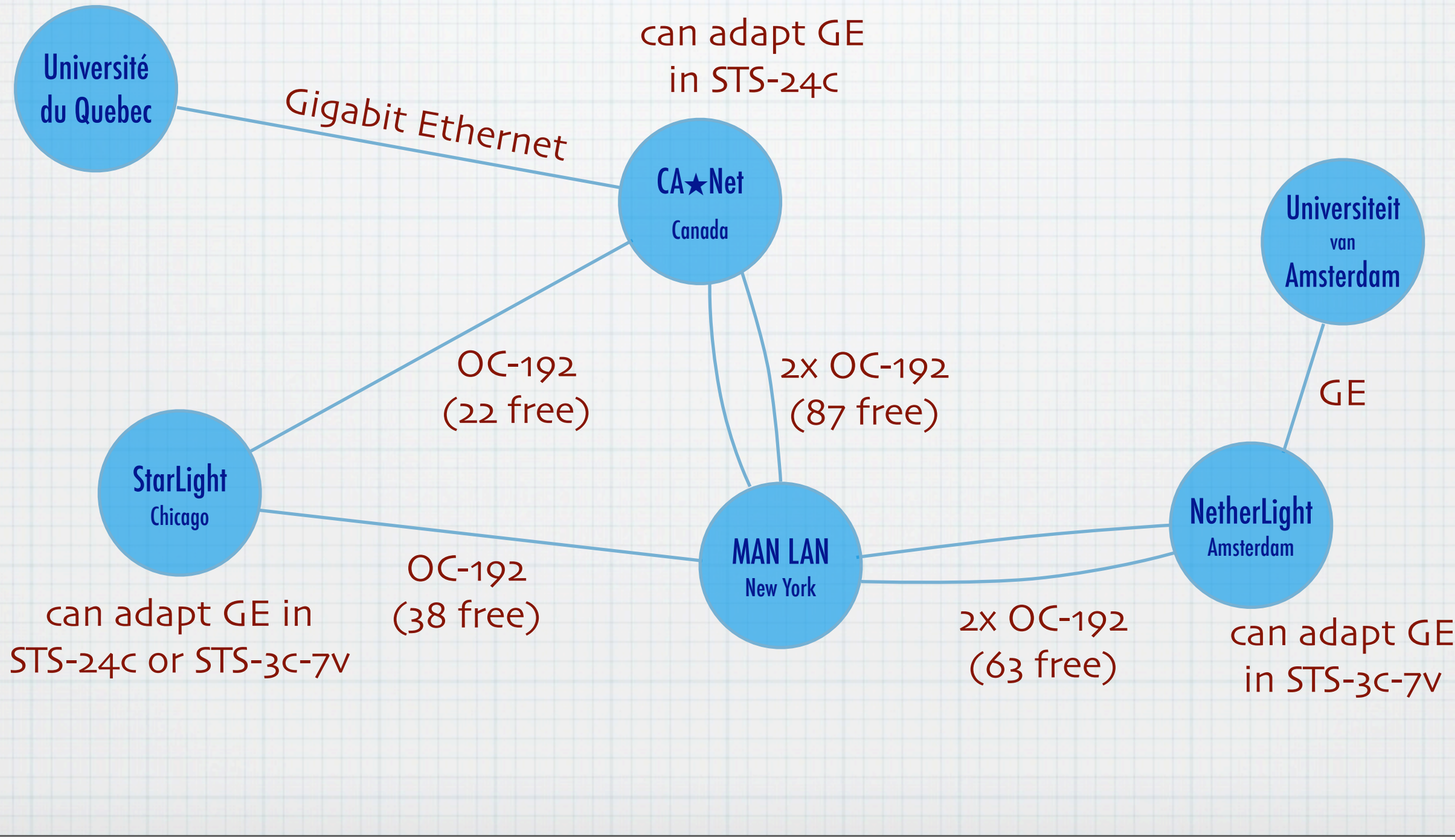
Network Description Language (NDL)

What is NDL?
Based on a model, technology independent. With syntax in RDF.

- G.805 and G.800 allow descriptions of the **state** of a network
- No model exist to describe how to **change that state**, and who may do so

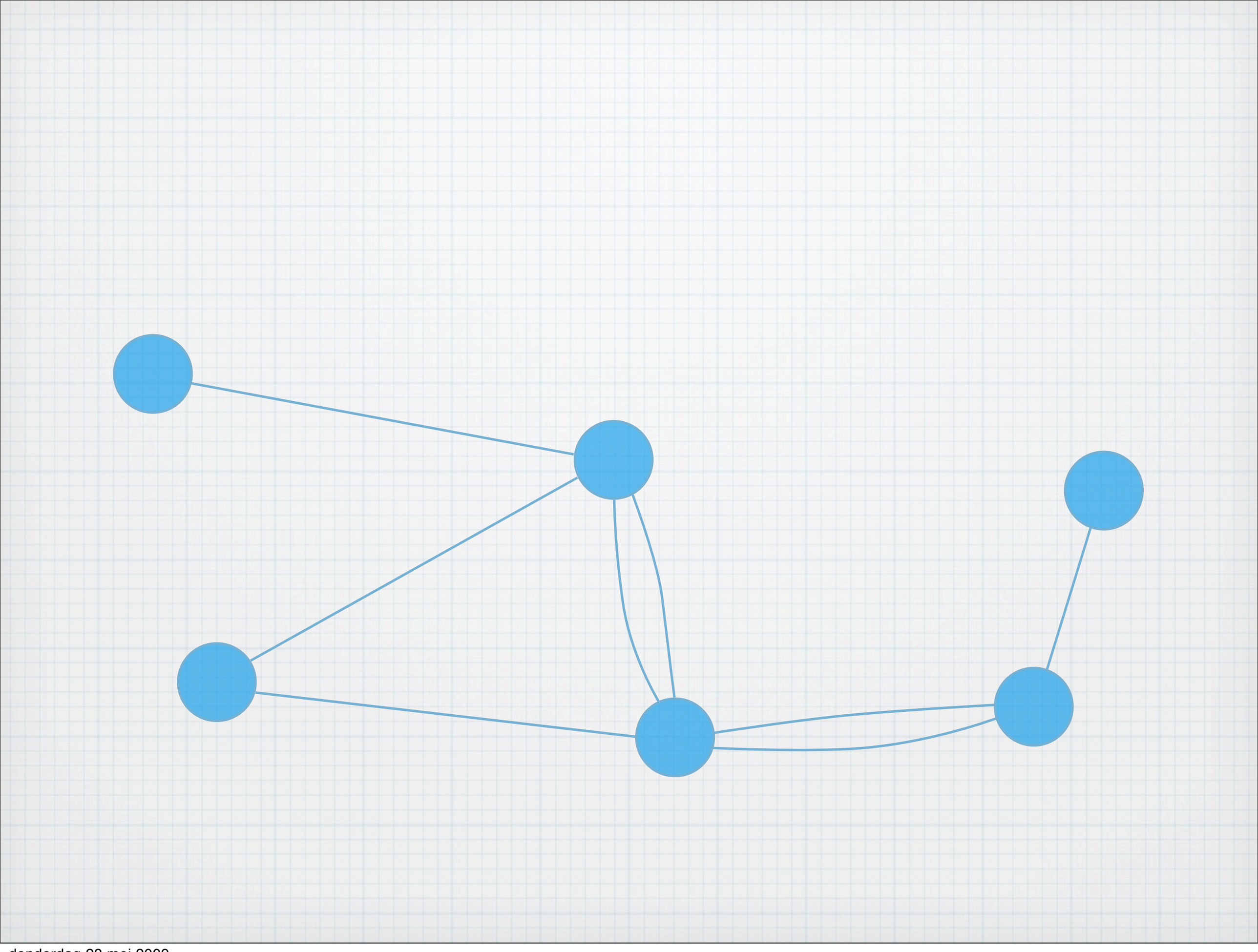


There are no models to describe state changes (= capabilities).
GMPLS can describe capabilities, but does not have a formal model.

