Framework for path finding in multi-layer transport networks

Dijkstra, F.

Publication date
2009

Document Version
Final published version

Chapter 9

Discussion and Conclusion

9.1 Context and Goals

A number of e-Science applications need deterministic point-to-point connections with very high bandwidth and predictable behaviour.

The need for dedicated network connections is in practice fulfilled by interconnecting multi-layer networks where the dedicated resources are allocated for each network connection.

In this field of multi-layer multi domain transport networks there is (until now) neither an agreed upon network model, nor a suitable end-to-end path provisioning algorithm.

The aim of this work is to make a system level analysis of multi-domain, multi-layer transport networks. In doing so, it contributes to the design of a comprehensive control plane for these transport networks.

Actual contributions include the proof that graphs that simply represent devices or domains as vertices and links as edges are not sufficient to describe multi-layer networks, because that ignore the adaptation functions. We have shown that that link-restriction algorithms are not sufficient for path finding algorithms but path-restriction algorithms are sufficient.

Logical reasoning and the implementation and demonstration of path-restriction algorithm is evidence of the correctness of our claims.

9.2 Contributions to the Field

This thesis contributes to the new field of hybrid networking, including the modelling of optical exchanges, ontology for network descriptions with separation of topology and technology, and proof that link-constrained path finding algorithms do not work in multi-layer networks. The work is validated by implementations and proof of concepts.

Path finding through multi-layer networks It was shown that link-constrained algorithms in use for single layer networks such as the Internet and the telephony network are not applicable for multi-layer networks. A path-constrained algorithm was introduced that is suitable for path finding in multi-layer networks. A shortest path in a multi-layer network can contain a loop.

Modelling of optical exchanges Both practical models and a concise terminology for exchanges was created. While the data plane for Internet exchanges and optical exchanges are similar, there is a clear distinction on the control plane. Optical exchanges can not be transparent to a path finding if they offer services such as data conversion.

Ontology for network descriptions In order to describe networks and network technologies, multiple ontologies were created. The ontologies are built upon the resource description framework (RDF), and allow integration with other ontologies. The network description ontology is used in network tools for visualisation, path finding and fault isolation, and is used in production at the SURFnet6 NOC.

Separation of topology and technology information Separate ontologies were created to describe networks topologies and network technologies. It is possible to reason about topologies independent from the technologies. The combination of the two allows network engineers to describe compatibilities and incompatibilities in their network.

9.3 Strengths and Weaknesses

9.3.1 Architecture

When we started this work, the concept of hybrid networks was just established [p23], but there was no concise model or even description of optical exchanges or hybrid networks. In two peer-reviewed articles, we described the architecture of optical exchanges, both at the data plane as well as the control plane. This work was essential to establish a more formal model, as we later did.

We asked ourselves two questions, a generic question and a specific question. The generic question asks if there is a fundamental difference between hybrid networks and existing networks such as the Internet or the telephony network? The specific question narrows this to: Can optical exchanges, just like Internet exchanges, be completely transparent to a path finding algorithm in circuit switched networks?

We could answer our specific question with a reasoning about the terminology for transparency. We conclude that exchanges that offer adaptation or interworking services can not be transparent to a path finding algorithm.

This conclusion is relevant, as we also reason that most optical exchanges will have to offer adaptation or interworking services. Technologies change over time, which implies that incompatibilities between interfaces, adaptations and layers will continue to occur. Exchanges must take potential incompatibilities into account to avoid non-working network connections. Services like adaptation and interworking must be offered by exchanges or elsewhere in the network to solve incompatibilities.

Our work did not stop by answering this question alone. We found that the distinction between Internet exchanges and optical exchanges lays at the control plane, rather than at the data plane. While Internet exchanges and optical exchanges may use the same technologies, they are different at the control plane: optical exchanges change state with each connection request, while the state of Internet exchanges only change when peering relations change.

9.3.2 Modelling

A large part of the work in this thesis is modelling of multi-domain and multi-layer networks.

This work was initiated after we found that, to our surprise, there was no comprehensive multi-layer network model available yet. Our work made three distinct choices:

- The clear distinction between a network, a functional representation of the network, and a syntax describing the functional description.
- The use semantic web technologies such as RDF [p41, a3].
- The use of G.805 functional elements and GMPLS labels to model multi-layer networks.

Using the resource description framework (RDF) immediately solved two engineering questions: naming of interfaces, domains and hosts, and the multi-domain problem using a distributed knowledge base. In addition, it allowed easy integration of the terminology we developed earlier into a formal ontology.

