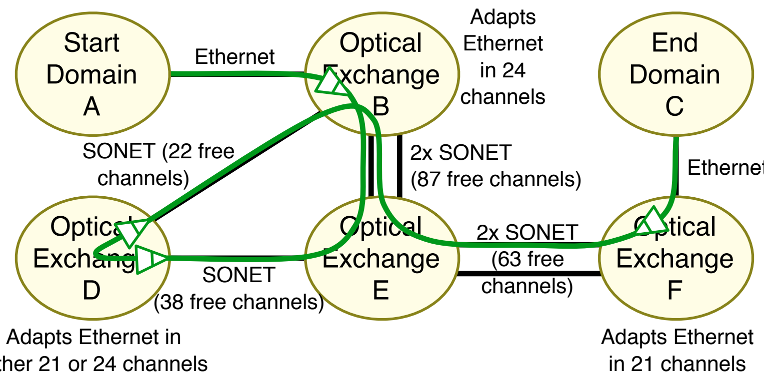
Going in Loops to Reach Your Goal

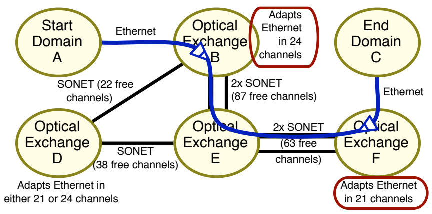
Multi-Layer Path Finding with NDL

Walking the Same Path Twice

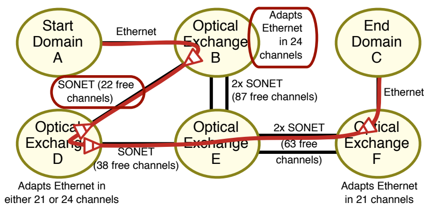
Multi-layer networks span different technologies. An example is a network with both Ethernet and SONET devices, where Ethernet packets are encapsulated (*adapted*) in SONET timeslots. Path finding in multi-layer networks is complex. In some cases, the shortest network connection must use the same link twice, as shown in this example network:



Correct: Shortest path between A and C. The link B – E is used twice.



Wrong: Adaptation of Ethernet in 21 channels is incompatible with Ethernet in 24 channels.



Wrong: There are only 22 free channels between B and D, while 24 channels are required.

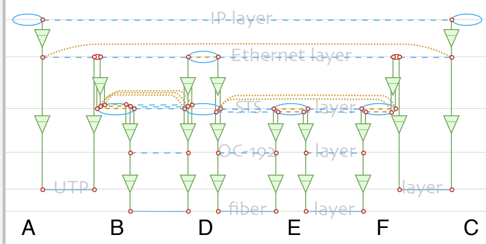
Multi-Layer Network Model

Given the complexity of path finding in multi-layer networks, we set the **goal**:

Create a **computer-readable network description**, that provides enough **information** for automated **path finding** in **multi-layer networks**.

Prior to defining a syntax, we first created an abstract **information model**. We based our model on **ITU-T G.805** functional elements and the label concept in **GMPLS**.

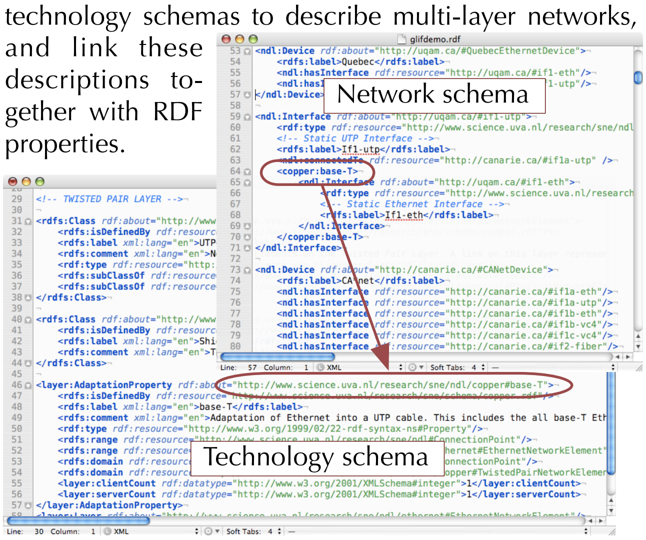
For example, the red path in the example on the left is modelled as:



While our model is clearly **layer aware**, it is still **technology agnostic**: All technologies are represented as an abstract set of layers with adaptations, labels, and layer properties. There is no need to change the model as new technologies emerge.

Network Description Language (NDL)

Multi-Layer NDL is an implementation of our network model based on RDF. It allows the creation of technology schemas. Network engineers can use these technology schemas to describe multi-layer networks, and link these descriptions together with RDF properties.



Path Finding Demonstration

We implemented a parser for the Network Description Language. A simple breadth first search algorithm is able to find shortest paths in multi-layer networks. The output below returns the green path in the example network:

```
           Starting point --> Domain A    if1   Ethernet
                  Link to --> Exchange B  if1   Ethernet
     Adaptation GE in 24 STS --> Exchange B  if1   Ethernet over 24 STS
   Through CAnetSwitchMatrix --> Exchange B  if4   Ethernet over 24 STS
    Adaptation STS in OC-192 --> Exchange B  if4   Ethernet over 24 STS over OC-192
                  Link to --> Exchange E  if4   Ethernet over 24 STS over OC-192
 De-adaptation STS in OC-192 --> Exchange E  if4   Ethernet over 24 STS
        Through SONET switch --> Exchange E  if3   Ethernet over 24 STS
    Adaptation STS in OC-192 --> Exchange E  if3   Ethernet over 24 STS over OC-192
                  Link to --> Exchange D  if3   Ethernet over 24 STS over OC-192
 De-adaptation STS in OC-192 --> Exchange D  if3   Ethernet over 24 STS
  De-adaptation GE in 24 STS --> Exchange D  if3   Ethernet
     Through Ethernet switch --> Exchange D  if2   Ethernet
     Adaptation GE in 21 STS --> Exchange D  if2   Ethernet over 21 STS
    Adaptation STS in OC-192 --> Exchange D  if2   Ethernet over 21 STS over OC-192
                  Link to --> Exchange B  if2   Ethernet over 21 STS over OC-192
 De-adaptation STS in OC-192 --> Exchange B  if2   Ethernet over 21 STS
        Through SONET switch --> Exchange B  if5   Ethernet over 21 STS
    Adaptation STS in OC-192 --> Exchange B  if5   Ethernet over 21 STS over OC-192
                  Link to --> Exchange E  if5   Ethernet over 21 STS over OC-192
 De-adaptation STS in OC-192 --> Exchange E  if5   Ethernet over 21 STS
        Through SONET switch --> Exchange E  if6   Ethernet over 21 STS
    Adaptation STS in OC-192 --> Exchange E  if6   Ethernet over 21 STS over OC-192
                  Link to --> Exchange F  if6   Ethernet over 21 STS over OC-192
 De-adaptation STS in OC-192 --> Exchange F  if6   Ethernet over 21 STS
        Through SONET switch --> Exchange F  if8   Ethernet over 21
  De-adaptation GE in 21 STS --> Exchange F  if8   Etherne
                  Link to --> Domain C    if8
```

Output of the path finding algorithm

Freek Dijkstra, Bert Andree, Paola Grosso, Jeroen van der Ham, Karst Koymans, Cees de Laat

http://www.science.uva.nl/research/sne/ndl/

GigaPort UvA Universiteit van Amsterdam TNO NDL