Cross-layer Visibility as a Service

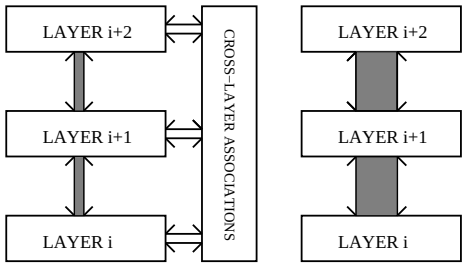
Ramana Rao Kompella[*], *Albert Greenberg*[†], *Jennifer Rexford*[‡], *Alex C. Snoeren*[*], *Jennifer Yates*[†]
[*]*UC San Diego* [†]*AT&T Labs – Research* [‡]*Princeton University*

Abstract. Accurate cross-layer associations play an essential role in today's network management tasks such as backbone planning, maintenance, and failure diagnosis. Current techniques for manually maintaining these associations are complex, tedious, and error prone. One possible approach is to widen the interfaces between layers to support auto discovery. We argue instead that it is less useful to export additional data between layers than to import information into a separate management plane. The specification of a management interface enables independent evolution of individual layers, side-stepping the challenges inherent in wide layer interfaces. Further, the management plane can leverage network-wide cross-layer visibility to provide enhanced services that depend on physical- or link-layer diversity.

1 Introduction

The Internet still lacks the kind of reliability and robustness we expect from critical infrastructure. Network events, such as equipment failures and planned maintenance, often cause disruptions in service, or even the complete loss of connectivity between end hosts. Diagnosing the root cause of these problems is surprisingly difficult. On the positive side, routing protocols and overlay networks respond automatically to network events (e.g., by computing new paths) and network designs intentionally incorporate redundancy (e.g., ISPs build transport networks with diverse optical facilities, and enterprises often connect to the Internet at multiple locations). However, the redundancy is not always as rich as it seems. Multiple IP links may run through the same optical components, and multiple fibers may share the same risks (e.g., as seen during the Baltimore tunnel fire [1]). In this paper, we argue that *poor visibility into the dependencies between layers is a major impediment to improving the reliability of the Internet*.

In essence, a link at one layer (e.g., IP) consists of a path—a sequence of components—at the next layer (e.g., fibers and optical amplifiers). Greater visibility across layers would significantly improve network planning, risk assessment, fault diagnosis, and network maintenance, as discussed in Section 2. In practice, ISPs address these problems by maintaining complex databases and analyzing large amounts of topology, configuration, and measurement data collected from network elements at each layer, as illustrated in Figure 1(a). This approach is driven primarily by the absence of any immediately viable alternative, since most layers have little or no visibility into the other layers. As a long-term alternative to this seemingly ad hoc approach, we could imagine "fattening" the interface between layers to make network elements more aware of dependencies they inherit from the layers below and impose on the layers above. These fat interfaces would enable network elements to select diverse paths that avoid shared risks and provide greater visibility to troubleshooting tools like traceroute.



(a) Thin interfaces (b) Fat interfaces

Figure 1: Two ways to provide cross-layer visibility

Despite the apparent advantages of wider interfaces, we argue that this is the wrong approach to the problem, *even if we had the luxury of a clean-slate redesign of the interfaces between layers.* First, the simple abstraction of a link plays an important role in containing complexity inside the network; wider interfaces make it harder for each layer to evolve independently. Second, some network elements cannot easily provide information about shared risks to the layers above them for fundamental reasons; for example, a fiber cannot easily notify an IP link about the underground locations it traverses, and whether other fibers lie nearby. Third, having the network elements store historical data and answer queries is challenging, because of the overhead involved and the need for providers to control access to the data for business and security reasons. Finally, and perhaps most importantly, detailed cross-layer visibility is primarily necessary for designing, managing, configuring, and troubleshooting the network, making it appealing to store the information *outside* of the network elements. We discuss these issues in greater detail in Section 2.3, and also present concrete examples that illustrate these points.