The very strict distinction between the physical network and the functional representation of that network (the model) allowed us to see that graphs are not sufficient to describe multi-layer networks, despite their use to describe single-layer networks. As far as we are aware, we are the first to recognise this.

Not only did we recognise that graphs are not sufficient to describe multi-layer networks, we also provided an alternative, using G.805 functional elements in combination with the label concept of GMPLS.

The distinction between model (using functional elements) and syntax was also beneficial. While this distinction may seem trivial, it is extremely important when it comes to creating control plane software for those multi-domain networks. As we write, each domain deploys their own control plane software (for example, UCLP, DRAC, G-Lambda, AutoBAHN, DRAGON, etc.). If each software uses a different syntax, but the same model, it is easy to translate between the different network topology descriptions. However, if the model is different, this translation is very hard, and sometimes even impossible. In reality, it is often ever harder, since most software packages do not define a formal network model in the first place. For example, GMPLS did not define a formal model, but only a syntax to describe networks. Our work makes software developers aware of this problem, and offers a solution by providing a syntax-independent model.

The strict of modelling not only helps software designers, but also protocol designers. Indeed, proof that our model is considered useful is the interest for our model by members of the network markup language (NML) working group in the Open Grid Forum (OGF), whose goal it is to come up with a standard for network descriptions. In addition, the interest of external parties for the work of the NML working group also shows the importance of such a model [u2]. More and more, the value of a solid network model is recognised.

The network markup language (NML) working group has another benefit: it allows a comparison between our model and the efforts of others. While our model still lacks practical extensions such as change information (our model assumes a static network, while in reality networks change over time), it appears to have a few unique properties that make it stand out in comparison. First is the technology independence: our model is the only that allows a path finding algorithm that does not need specific knowledge about each technology. The clear advantage is that our model is *forward compatible*: it is not only compatible with existing technologies, but also with future technologies, provided that they can fit in our model. Secondly is our use of the label concept, we took from GMPLS. This technology independent feature is considered a asset of our model by others. Third, and finally, our model is one of the few that distinguishes in terminology between the network at that data plane and the domain at the control plane.

We have tested the technology independence of layers, adaptation and labels for all network technologies we are familiar with (IP, Ethernet, PPP, ATM, SONET/SDH, VPN tunnel, Wireless, DWDM and CDWM, OXC, Fibre bundles). We concluded that the model works correct for circuit-switched technologies, including all technologies that are used at existing optical exchanges (Ethernet VLANs, SONET/SDH, DWDM and fibre layers), while extensions would be required for physical layers and packet switch technologies that use a routing table. We believe this to be very strong result.

9.3.3 Path finding

By using logical reasoning, we have shown that link-constrained path finding algorithms are not sufficient for path finding in multi-layer networks, if links are 1:1 mapped to edges. We defined a shortest network connection with a loop is defined as a shortest network connection that uses the same physical link twice. We assert that link-constrained path finding algorithms never find a solution with the same edge used twice. Therefore, our claim is valid.

Routing protocols such as BGP and SS7 use a path-vector algorithm and the routing protocol in OSPF-TE/PCE (as used in GMPLS) uses a link-state algorithm for path finding. Both are link-constrained algorithms, and can not be used as-is in multi-layer networks.

We have given two path finding algorithms, both path-constrained, that are designed to find paths in multi-layer networks. One of these is a broadcast-like algorithm, which has the advantage that it uses a graph that is very similar to the actual network topology. It does not a-priori need information about other domains until it has to check for paths through those domains.

This variant is currently the only variant that is capable of dealing with loops, as well as the assertion that a segment of a shortest path does not need to be a shortest path in itself.

Our strength is that we not only defined a model and algorithm, but also made an implementation of our algorithm. By applying this algorithm to a few complex example networks that contain loops, we have shown its applicability. We conclude that path-constrained algorithms are sufficient for path finding in multi-layer networks.

This last proof is not a formal mathematical proof, but merely a proof of concept. By strictly retaining the technology independence of the algorithm, we have shown that the path-constrained path finding algorithm works for all networks and technologies that can be described by our model. That includes all relevant technologies, since those can all be described with our model.

However, we did not formally prove that there is no future technology can be invented that can not be applied to our model, and which would require an even more complex algorithm.

We have seen examples of incompatibilities that are not described in our model. Incompatible packet sizes in Ethernet, and incorporation of scheduling and policy constraints have been explicitly mentioned. Our path-constrained path finding algorithm can still handle those scenarios, by simply adding more constraints to the “feasible path” constraint code. This is a trade-off between the number of constraints that can be handled and the technology independence of the model and algorithm.