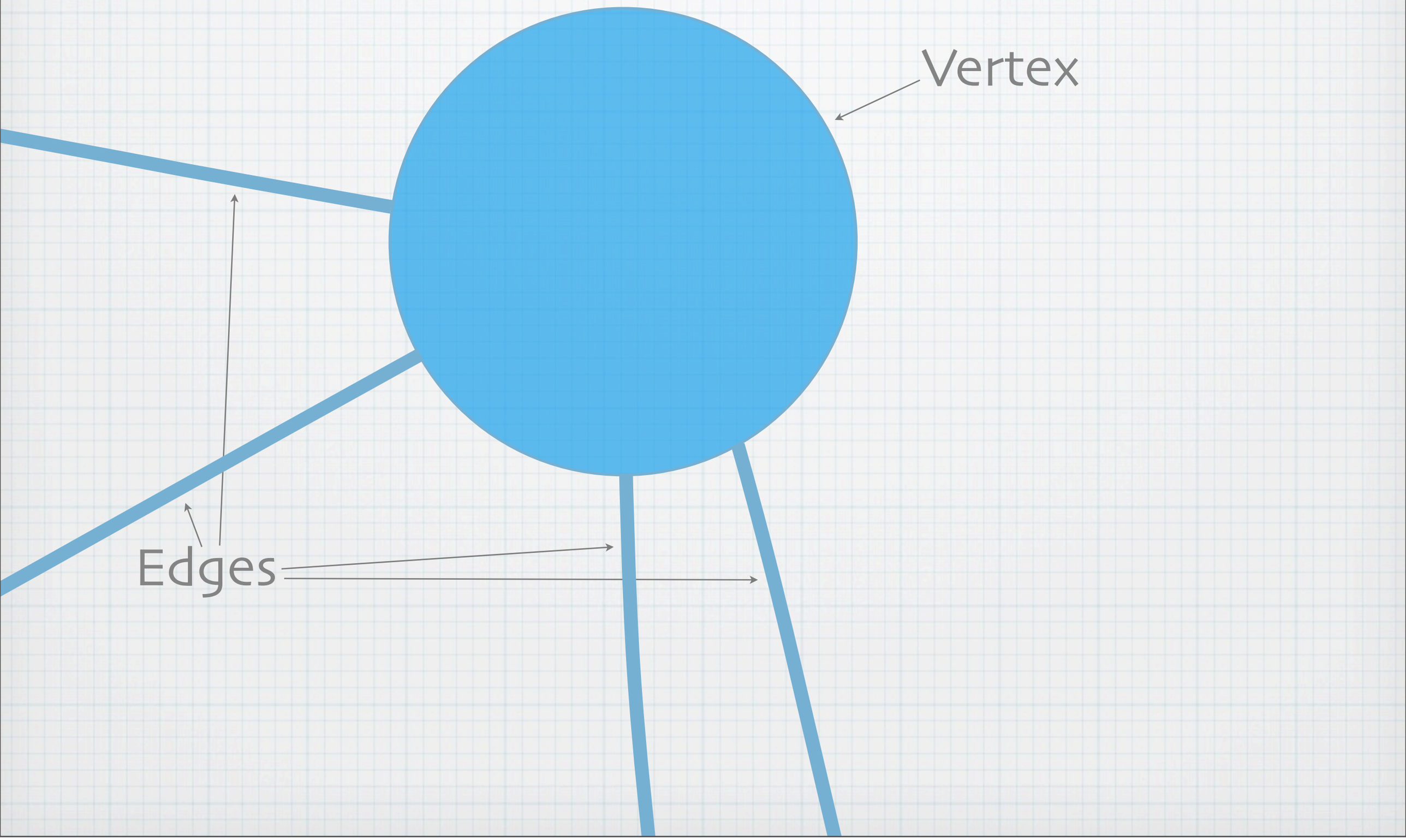


donderdag 28 mei 2009

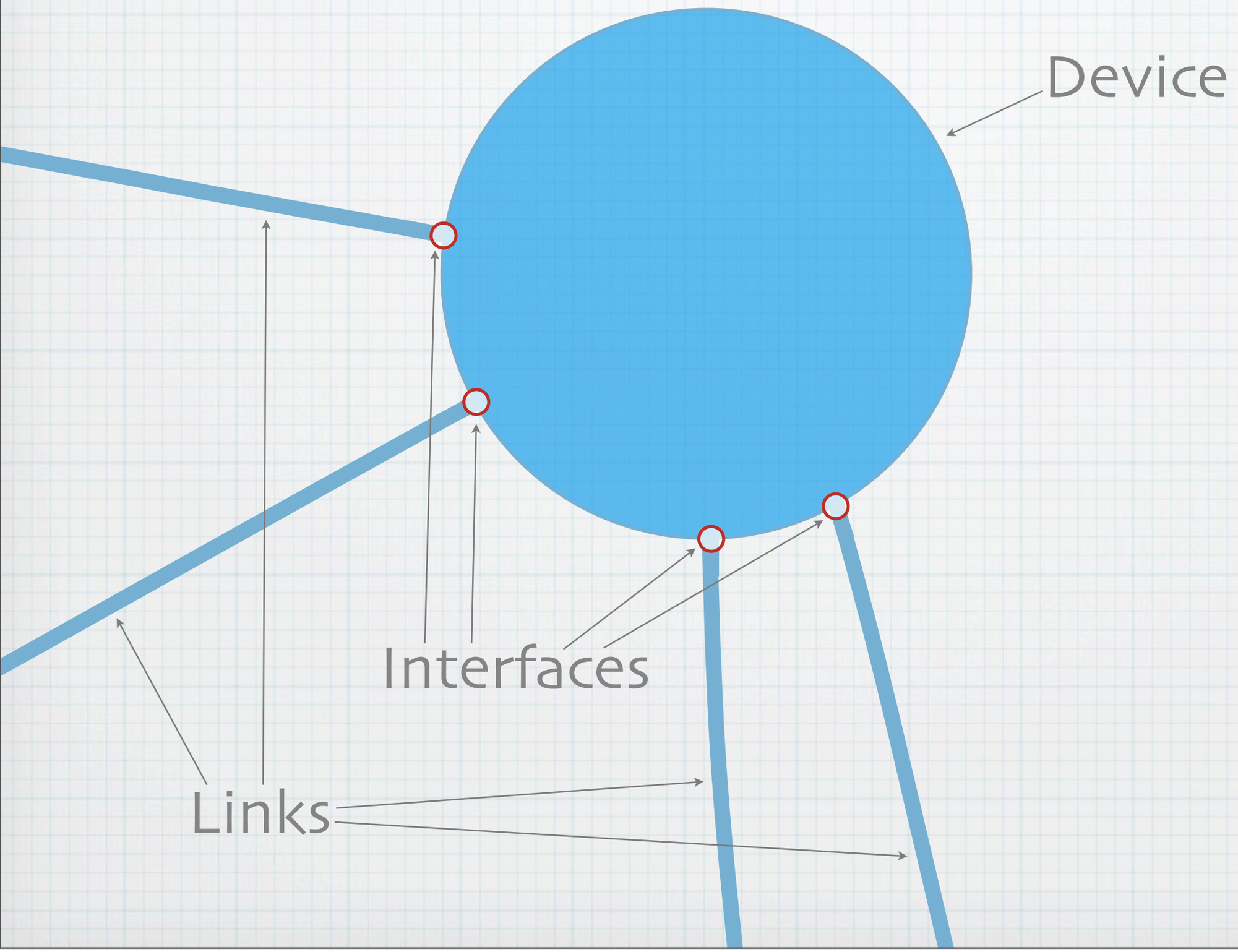
You can not just consider one layer in this example: Quebec and Amsterdam do not even know about SDH. MAN LAN does not understand Ethernet. Adaptations are important.



When we think about network models, we think: graphs .
Simple graph. Not very accurate. Let’s zoom in on vertex.



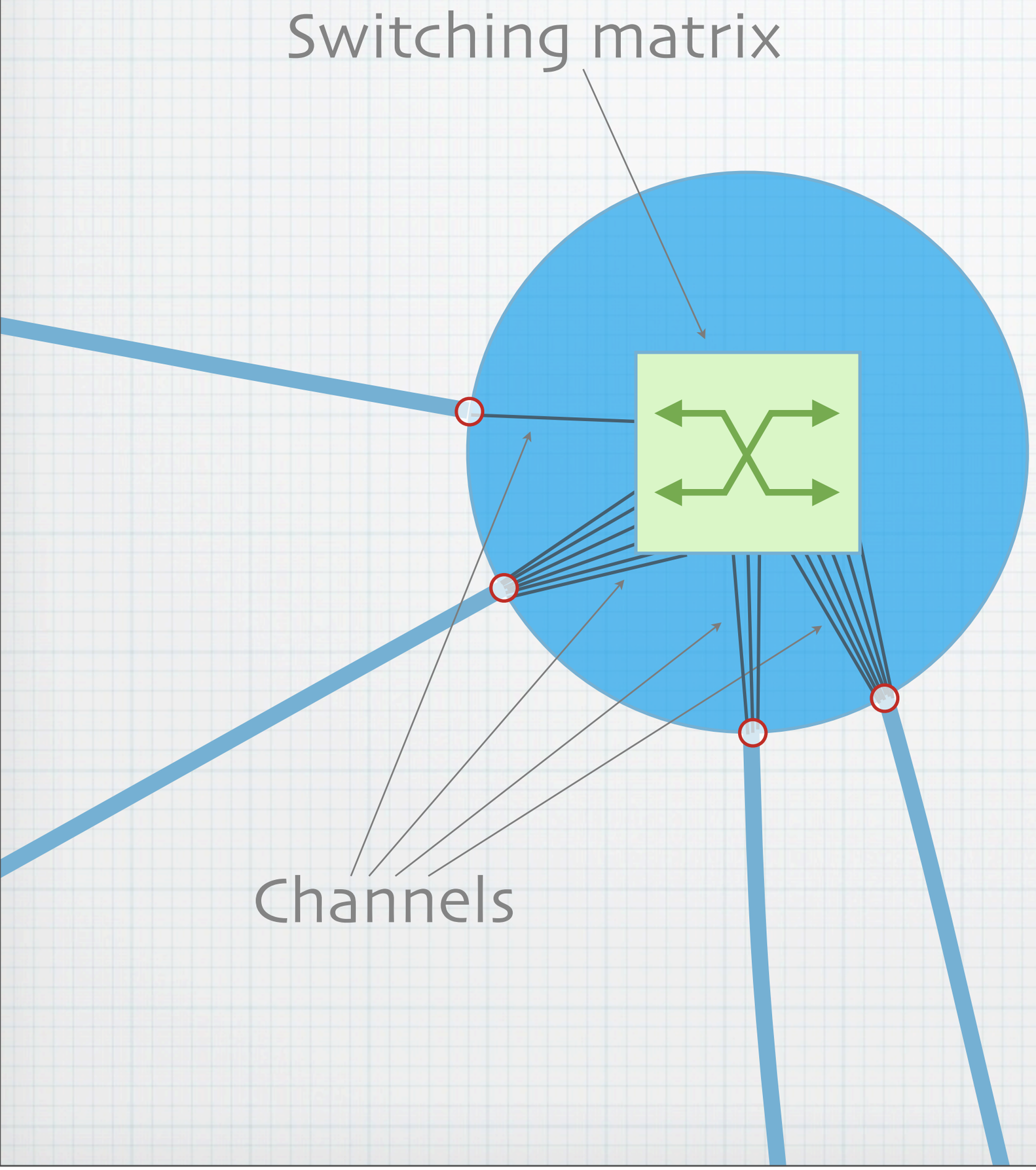
There are vertices and edges. The incident of an edge on a vertex is an “interface”



donderdag 28 mei 2009

The incident of the edge on the vertex retain properties, even when disconnected. The interface is still there (with properties like capacity, wavelength, type). We want to model that too.

So we now already want three classes: Device, Interface, Link. You'll recognize these from NDL.



A devices switches data based on:

- The source interface
- One or more labels

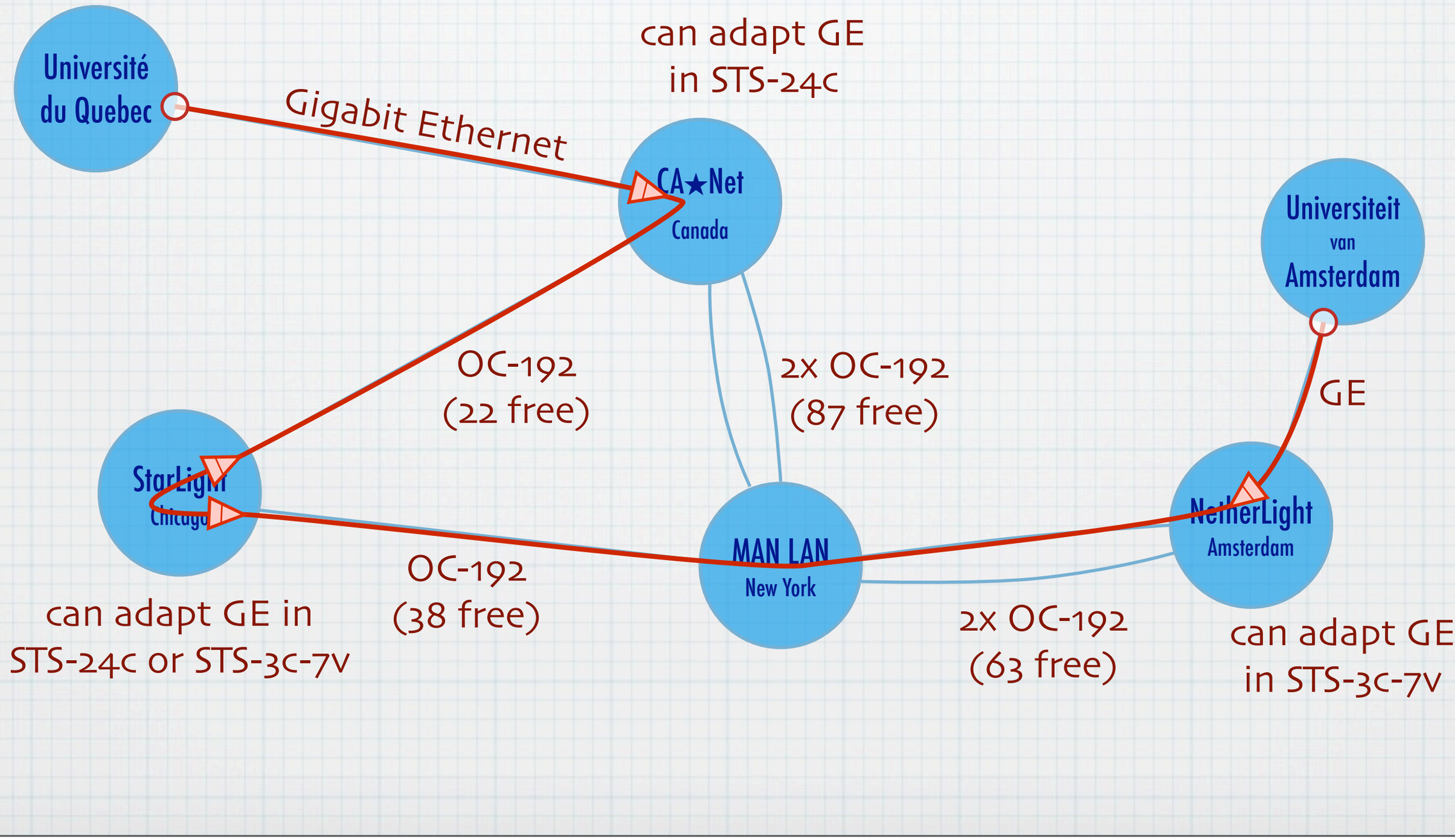
Example label types:

- Ethernet VLAN
- SONET STS Channel
- Wavelength (λ)

For example, all data from channel 31 of interface 2 is forwarded to channel 28 of interface 4.