Still, today's ad hoc approach of collecting and analyzing data in home-grown databases is not a sufficient solution, either. Instead, we argue that cross-layer visibility should be provided as a *service*, with well-defined interfaces for populating the external databases and querying the information, as discussed in Section 3. Rather than

dictating what the network elements store and export—the approach taken by the Simple Network Management Protocol (SNMP) [2]—we focus on what information is *imported* into the management database. This subtle distinction is extremely important, as it allows many different solutions for providing the information. Although the network elements themselves could generate the data (as in SNMP), the information could also come from separate measurement devices or even human operators. This approach accommodates the inherent diversity across the layers and the natural evolution of techniques for collecting the data. Section 3 presents a possible evolution path for three layers—determining the IP forwarding path, mapping an IP link to optical components, and identifying fibers running through the same geographic location. These examples could easily be extended to include other protocol layers, such as paths through an overlay network or a sequence of tunnels or MPLS label-switched paths.

Greater uniformity in the data representation would make it easier to evolve a network, integrate two networks after an acquisition, and employ third-party network-management tools. More broadly, we argue that the management system should have interfaces for different stake holders—such as network designers, network managers, and customers—to query the data, with explicit policies governing the kinds of information each party can access. For example, a customer could ask if two IP paths (or two access links) are physically diverse but might not be told that the fibers run through the same tunnel. In contrast, a network manager troubleshooting a reachability problem could perform a complete "traceroute" of an IP path across all of the layers. A network designer could conduct a "what-if" analysis of the effects of planned maintenance on the link loads. The system can also keep a log of past queries, to learn more about the cause and impact of failures by analyzing patterns in the queries. Maintaining explicit cross-layer visibility information presents a number of interesting research and operations issues which we discuss in Section 4.

2 The case for cross-layer visibility

Layering in IP networks fundamentally hides complexity of lower (upper) layers (e.g., the underlying optical network topology) and exposes very simple interfaces to upper (lower) layers (e.g., a logical link) thus allowing parallel and independent evolution of these layers while still preserving the interface between them. However, we argue that strict layering results in *poor visibility* across layers affecting certain operational tasks that rely on accurate cross-layer visibility.

2.1 Accurate associations are critical

In today's IP backbone networks, each IP link consists of a connected set of optical components organized in different topologies (e.g., ring, mesh, etc.). A single link consists of many different optical components and many different links can share a particular component, thus creating a many to one, one to many mapping. Cross-layer visibility refers to the associations between higher layer abstractions to lower layers and vice versa. For example, in IP networks it refers to the association between an IP point-to-point link to the set of optical components that comprise the link.

Accurate associations are critical to the functioning of various operational tasks—some of which have been described below.

- *Backbone planning.* Backbone planning involves engineering the network to withstand a wide range of potential failure scenarios including possible attack scenarios, plan traffic growth, and to support additional services and features in the network. An accurate audit of the network that transcends all layers, therefore, is a key ingredient in backbone planning. Often, operators perform a "what-if" analysis before maintenance and other activities; this also requires accurate associations between layers. For example, before shutting down a link between Los Angeles and San Diego, operations analyze if the network has sufficient spare capacity to re-route additional traffic through other paths. IP paths are typically selected to avoid any single points of failures commonly referred to as Shared Risk Link Groups (SRLGs) [3]. Accurate IP-to-optical associations in databases are required to choose physically diverse paths to carry traffic to withstand failures in the lower layers. If IP-to-optical associations in databases are erroneous, it can result in engineering the network to erroneously choose non-diverse paths to carry traffic; a single failure in turn can partition the network.
- *Customer fault-tolerance.* Customers (e.g., e-commerce businesses) are primarily interested in obtaining uninterrupted network connectivity either from one single service provider or through different service providers via multi-homing. One common question they often face is about the level of diversity in their connectivity to the backbone. Even when they connect to different points-of-presence (PoPs) within the same provider, or to two different carriers (e.g., Sprint and AT&T) there could be shared risks lurking (e.g., fibers passing through the same tunnel) that could be of concern to the customer. Whether to disclose such information completely or in part about physical connectivity is often a policy decision; accurate cross-layer mappings are important in order to answer these questions.
- *Alarm suppression and diagnosis.* Faults are commonplace in large-scale IP networks. During fail-