Our algorithm is basically a breadth first search algorithm. We have shown a few optimisations that reduce the flooding mechanisms by dropping paths that will not result in the shortest path for whatever reasons (e.g. we drop paths with unnecessary loops). Further research may reveal radical new algorithms that are even more efficient than these optimisations. Despite our collaboration with renowned algorithmic experts from Delft University, such radical new approaches have not surfaced yet.

9.4 Claims and Statements

We focused on one specific question in this thesis.

- Is it possible to use the same path finding algorithms in multi-layer transport networks as those in use for the Internet and telephony networks?

The research we carried out in order to answer this and subsequent questions have accumulated in miscellaneous statements as we put forward in this thesis.

- The use of multiple technologies causes incompatibilities (Section 2.3.1, repeated in Chapter 3).
- Technologies evolve over time. Thus, incompatibilities will continue to exist (Section 2.3.1).
- Exchanges must take potential incompatibilities into account to avoid non-working network connections. These incompatibilities must be described to deal with them (Section 2.3.2).
- Exchanges can only be ignored during path finding if the connections through an exchange are modelled as direct connections, and the exchange does not define a usage policy on its own (Section 2.5).

- Path finding in a single layer network belong to a different complexity class (P) than path finding in a multi-layer network (NP-hard) (Section 3.3, source: Kuipers [a12]).
- the algorithms used in the Internet and telephony network can not be used for path finding in multi-layer transport networks, if links in the network are 1:1 mapped to edges in the graph (Section 3.3.1).
- Link-constrained algorithms are not sufficient for path finding in multi-layer networks, if links are 1:1 mapped to edges (Section 3.3.2).
- Graphs can not be used for path finding in multi-layer networks, if it is required that (1) the outcome of the path finding algorithm is sufficient to reconstruct the original path in the network; (2) the graph can be created from the actual network in polynomial time; and (3) a link-constrained path finding algorithm is used (Section 3.3.2).
- Each of the conditions in the above claim is essential for the claim (Section 3.3.3 and 3.3.4).
- Multi-layer incompatibilities can not be resolved locally, but need to be distributed across domains (Section 3.3.2).
- Path-constrained algorithms are sufficient for path finding in multi-layer networks (Postulated in section 3.3.2, proven in section 8.3.6).
- Multi-layer networks can only be mapped to a graph if devices are mapped to multiple vertices, or if information about the adaptation is lost (Section 3.3.4).
- G.805 and G.800 allow descriptions of the state of a network. No model exists to describe how to change that state, and who may do so (Section 4.2).
- It is easy to translate between two different syntaxes with the same model. It is hard to convert between two models (Section 4.3.1).
- A model can be verbose, while the syntax is compact (Section 4.4.3).
- It is possible to create a distributed network description, without a central repository (Section 5.3.3, source: Van der Ham [a3]).
- The use of semantic web provides the URI as a solution for globally unique addressing of network resources (Section 5.3.4, source: Van der Ham [a3]).
- Incompatibilities change over time, and what needs to be described changes over time (Section 6.1.2).

- A multi-layer path finding algorithms should be layer independent (Section 6.1.2).
- A segment of a shortest path in a multi-layer network does not have to be a shortest path in itself (Section 7.3.1).

9.5 Conclusion

In chapter 1, we put forward a general research question of the research in this thesis is only a small part of, *Is there a fundamental difference between hybrid networks and existing networks such as the Internet or the telephony network? Which of the existing models and approaches can be re-used and which can not?*

Given the statements we made in this thesis and repeated in the previous section, we can only conclude that indeed there are fundamental differences between hybrid networks and existing networks such as the Internet or the telephony network. The concept of networks, and operational domains remains the same. Hybrid networks encompass both circuit switched as well as packet switched technologies. The largest difference between hybrid on one hand, and the Internet or telephony network on the other hand is not packet versus circuit switching, but the fact that the circuit switched technology of hybrid networks is a **multi-layer network** service. This multi-layer nature gives rise to interworking services to stitch technological incompatibilities together. This causes that optical exchanges that offer these interworking services are visible to a path finding algorithm, and that link-constrained path finding algorithms as used for single layer networks can no longer be used. Instead, we have shown that path-constrained algorithms are required, and that simple graph as common to describe single layer networks can not simply describe the multi-layer networks.

All these differences mean that if no single specific technology for hybrid networks is chosen, new multi-layer models and path finding algorithms needs to be developed.

Given that technologies evolve, and thus incompatibilities continue to exist, no single layer or single technology is standardised, and multi-layer network models and path finding algorithms continue to be needed for hybrid networks.