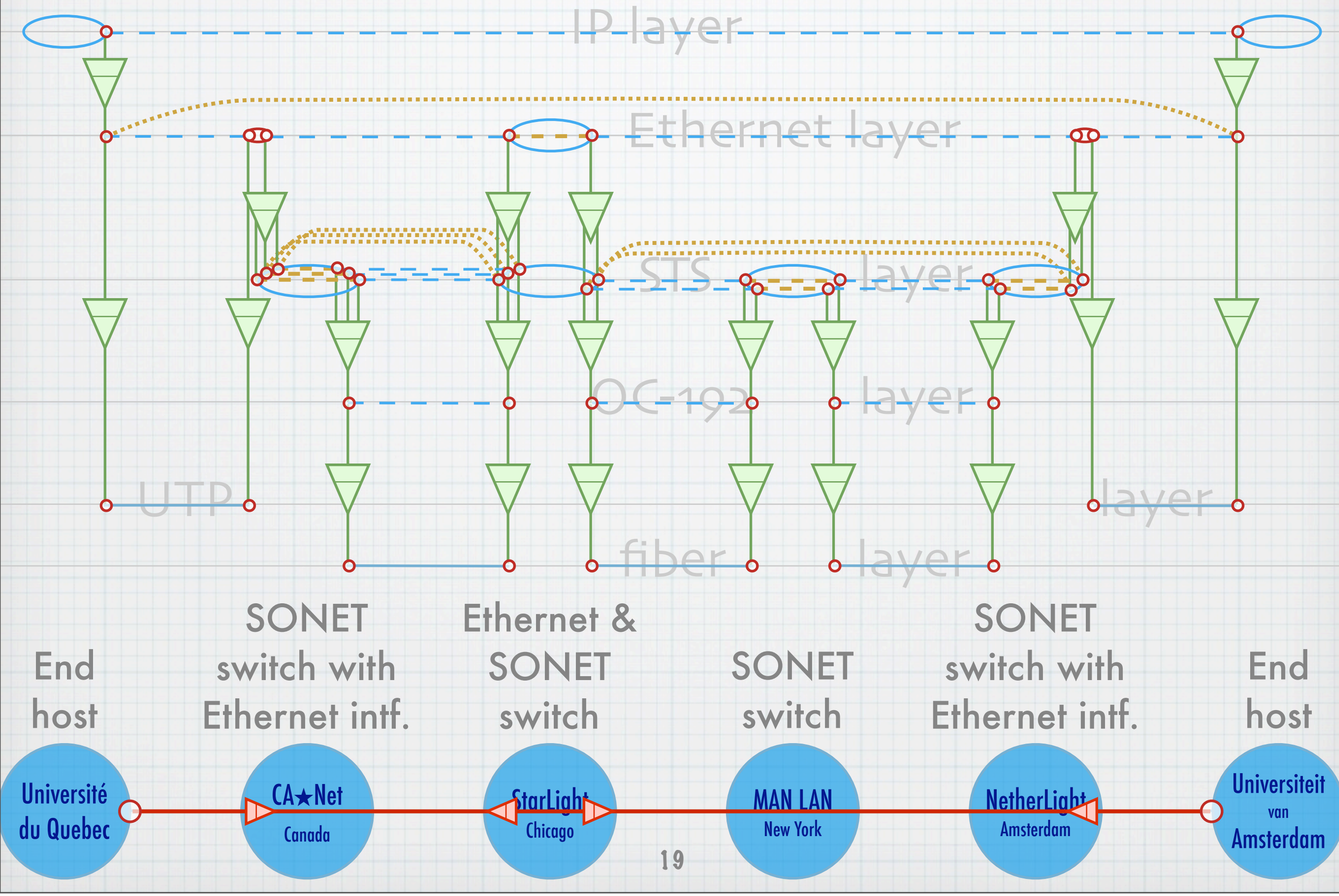
donderdag 28 mei 2009

Core of a device is a switching matrix. Typically, every connected link is split (demuxed) into multiple channels, each of which is connected to the switching matrix.
Any property that is used to make a switching decision is a label type.
GMPLS concept.



donderdag 28 mei 2009

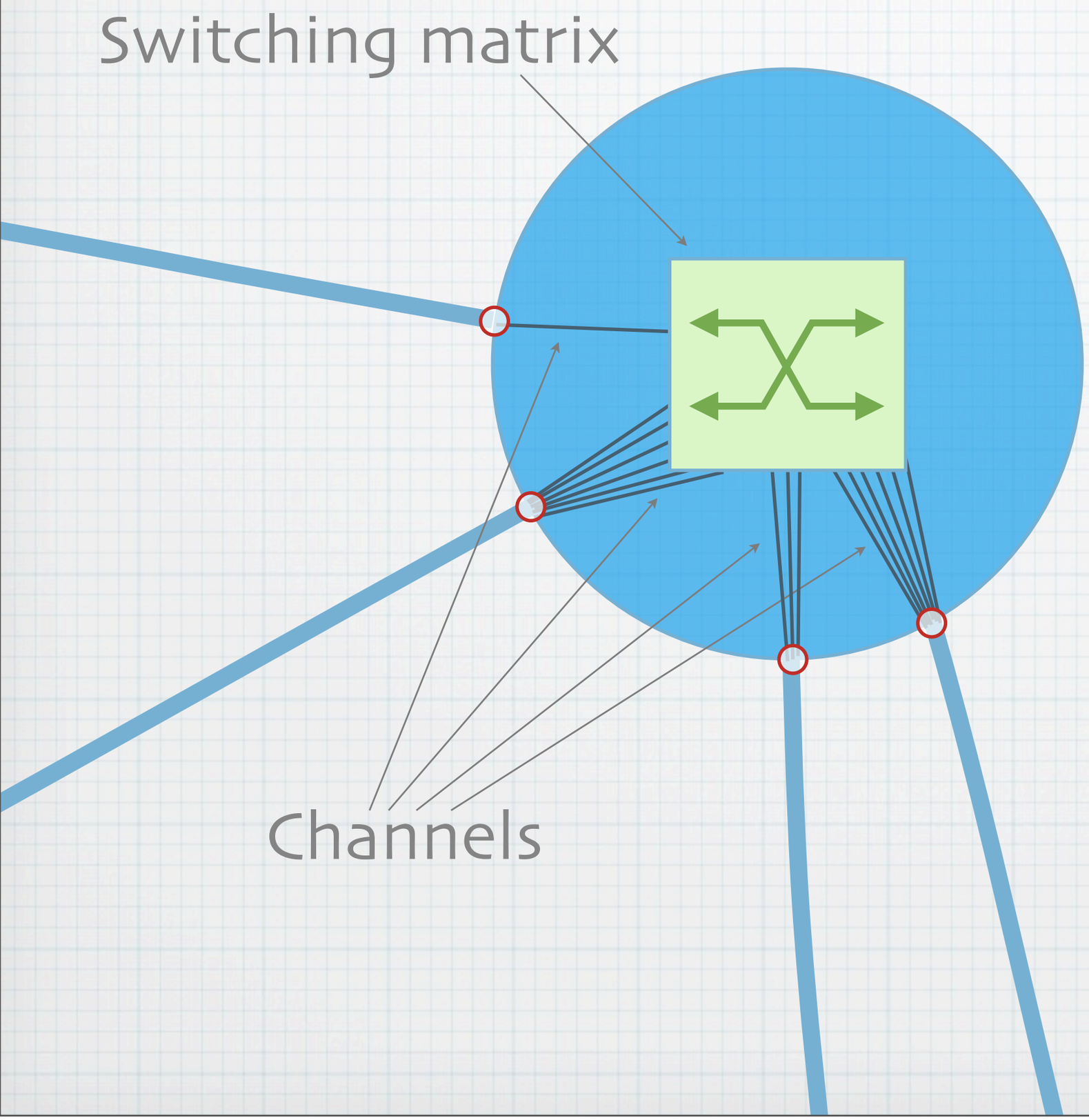
Examine G.805. Let's go back to our second attempt and examine the adaptation incompatibilities.



donderdag 28 mei 2009

We use G.805 functional elements for our information model.
subnetwork, connection points (few per interface), adaptation (+termination) functions, links, link connections, subnetwork connections (configuration), network connections.
In addition, we use the label concept of GMPLS.

GMPLS: label concept



A devices switches data based on:

- The source interface
- One or more labels

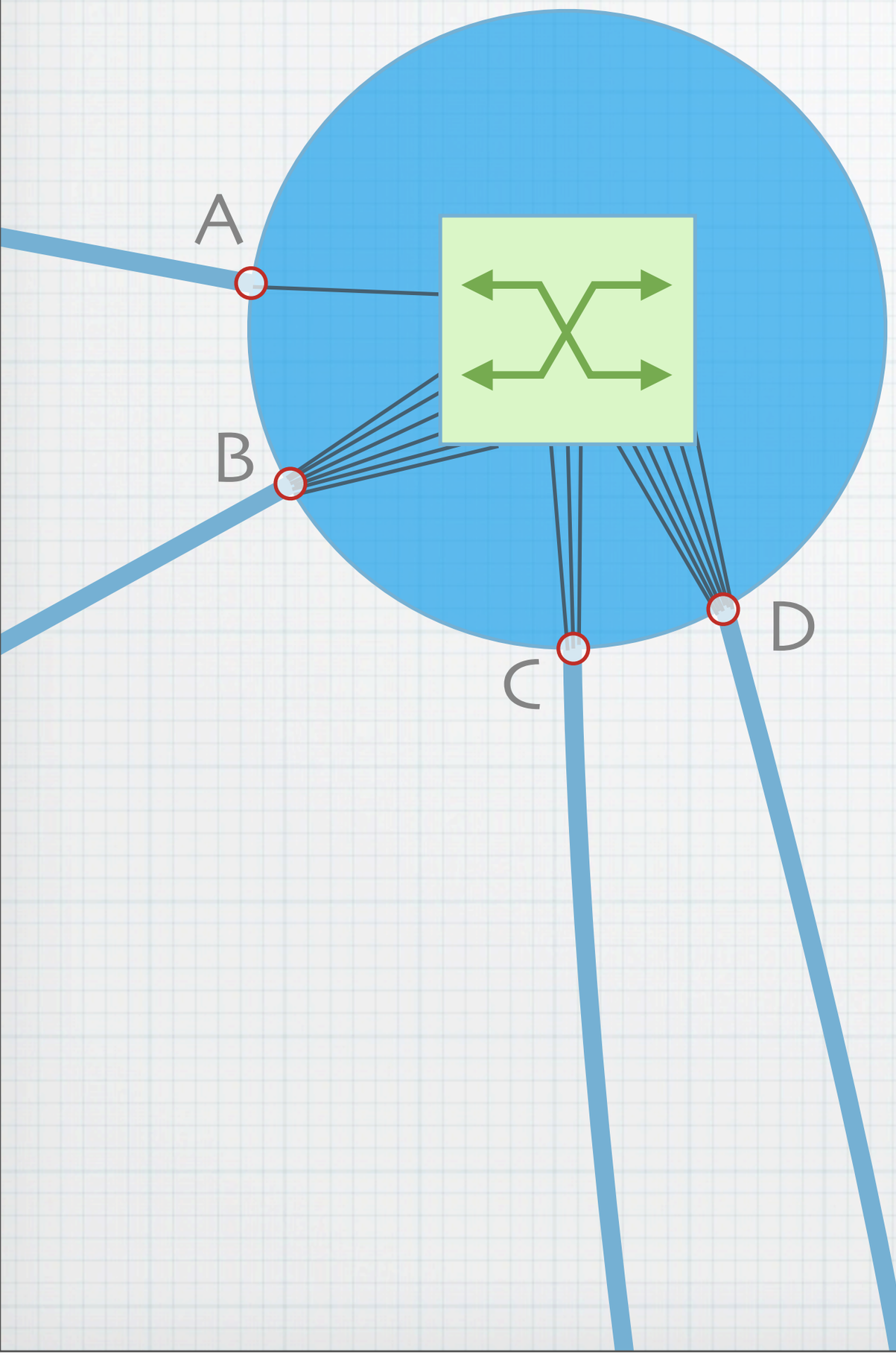
Example label types:

- Ethernet VLAN
- SONET STS Channel
- Wavelength (λ)

For example, all data from channel 31 of interface 2 is forwarded to channel 28 of interface 4.

donderdag 28 mei 2009

In addition to G.805, we use the label concept of GMPLS. The recent G.800 also contains this concept.
Core of a device is a switching matrix. Typically, every connected link is split (demuxed) into multiple channels, each of which is connected to the switching matrix.
Any property that is used to make a switching decision is a label type.



SONET Device

- Switches based on STS channels

Four Interfaces:

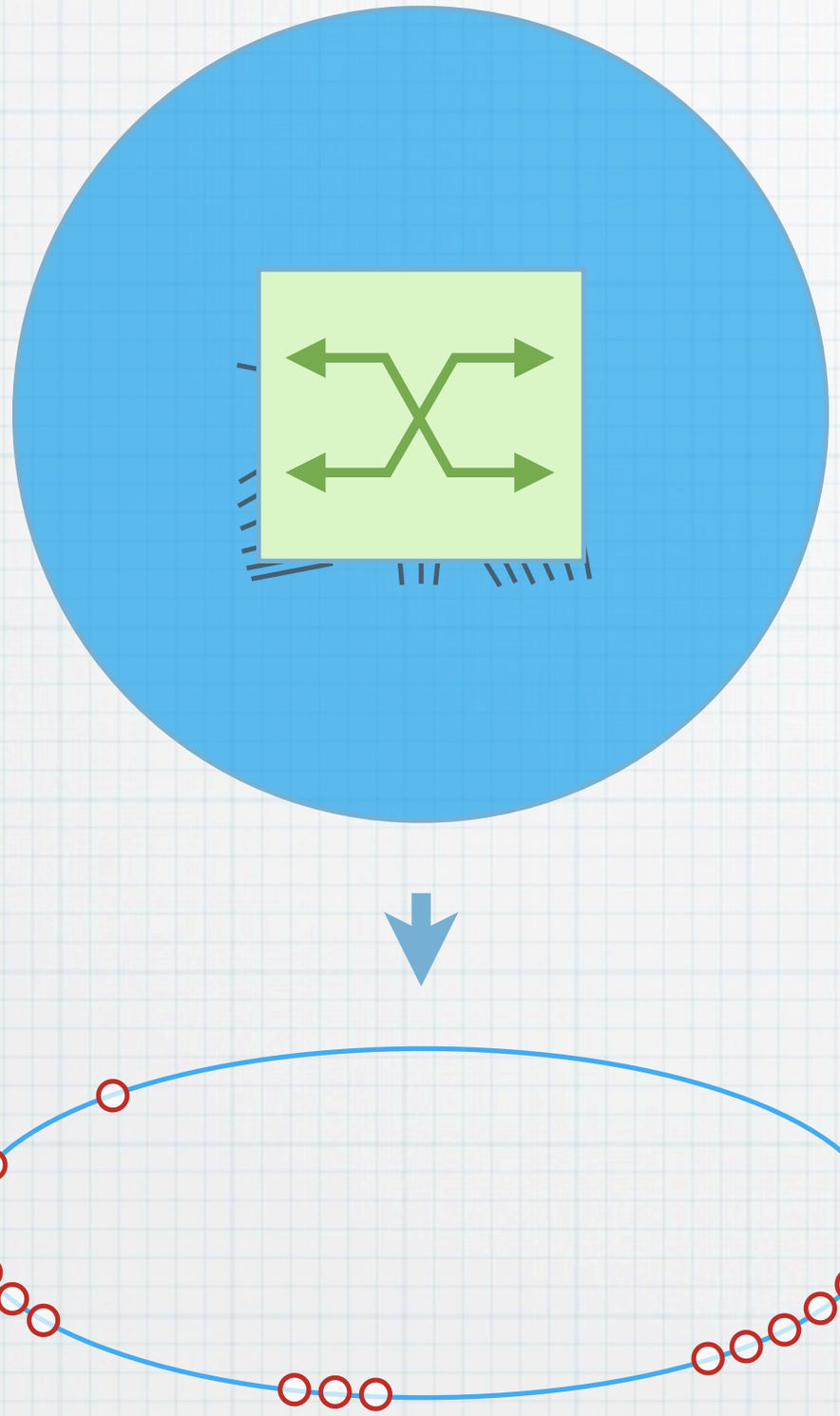
A. Ethernet interface (over UTP) (adapts in STS-3c-7v)