ure events, alarms are generated from various network elements (e.g., optical equipment, routers etc.), sometimes at different layers to indicate the failure. For example, a single fiber cut can cause the router to raise alerts indicating the interface is down at SONET, PPP, IP, and MPLS layers in addition to the loss of signal (LOS) alarms raised by certain optical components . If multiple links are affected due to SRLGs, all of the links, and potentially their associated optical components, raise alarms at all impacted layers overwhelming the network operator. Accurate associations are required to group these alarms together into a single event. Further, the accuracy of diagnosis (either manually or through automated correlation tools [4, 5, 6]) of these alarms is limited by the consistency of the IP-to-optical database. Accurate associations are also critical in proactive root-cause analysis of other performance related problems such as chronic intermittent flapping of interfaces, link degradation, etc., that potentially may have not (yet) triggered alarms.

- *Maintenance*. Network operators often gracefully remove the traffic on a link (by increasing the OSPF weight of a link or some such mechanism) before performing maintenance (e.g., repairing a faulty component, provisioning a new link, software upgrades, etc). Mis-associations across layers can cause operations to induce unwanted faults into the network. For example, if the IP-to-optical associations were wrong, operators intending to perform maintenance on a link between Los Angeles and San Francisco might instead inadvertently impact traffic flowing between San Diego and Los Angeles.

All of these applications rely on accurate cross-layer associations, the lack of which can seriously affect the overall network reliability.

2.2 Why is it hard?

It might appear to the reader that accurately maintaining such associations should be a straightforward task. After all, the network operators provision the network in a centralized manner; therefore, they can log these associations in databases. However, a live operational network incurs significant churn as links are provisioned, old equipment is replaced with new equipment, faulty components are repaired, interfaces are re-homed and so on. Database errors can result from this inherent churn - for example, if operations fails to update the relevant databases as an IP link is moved from a failed line card (slot) to a different, operational card (slot).

Additionally, this task is complicated by the presence of restoration at individual layers. For example, a failure within a SONET ring is recovered by rapidly protection switching to re-route the traffic the other way around the ring. In more "intelligent" optical networks, optical layer restoration will cause the path to re-route from the primary to an alternate path. These dynamic path changes at lower layers are typically achieving without impacting the upper layer connectivity; IP links are, by design, oblivious to restoration at lower layers. On one hand, one can argue that restoration in lower layers reduces the need or in some cases obviates the need for cross-layer visibility. While this is partially true, cross-layer visibility is still important because:

- IP layer might experience subtle changes in other performance metrics such as end-to-end delay;
- operations need to ensure that restoration itself does not have any problems;
- it is cheaper with the current technology to provide IP level restoration than optical; thus optical layer protection is often not used - particularly on high speed links[7].

This flux in topology can make it harder to diagnose failures or other performance issues without the presence of accurate cross-layer associations. One can, therefore, conceive that the network ought to be engineered to provide such information, perhaps by widening the interface between layers (e.g., exposing changes in optical topology to IP links), in the context of network management. While this conceptually clean design exposes such associations as a part of the network, we argue that this is not *practical or desirable* in the next section.