B. OC-192 interface

C. OC-48 interface (over fiber)

D. OC-192 interface (over DWDM at 1552.52nm over fiber)

How does a device work internally? Core of a device is a switching matrix. Typically, every connected link is split (demuxed) into multiple channels, each of which is connected to the switching matrix.



Subnetwork with 433 connection points

Device

switchingCapability → LabelType

Can switch, but not change label.
E.g. from STS 31 of interface 2 to STS 31 of interface 4.

swappingCapability → LabelType

Can change label.
E.g. from STS 31 of interface 2 to STS 28 of interface 4.

Interface

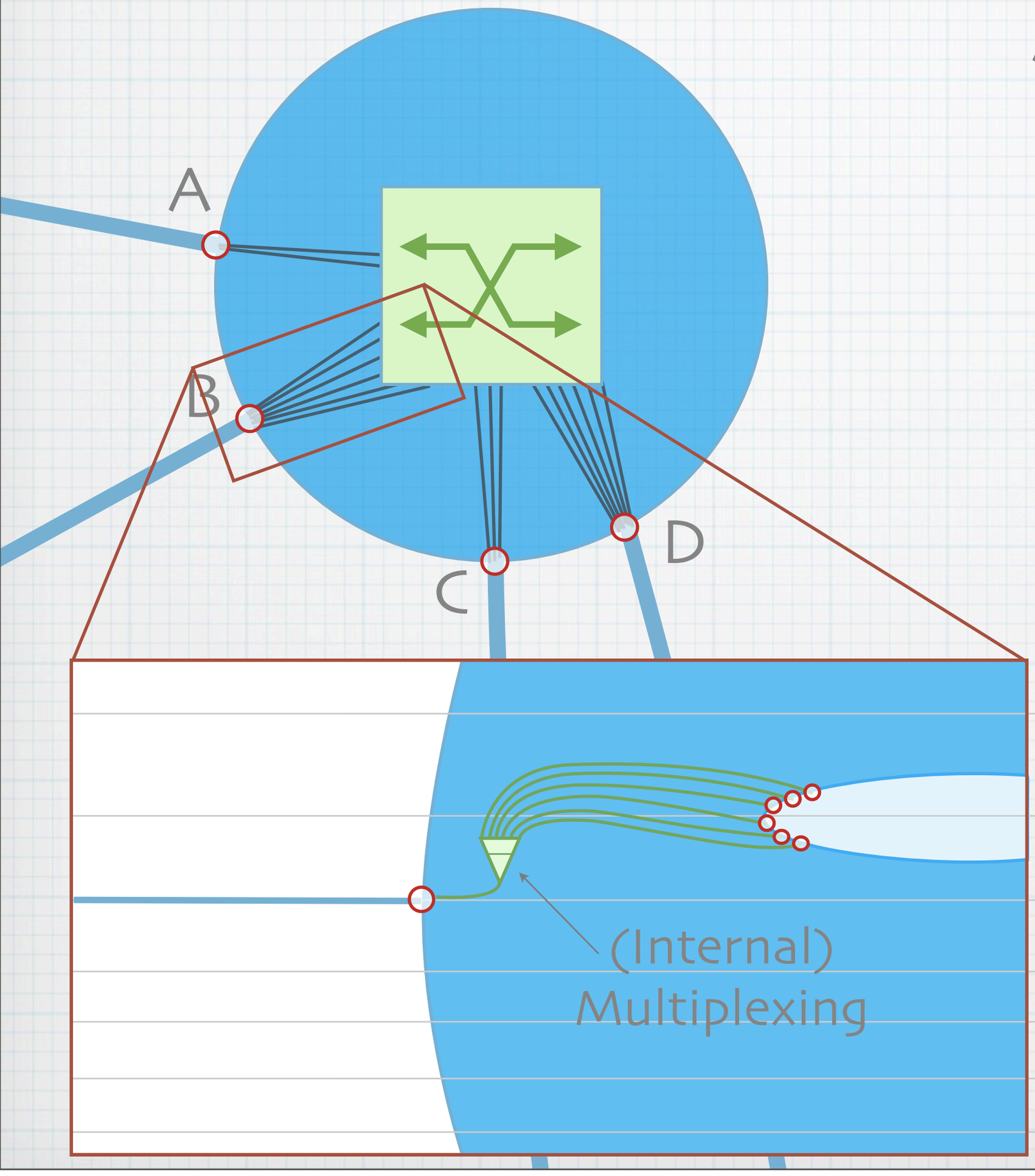
hasLabel → Label

Channel Identifier

switchedTo → Interface

A subnetwork connection

mapping device → subnetwork
Explain switching and swapping!



Interfaces:

A. Ethernet interface (over UTP) (Adapts in STS-3c-7v)

B. OC-192 interface

C. OC-48 interface (over fiber)

D. OC-192 interface (over DWDM at 1552.52nm over fiber)

Ethernet layer

STS layer

Optical Carrier layer

Lambda layer

DWDM layer

UTP layer

fiber layer

Examples of mapping interface → functional elements (connection points and adaptation functions)

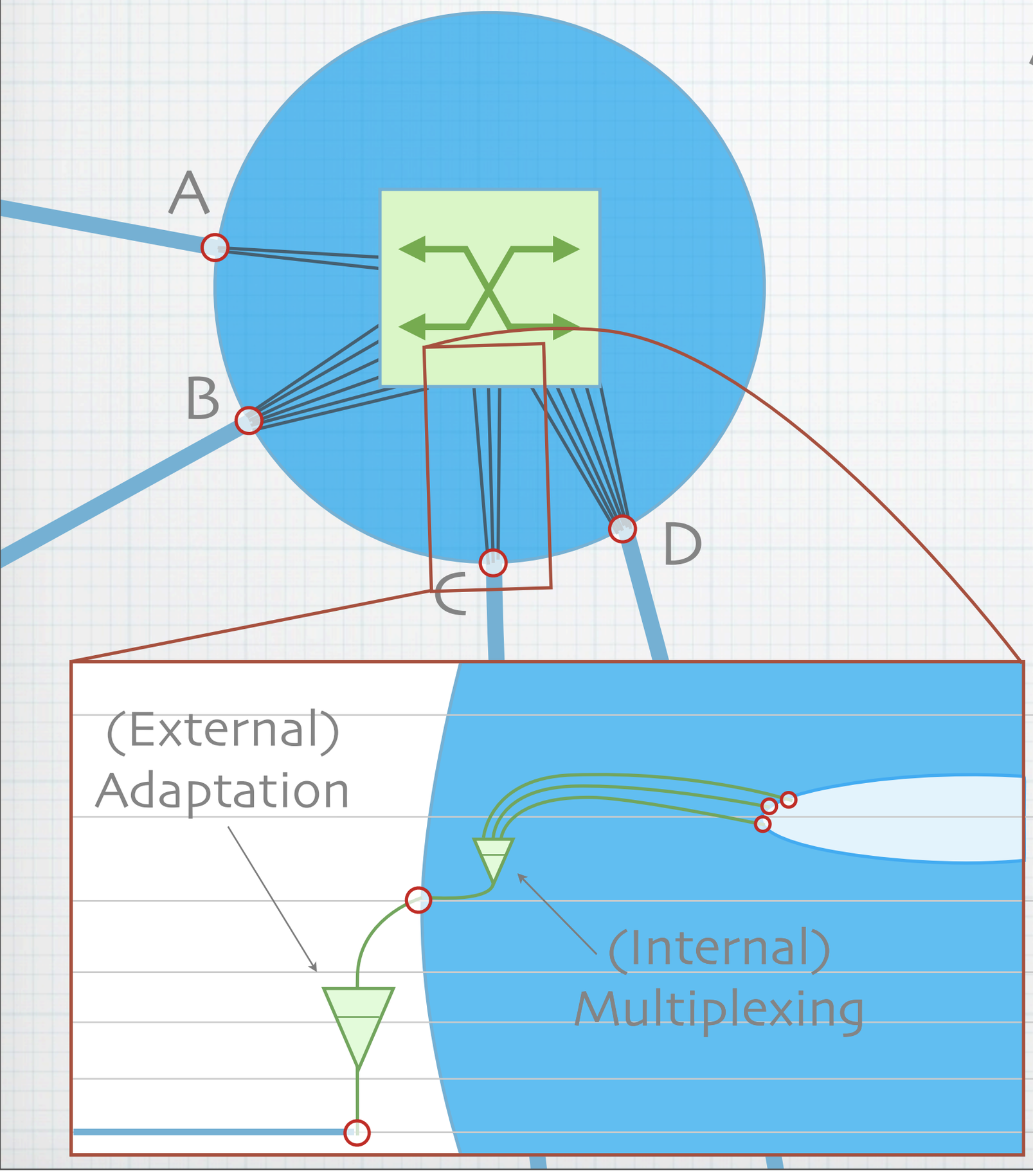
Interfaces:

A. Ethernet interface (over UTP) (Adapts in STS-3c-7v)

B. OC-192 interface

C. OC-48 interface (over fiber)

D. OC-192 interface (over DWDM at 1552.52nm over fiber)



Ethernet layer

STS layer

Optical Carrier layer

Lambda layer

DWDM layer

UTP layer

fiber layer

Examples of mapping interface → functional elements (connection points and adaptation functions)

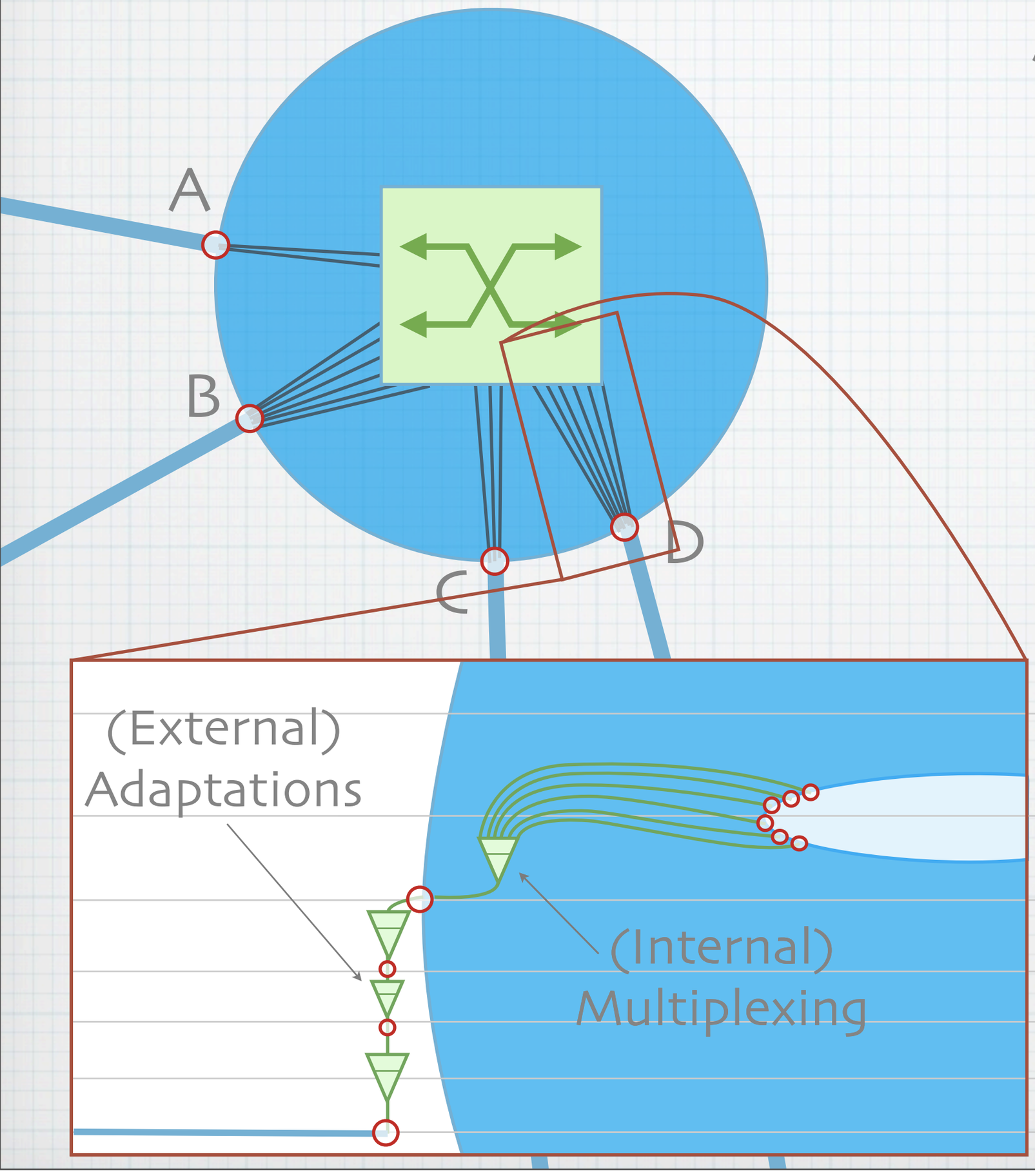
Interfaces:

A. Ethernet interface (over UTP) (Adapts in STS-3c-7v)

B. OC-192 interface

C. OC-48 interface (over fiber)

D. OC-192 interface (over DWDM at 1552.52nm over fiber)



Ethernet layer

STS layer

Optical Carrier layer

Lambda layer

DWDM layer

UTP layer

fiber layer

Examples of mapping interface → functional elements (connection points and adaptation functions)

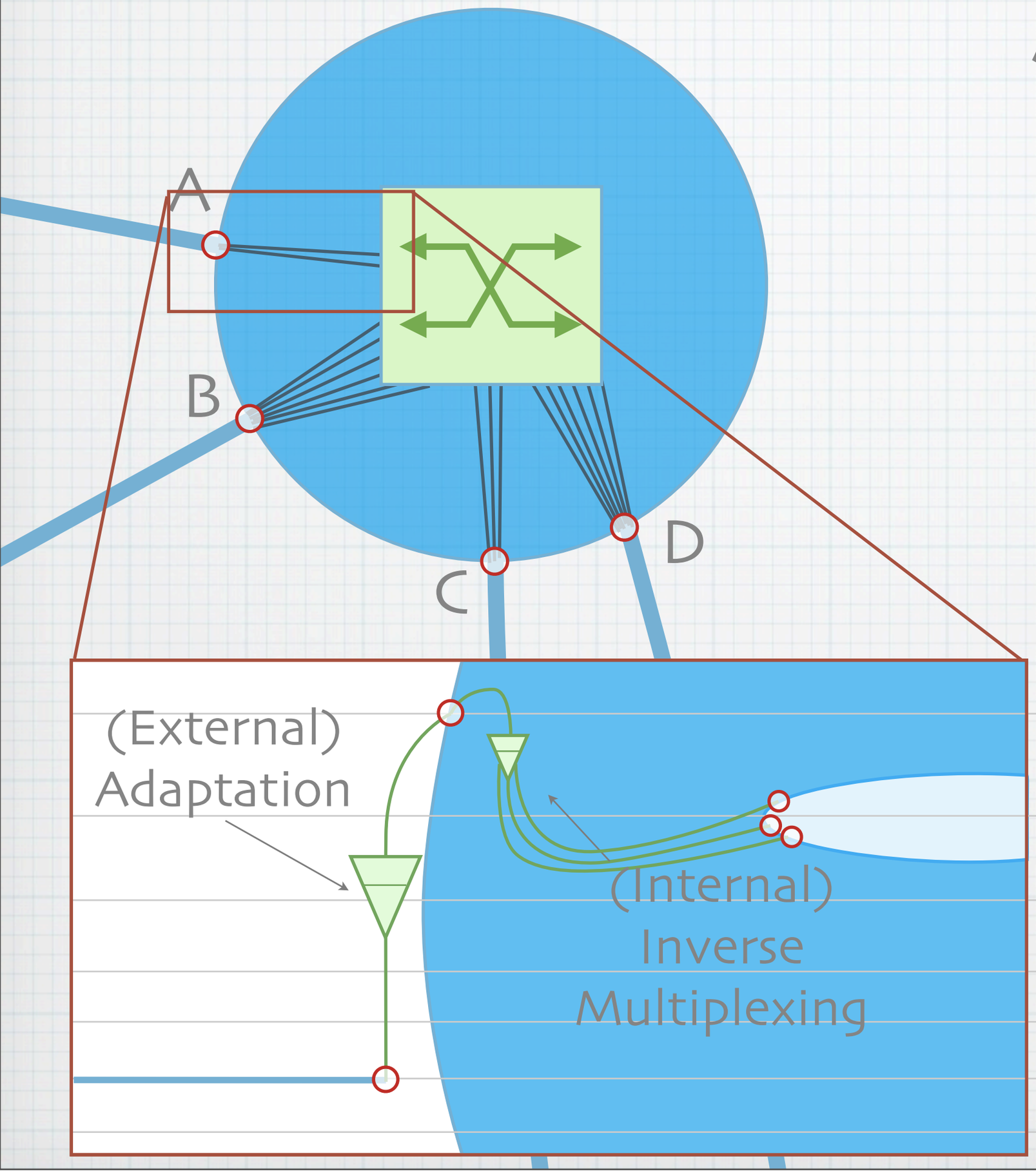
Interfaces:

A. Ethernet interface (over UTP) (Adapts in STS-3c-7v)

B. OC-192 interface

C. OC-48 interface (over fiber)

D. OC-192 interface (over DWDM at 1552.52nm over fiber)



Ethernet layer

STS layer

Optical Carrier layer

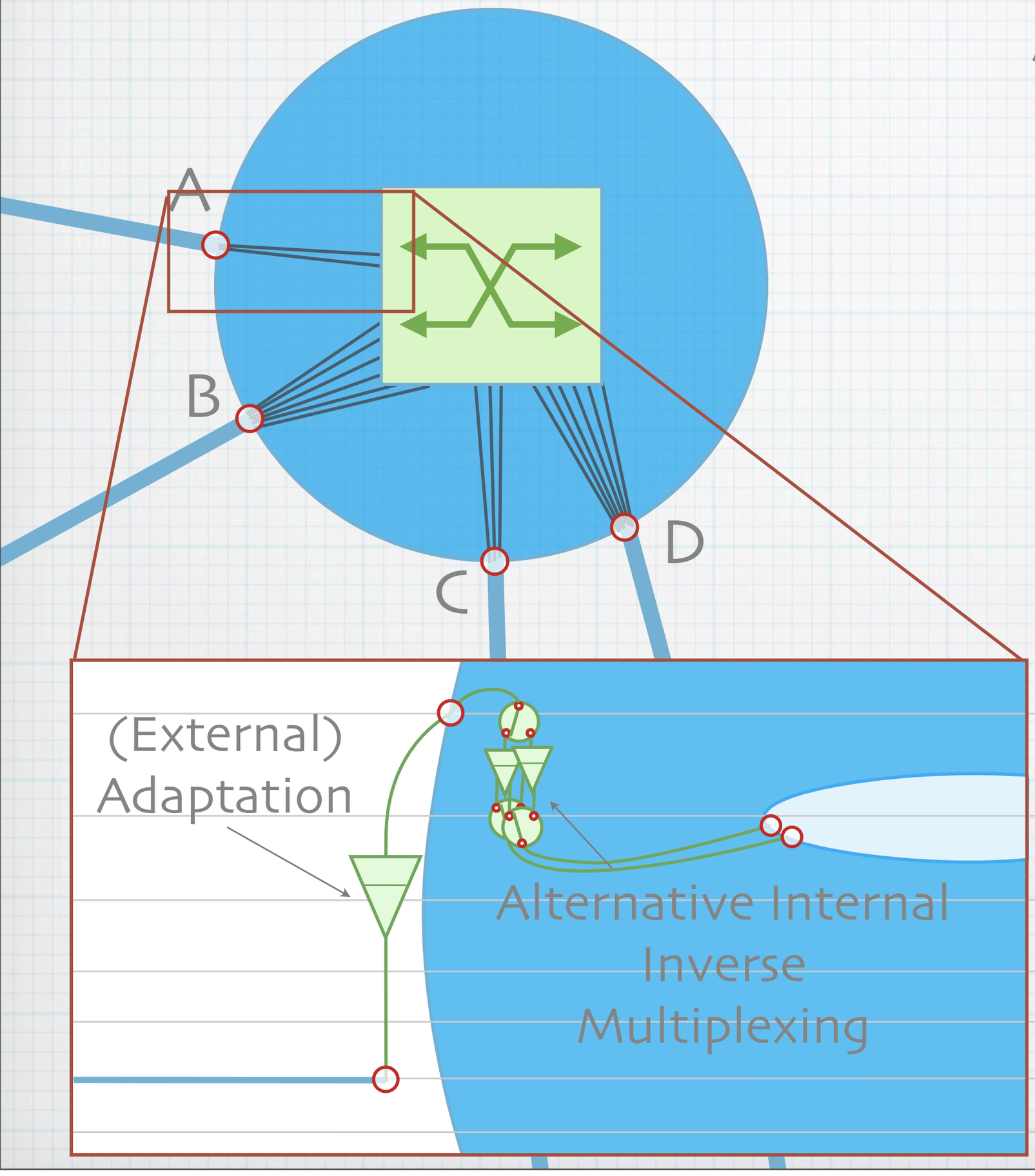
Lambda layer

DWDM layer

UTP layer

fiber layer

Examples of mapping interface → functional elements (connection points and adaptation functions)



Interfaces:

A. Ethernet interface (over UTP) Embeds in either STS-3c-7v or STS-24c

B. OC-192 interface

C. OC-48 interface (over fiber)

D. OC-192 interface (over DWDM at 1552.52nm over fiber)

Ethernet layer

STS layer

Optical Carrier layer

Lambda layer

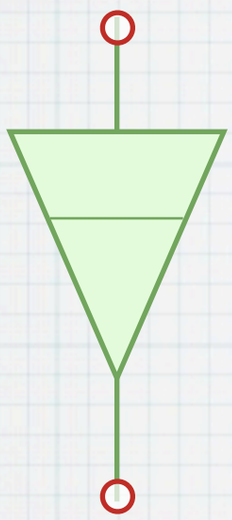
DWDM layer

UTP layer

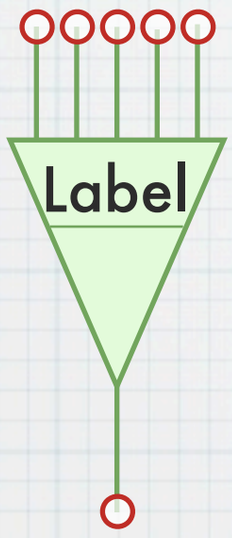
fiber layer

donderdag 28 mei 2009

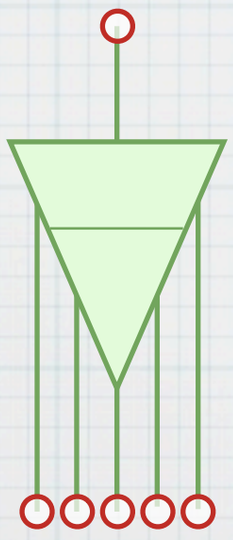
Examples of mapping interface → functional elements (connection points and adaptation functions)



Adaptation



Multiplexing Adaptation



Inverse Multiplexing Adaptation

Adaptation

rdf:ID → URI

clientLayer → Layer

serverLayer → Layer

labels → LabelSet

allowed/available labels

clientLayerCount → integer

>1 for multiplexing.

serverLayerCount → integer

1 by default.

>1 for inverse multiplexing.

clientCapacity → float

provided max. capacity in Bytes/s to the client layer.