2.3 Fattening layers is not a good idea

A fat interface between layers allows information to flow from one layer to another layer as a part of the architecture itself. For example, if the network layer (IP/MPLS) were made aware of the underlying components in optical topology, this could allow the network layer to make better choices in recovering from failure situations. Indeed, in the context of fast restoration from failures in the MPLS domain, Interior Gateway Protocol (IGP) extensions in [8, 9] incorporate shared risk link groups (SRLGs) in their link state advertisements (LSAs). These SRLGs themselves could be either populated through management plane or auto-discovered through other means (such as through optical topology information obtained through link management protocols such as LMP [10]). This allows the computation of backup paths that are physically diverse from the primary paths. While such an approach has the clear advantage that cross-layer associations can be directly and accurately obtained from the network, we argue that this approach does not scale well. Some of the reasons are listed below.

- *Complexity*. Exposing lower-layer topology to upper layers adds complexity into the network (increased

processing due to new types of messages) and limits scalability (too many devices results in higher messaging overhead).

- *Interoperability*. Interoperability does not scale well with number of different types of devices; the larger the number of devices that need to be interoperable, the more difficult it becomes to achieve consensus on one protocol. Besides, it necessitates long design and testing cycles across large number of devices and manufacturers.
- *Security*. This additional visibility can affect the security of the network as one compromised network element (either physically tapping a fiber or through network exploits) can reveal a lot more details of the network (including lower layers).
- *Incompleteness*. Fundamentally it is difficult to achieve complete cross-layer visibility (e.g., automatically identifying proximity of two fibers, or two geographical properties such as fault lines, etc).

3 Architecture for cross-layer visibility

Rather than widening the boundaries between protocol layers, we argue that cross-layer visibility should be provided by the network-management layer as a *service*. After describing our architecture and its advantages, we explain how our solution accommodates the natural evolution of technology using three example protocol layers.

3.1 Cross-layer visibility as a service

Cross-layer visibility is primarily important for network-management applications, such as network design, planning, and troubleshooting. This argues for having thin interfaces between the layers *inside* the network, and providing the cross-layer views in the management system. As some ISPs do already today, we advocate that each AS have a management database (possibly distributed) that stores the topology at each layer and how a link at one layer maps into a set of components at the next lowest layer. For example, the database would store the IP topology (i.e., the routers and the links between them) as well as the forwarding paths between each pair of routers. The database would also store the optical topology and which sequence of optical components, such as fibers and amplifiers that form the link between two adjacent IP routers. Similarly, the database would keep track of which fibers run through the same conduit, as well as the geographic path the conduit traverses from one termination point to another. The database should have unique names for devices at each layer, as well as indices necessary to map between layers.

Today's management databases are a mixture of human-generated inventory and measurement data, with little compatibility from one AS to another; even in a single AS, the representation of data often changes over time, as the network design and measurement infrastructure evolve. The poor level of uniformity makes it exceptionally difficult to evolve a network, integrate two networks after an acquisition, or incorporate third-party network management tools. We believe that part of the problem is that the research and standards community has focused on defining the information that comes *out* of the network elements (e.g., SNMP Management Information Bases and Netflow measurement records), rather than what goes *in* to the database (e.g., IP topologies and traffic matrices). Often, the views needed by the network-management applications are not available in any one network element, and must be constructed by joining data from many parts of the network. In addition, there are multiple viable ways to construct these views, depending on the sophistication of the network elements and the monitoring infrastructure. Specifying only what goes *in* to the database allows the network technologies and monitoring infrastructure to evolve over time, while still providing cross-layer visibility.