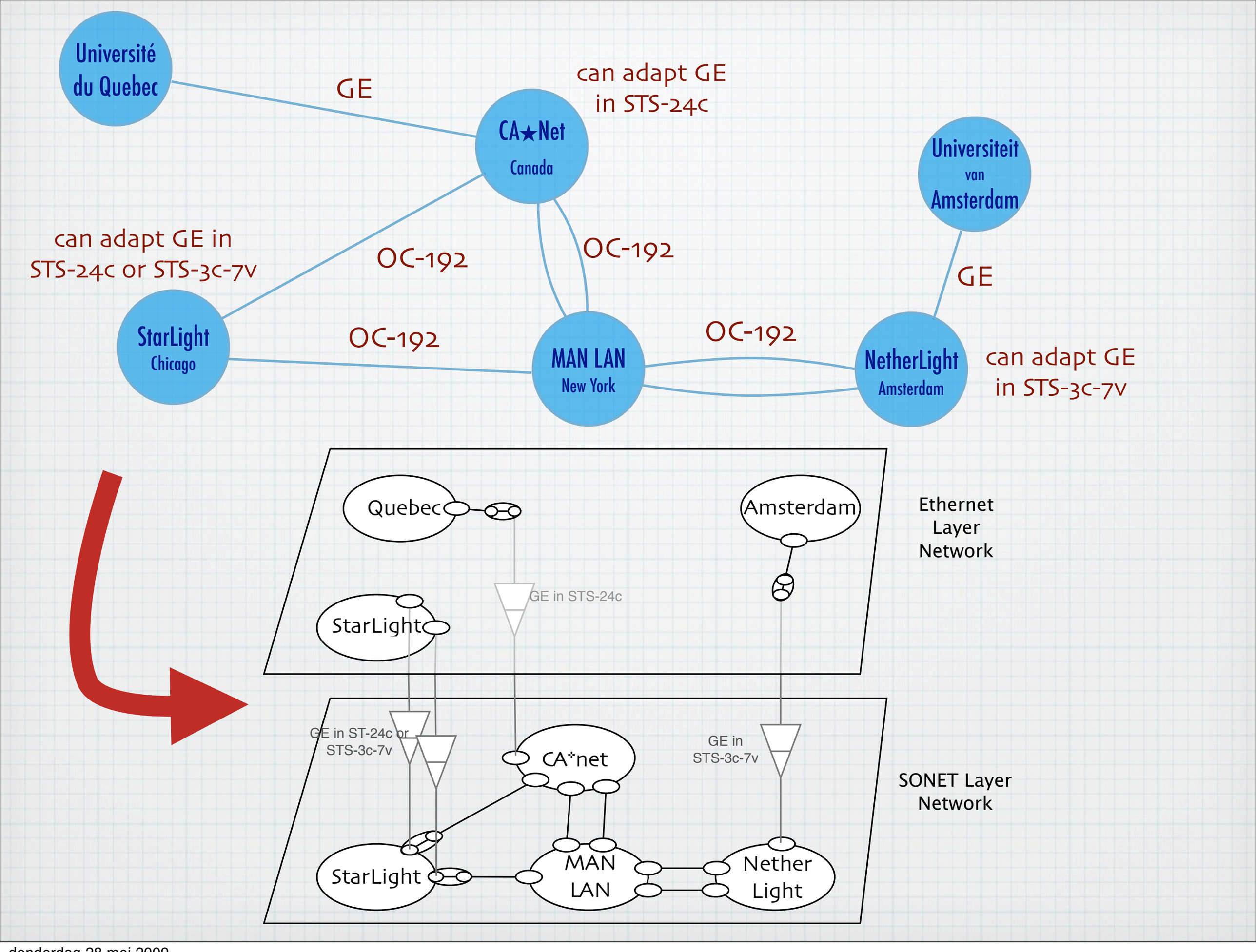
serverCapacity → float

required min. capacity in Bytes/s per channel from the client layer.

donderdag 28 mei 2009

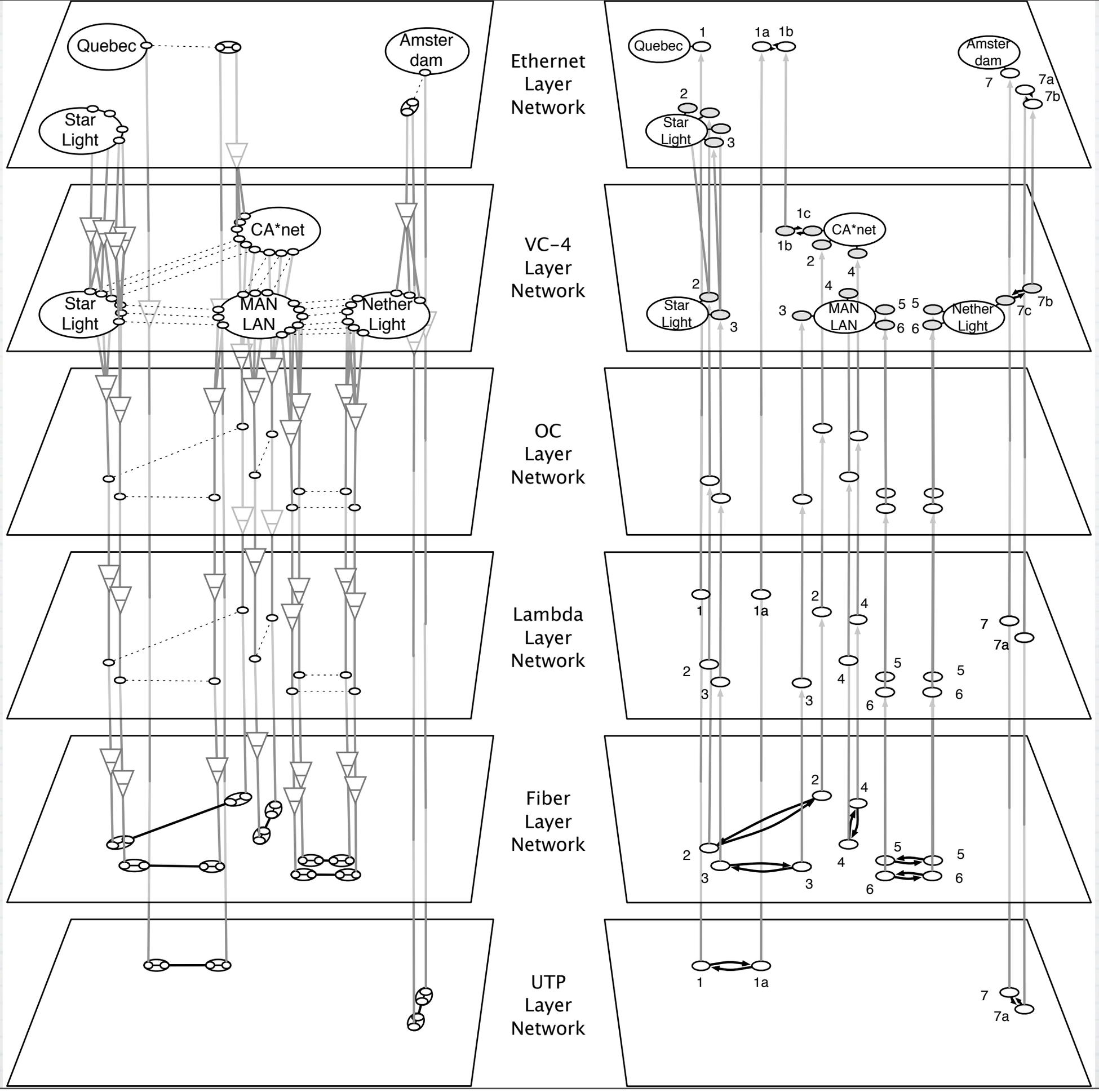
Example of adaptation: Ethernet over UTP, or Ethernet over Fiber.
Example of multiplexing: different data streams, each in a separate wavelength

Example Model



donderdag 28 mei 2009

mapping of the network to function elements. (domains & devices → subnetwork; links → link connections; adaptations; logical interface → connection points)



donderdag 28 mei 2009

Verbose model on the left; compact syntax on the right.

NDL
RDF Syntax

Different Subtopics

- Layer specification
  Definition of different Layers: Layer, Label, Adaptation, etc.
- Topology
  First NDL schema. Recent addition: Path description
- Device capabilities
  Configurable Interfaces, switching & swapping capability.
- Device configuration
  Internal connections, available labels (e.g. free VC-4 channels)
- Domain aggregation
  Functional (network domain) and organizational (admin domain)
- Physical properties
  Location, inventory management (later is based on CIM).

donderdag 28 mei 2009

Each subtopic got it's own schema. We have 4 basic schemas (not mentioned: physical properties, re-use CIM).
In addition, we have 6 layer-specific schema.
Capability: needed for path finding; Configuration: needed for fault isolation.

Technology Properties

- Layer = Specific Encoding
  (1 technology: >1 layers)
- Adaptations
  (from ITU-T G.805)
- Label = Channel Identifier and Switching Capability Identifier
  (from GMPLS)
- Other layer-specific properties
  e.g. MTU size, power level

- IP
- Ethernet
- ATM
- SONET/SDH
- WDM
- Physical layer
- Fiber bundle
- Wireless

donderdag 28 mei 2009

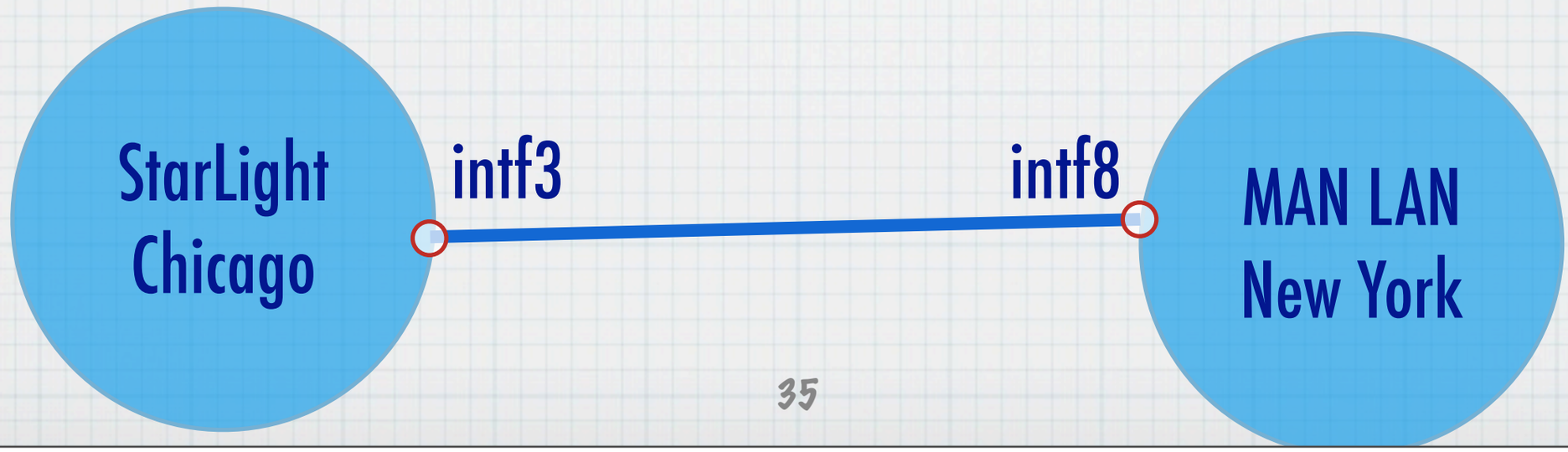
Each subtopic got it's own schema. We have 4 basic schemas (not mentioned: physical properties, re-use CIM).
In addition, we have 6 layer-specific schema.
Capability: needed for path finding; Configuration: needed for fault isolation.

http://startap.net/starlight.rdf

```
<Domain "#StarLight">
  <hasInterface>
    <Interface "#intf3">
      <connectedTo "http://internet2.edu/
          manlan.rdf#intf8"/>
      <seeAlso " http://internet2.edu/
          manlan.rdf"/>
    </Interface>
  </hasInterface>
</Domain>
```

http://internet2.edu/manlan.rdt

```
<Domain "#MANLAN">
  <hasInterface>
    <Interface "#intf8">
      <connectedTo "http://startap.net/
          starlight.rdf#intf3"/>
      <seeAlso " http://startap.net/
          starlight.rdf"/>
    </Interface>
  </hasInterface>
</Domain>
```



Every domain publishes its own data. The seeAlso connect the different data publications together. This is a distributed topology description.

NDL topology

NDL layer

Technology Description

Network Description

topology.rdf

```
57 <rdfs:Class rdf:about="http://www.science.uva.nl/research/sne/ndl#Interface">
58     <rdfs:isDefinedBy rdf:resource="http://www.science.uva.nl/research/sne/schema/topology.rdf"/>
59     <rdfs:label xml:lang="en">Interface</rdfs:label>
60     <rdfs:comment xml:lang="en">A network element(s) that can be represented as a connection po
61     <rdfs:subClassOf rdf:resource="http://www.science.uva.nl/research/sne/ndl#NetworkTransportE
62     <rdfs:subClassOf rdf:resource="http://www.science.uva.nl/research/sne/ndl#ConnectionPoint"/
63 </rdfs:Class>
```

Line: 57 Column: 78 XML Soft Tabs: 4

layer.rdf

```
34 <rdfs:Class rdf:about="http://www.science.uva.nl/research/sne/ndl/layer#AdaptationProperty">
35     <rdfs:isDefinedBy rdf:resource="http://www.science.uva.nl/research/sne/schema/layer.rdf"/>
36     <rdfs:label xml:lang="en">Adaptation Property</rdfs:label>
37     <rdfs:comment xml:lang="en">Adaptation Property are a special kind of rdf:Property. An adaptation de
38     <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
39 </rdfs:Class>
40
```

Line: 34 Column: 93 XML Soft Tabs: 4

wdm.rdf

```
517 <layer:AdaptationProperty rdf:about="http://www.science.uva.nl/research/sne/ndl/wdm#WDM">
518     <rdfs:isDefinedBy rdf:resource="http://www.science.uva.nl/research/sne/schema/wdm.rdf"/>
519     <rdfs:label xml:lang="en">WDM</rdfs:label>
520     <rdfs:comment xml:lang="en">Wavelength Division Multiplexing (WDM): embedding of one or more wavele
521     <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
```

Line: 517 Column: 1 XML Soft Tabs: 4

force10.rdf

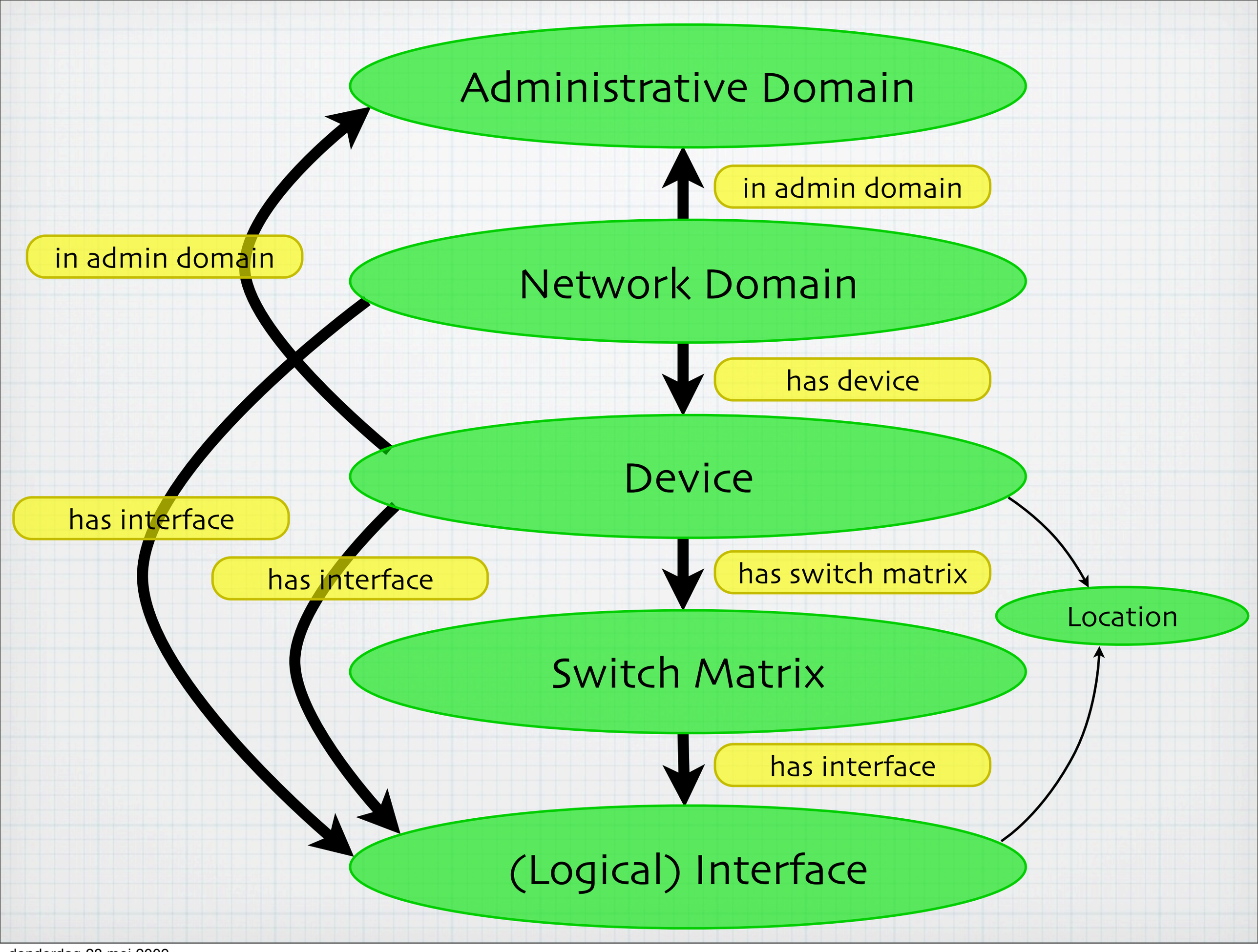
```
1169 <ndl:Interface rdf:about="http://force10.uva.netherlight.nl#gi5/1-fiber">
1170     <rdf:type rdf:resource="http://www.science.uva.nl/research/sne/ndl/wd
1171     <!-- static FiberInterface -->
1172     <rdfs:label>gi5/1 fiber</rdfs:label>
1173     <wdm:WDM>
1174         <ndl:Interface rdf:about="http://force10.uva.netherlight.nl#gi5/1
1175             <rdf:type rdf:resource="http://www.science.uva.nl/research/sn
1176             <!-- static LambdaInterface -->
1177             <rdfs:label>gi5/1 lambda</rdfs:label>
1178
1179             <wdm:eth1000base-X>
1180                 <ndl:Interface rdf:about="http://force10.uva.netherlight.
```

Line: 1173 Column: 1 XML Soft Tabs: 4

donderdag 28 mei 2009

NDL defines a.o. a topology and layer schema. Technologies are specified using the layer schema. Networks are defined using the topology schema, and specific technology descriptions.

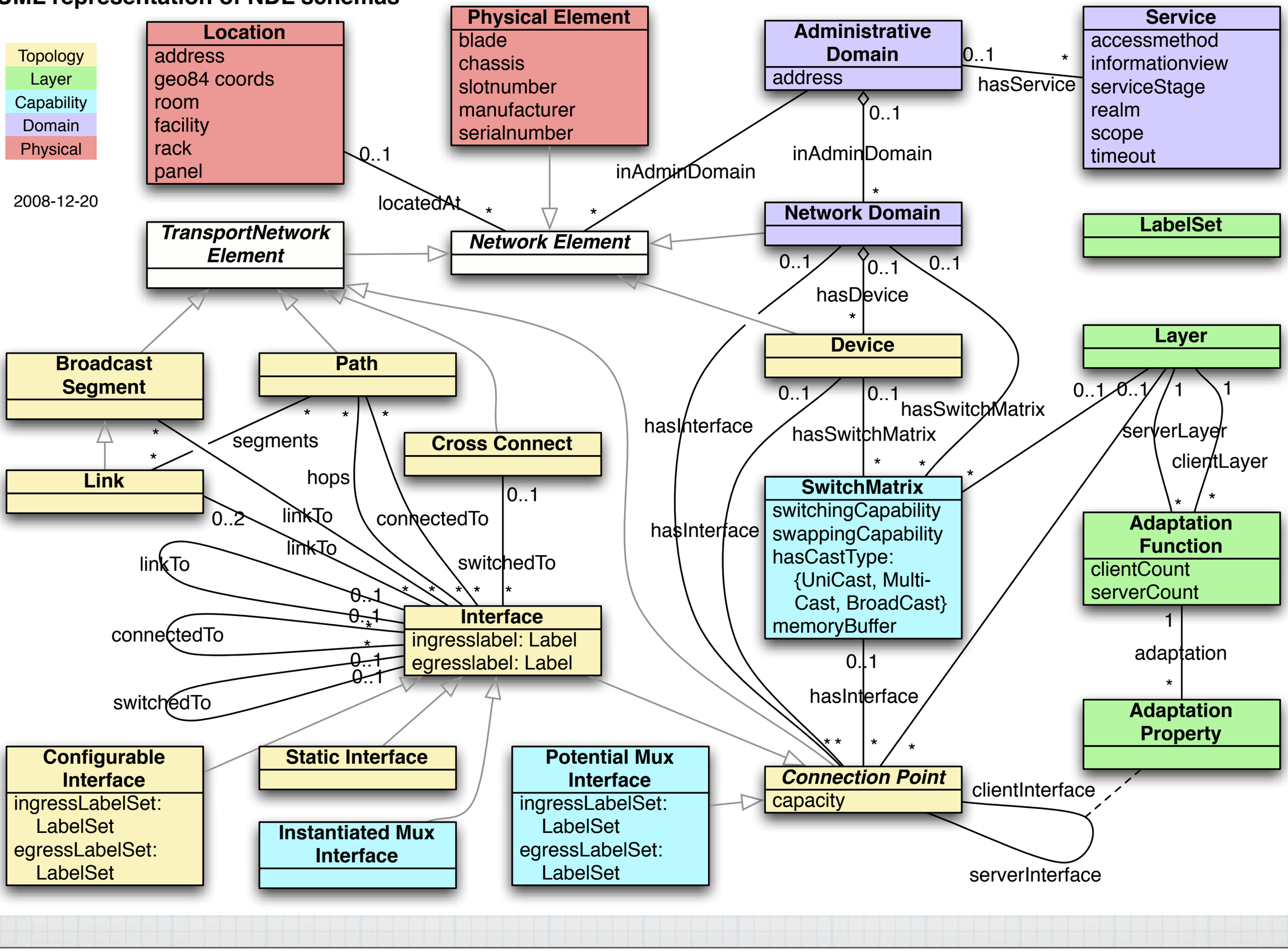
Open Issues



donderdag 28 mei 2009

5 main classes (the class hierarchy)

UML representation of NDL schemas



donderdag 28 mei 2009

All NDL classes, with some properties. Coloured by the schema.

Basics Concepts

- Adaptation stacks
- Switch matrix
- Switching and swapping
- Labels
- Multiplexing (potential interfaces)

The Schema

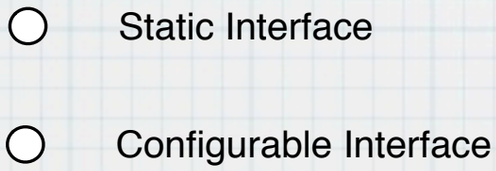
Advanced Concepts

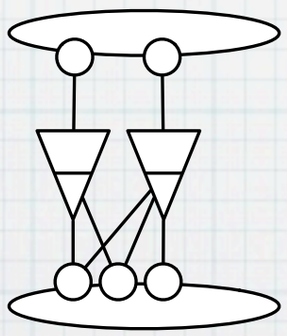
- Optional vs. compulsory labels
- Ingress/egress label (packet switching)
- Internal labels (Untagged Ethernet)
- Internal adaptation stack
- Inverse multiplexing (> 1 server layer)
- Multicast switching
- Broadcast switching

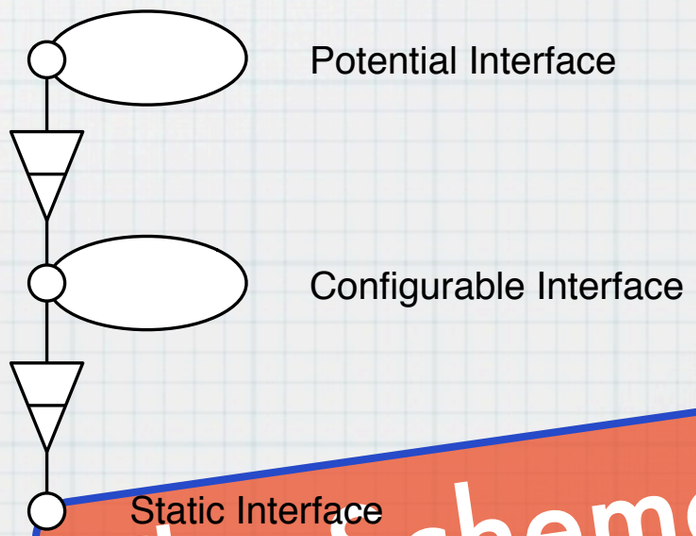
The Schema

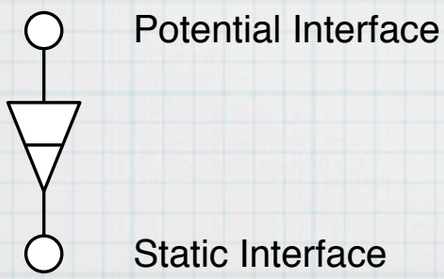
Interfaces

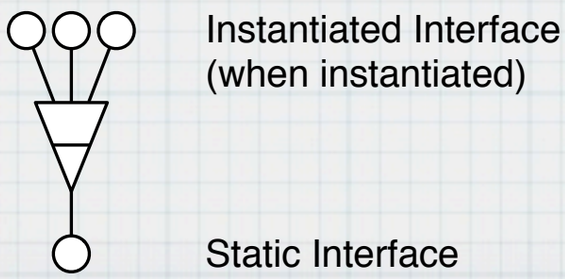
- **Static Interface**
  Fixed interface. Can not be changed in any way.
  laser at 1310 nm
- **Configurable Interface**
  Interface always exists, but can still be configured.
  tunable laser
- **Potential Interface**
  Abstract interface. 0, 1 or many of these interface can be configured.
  "It is possible to create Tagged Ethernet channels"
- **Instantiated Interface**
  Instantiation of a Potential Interface.
  Configured timeslot on VC-4 layer.

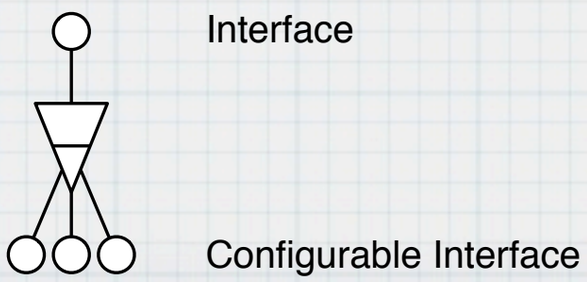












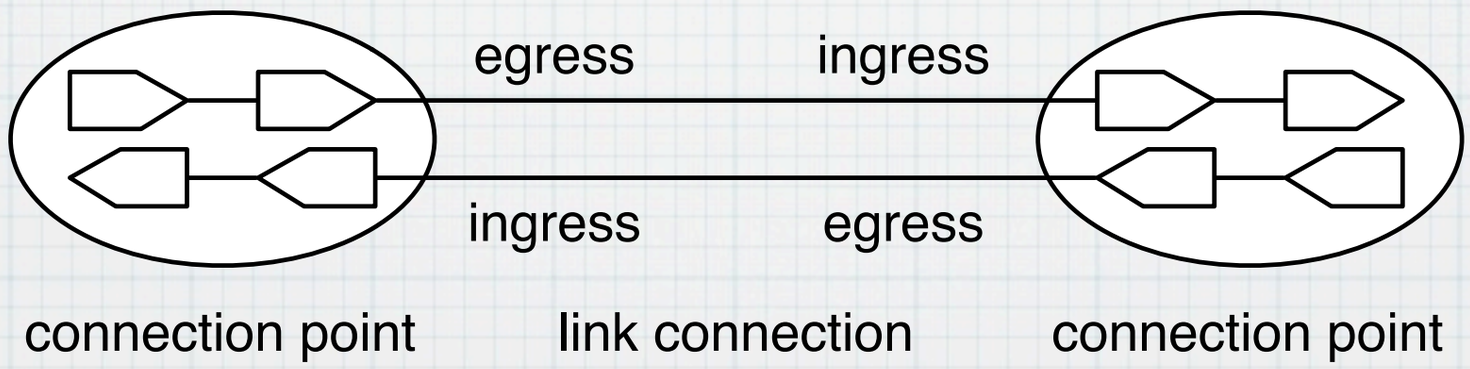
The Schema

Semantic Challenges

X linkTo Y, but not **Y linkTo X** means:

- a unidirectional link
- only X is configured, Y is not (but X would accept data from Y).

Which of the connection points below is configured (admin up/link up), and is there a fiber?



Challenges

Semantic Challenges

If a layer has a label, does it have to exists for an actual Interface?

- The Ethernet label is the VLAN (IEE 802.1Q) label.
- It is only embedded in the data itself for Ethernet over Ethernet (Tagged Ethernet).
- For untagged Ethernet, it is used for switching within a switch matrix
- An untagged channel can have different "label" at each end.

Our solution: we use the "empty label" as concept, but still sometimes it MUST be empty, sometimes it MUST NOT be empty

We only use the IEEE 802.1Q label as the actual label (in the GMPLS sense), and the VLAN tag as an "internal label", for switching only.

Challenges

Semantic Challenges

What does a **Potential** or **Available** configuration mean:

- Is it technically possible? Possible without breaking other connections ?
  If so, what does "breaking" mean? What if I reconfigure the other switch connection? Is that broken?
- Is it administratively possible?

We distinguish between actual (is configured/static), potential and available

Logical Challenges

Give me all "switchTo" means:

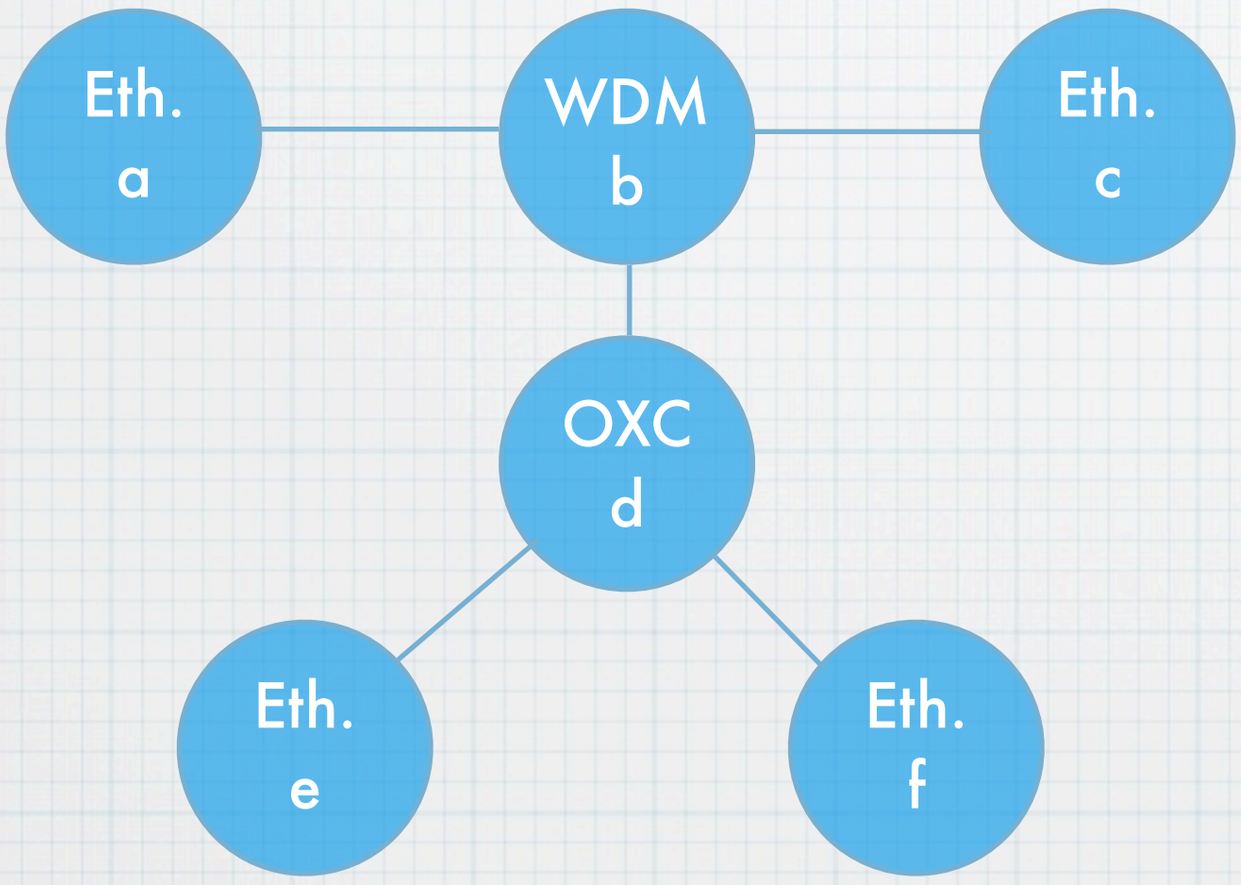
Depends on:

- Question: do you want Actual/Potential/Available switchTo?
- What kind of interfaces are we talking about: Static/Configurable/Potential/Instantiated
  Do we return one or two switchTo for a Potential and Instantiated interface?
- What type of switch matrix, if any: None (patch panel)/Unicast/Multicast/Broadcast
- Can the switch matrix convert between labels (switching & swapping)

Logical Challenges

When is a switchTo (subnetwork connection) in use?

- We can re-use a connection at a lower layer, as long as the labels are different on higher layers (different channels).



Example:

Ethernet A -> C need to go to E and F:

- A -> E: 10 Gb/s LAN PHY
- E -> F: 10 Gb/s WAN PHY
- F -> C: 1 Gb/s

How/when to detect that this is not possible due to a conflict at OXC d?

Challenges

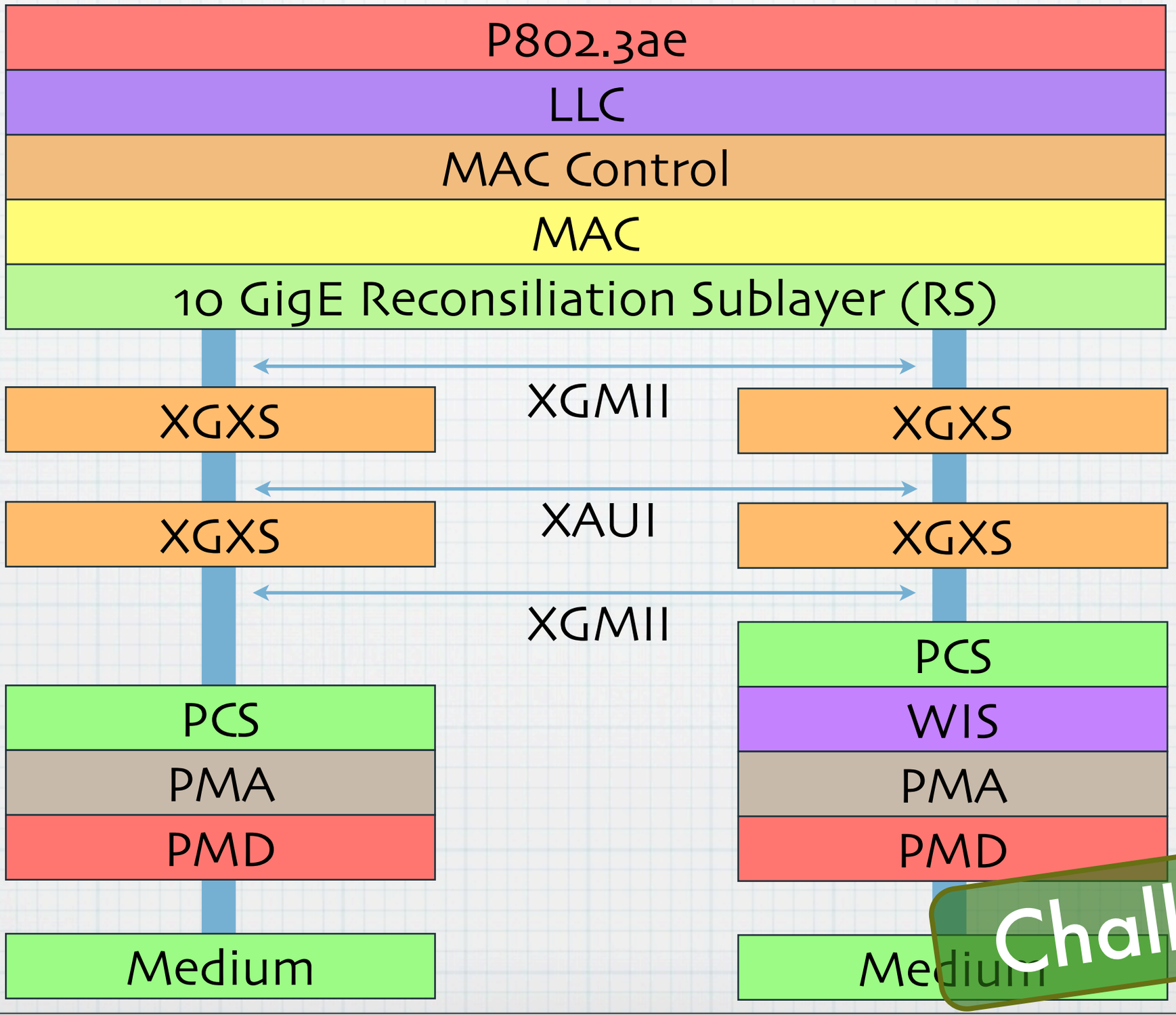
Logical Challenges

For a path, 4 channels over the same client layer are required:

- A. Must have label in set {3,4}
- B. Must have label in set {3,4}
- C. Must have label in set {3}
- D. Must have label in set {4-11}

How to detect this is not possible? If we sequentially pick a label for each channel, we may get a false negative.

Practical Challenges



Which layers should be specify in practice?

Questions