In addition, we argue that cross-layer visibility should be provided as a *service* to a variety of clients. Today, traceroute is the primary way a customer determines the path its traffic takes through the network. Yet, traceroute is problematic for several reasons: (i) ISPs often disable or rate-limit ICMP to avoid overloading their routers, or to hide their topology information, (ii) the probes do not see the network elements at lower layers (e.g., inside an MPLS label-switched path, or the optical components between two routers), and (iii) analyzing changes in the path requires frequent probes to capture both the old and new paths. Instead, our management system could provide a "cross-layer traceroute" service, without customers probing the network directly. Similarly, the management system could support queries for network designers to identify shared risks and model the effects of failures on the flow of traffic through the network. Providing cross-layer visibility as an off-line service has several advantages:

- *Lower overhead on the routers:* Queries are answered by the management system, rather than the routers themselves. The system can also cache the results of recent or common queries, to reduce the overhead of satisfying future queries.
- *Answering historical questions:* By maintaining a log of network changes over time, the service can answer queries that require historical data. For example, a customer could inquire about a performance problem that started ten minutes ago, and the service could report whether a failure forced the customer's traffic onto a path with a longer round-trip time.
- *Application of security policies:* The management system can apply explicit policies to control what

kind of information is revealed, and to whom. For example, a customer may be allowed to ask if two paths have a shared risk, but not learn exactly what component is shared and where it is located. In addition, by forcing all queries through the service, the AS can protect its routers from probe traffic while still providing good network visibility to customers.

- *Flexible policies for defining shared risks:* The notion of a shared risk is extremely subjective [3], and the service can accommodate this by allowing queries at different granularities and incorporate extra information. For example, a network designer may want to know if two fibers lie near the San Andreas Fault in San Francisco. Or, one customer might be interested in link-disjoint paths and another in PoP-disjoint paths through the network.
- *Cooperation between ASes:* ASes could cooperate to provide greater visibility into shared resources. For example, an ISP that leases fiber from another provider could automatically learn the geographic path it follows (abstracted as deemed fit by the providers), or a multi-homed customer could determine its vulnerability to failures affecting both of its providers. Or, a governmental agency could conduct a realistic study of the effects of a serious catastrophe (such as a terrorist attack) on the Internet infrastructure.

With standard representations of the topology and paths at each layer, and the dependencies between layers, ASes can provide these kinds of valuable services.

3.2 Independent evolution of each layer

By defining the data *imported* by the management system, rather than *exported* by the network elements, our architecture supports many ways of learning the intra-layer topology and paths, and the cross-layer mappings:

IP topology and forwarding paths: The IP-level topology for an AS consists of routers and links, and a forwarding path consists of one or more sequences of IP links. The topology and paths can be learned in various ways, with different degrees of accuracy and timeliness:

- *Static view:* The topology can be recorded by the operators as equipment is installed, or reverse engineered from the router configuration state. The IP forwarding paths can be computed by modeling which paths the routers, as configured, would select. However, these static views do not capture which routers and links are unavailable at a given time.
- *Periodic snapshot:* A monitoring system can poll the routers for their status and forwarding tables, or run traceroute probes to map the topology. The forwarding paths can be computed on the measured topology, identified from the forwarding tables, or extracted directly from the traceroute results.
- *Continuous view:* A monitor could collect routing-protocol messages, field alarms when equipment goes up/down, or analyze syslog output generated by the routers, to provide an up-to-date view of the topology and paths. If the AS supports explicit routing (e.g., using MPLS label-switched paths), the management plane would know the forwarding paths because it was responsible for configuring them.

With our architecture, an AS can easily evolve its network design and monitoring infrastructure, while maintaining the same representation of the topology and paths in the external database and management applications.

Optical components and paths: The optical topology consists of a diverse array of devices, including fibers, amplifiers, cross connects, and add-drop multiplexors. The sequence of optical components underlying an IP link could be learned in various ways, depending on the sophistication of the optical components:

- *Completely manual:* The operators can keep track of optical components and their relationship to IP links as the equipment is installed. To reduce the likelihood of inaccuracies in the database, the AS can apply basic consistency checks, such as verifying that two ends of an IP link map to the same circuit identifier. As a second line of defense against errors, the AS can monitor the effects of optical failures on the IP layer to identify and apply correlation algorithms to identify incorrect mapping information [6, 5].
- *Partially automated:* Manually constructing the list of optical devices underlying a link is not sufficient if any of the underlying components adapt automatically to failures. For example, an intelligent optical cross-connect may reroute the traffic through an intermediate cross-connect when a component along the direct path has failed. Similarly, a SONET ring may adapt by redirecting traffic around the ring in the opposite direction. Capturing these changes requires logging of alarms or periodic probing of the adaptive components and correlation across layers[1]. Still, some parts of the database may remain human-generated, such as the identity of the ingress and egress cross connects, or the list of optical amplifiers between any two cross-connects.
- *Completely automated:* Discovering the optical components becomes much easier if the network elements have a common control plane, such as Generalized MPLS (GMPLS) [11]. For example, GMPLS

[1] Although automatic restoration protects the IP layer from optical failures, knowing the new mapping is important for troubleshooting performance problems (e.g., a sudden increase in round-trip times) and identify new shared risks. As an added benefit, these automatic routing changes at the optical level also provide opportunities to identify mistakes in the human-entered databases, while reducing the effects of the failure from the IP layer.

includes a Link Management Protocol (LMP) [10] that performs neighbor discovery between adjacent network elements so they can dynamically establish a light path from one router to another. LMP provides the names and attributes of the optical components, obviating the need for human-generated databases to map between the IP and optical levels.

In our architecture, an AS can gradually deploy more intelligent optical devices and new auto-discovery protocols, while maintaining the same representation of the path through the optical layer between two routers.

Fiber and fiber spans: A fiber map captures the topology of the underlying transport network. A fiber consists of multiple *spans*, a segment of fiber traversing a single conduit; a fiber span, in turn, consists of multiple fibers traversing the same conduit. This information could be learned in various ways:

- *Completely manual:* As with other optical components, the operators can keep track of the location of fiber and the mapping to/from spans as the fibers are installed, or leased from other providers. The failure of fiber spans (e.g., due to a physical cuts), as they occur, provide an opportunity to identify incorrect mappings. Measurements of propagation delay across a link (and comparison with the supposed fiber path) is another way to detect serious inconsistencies.
- *Intelligent conduits:* Since fibers are passive devices, they do not automatically advertise their operational status (e.g., Loss of Light), presence in a particular conduit, or the physical paths they traverse. Creating new techniques for auditing the management database, or even automatically generating the data, is an exciting direction for future research. We envision several possible approaches, including:
  - *Active devices at conduit end-points:* Optical amplifiers along the optical path could report their identity and geographic location [12]. In addition, the individual fibers could have RFID tags where they enter and leave and conduit.
  - *Active devices along the conduit:* For even higher accuracy, the conduits could have active devices, such as audio or wireless transmitters, placed at fixed intervals. These devices could be coupled with GPS receivers (to allow the devices to broadcast their geographic locations), or a separate measurement system could analyze the signal strength to aid in locating the devices. Closer spacing of these devices would provide more fine-grain data, at the expense of higher cost.
  - *Multi-layer packet monitoring:* To verify the mapping of fibers to IP links, we could envision a new generation of packet monitors that combine IP packet capture, reading of audio or RFID tags, and reporting of geographic positioning information. For example, a packet monitor could be used to tap a fiber and analyze the IP packet stream, perhaps on a per-wavelength basis. By capturing the routing protocol messages (e.g., OSPF HELLO messages or link-state advertisements), the monitor can determine the IP addresses of the routers on either end of the associated IP link. Over a period of time, the packet monitor could be installed at various points in the network to collect accurate mappings of IP links to/from fibers (and fiber spans) to check and update the information in the database.

In our architecture, the management database would store the mappings of fibers to spans, as well as the geographic path of the spans (at some known level of accuracy), however they are determined.

4 Conclusion

This paper addresses the challenges of providing cross-layer visibility to network-management applications, and advocates against expanding the interfaces between layers for auto-discovery of the cross-layer associations. Instead, we propose an architecture where such associations can be learned or maintained automatically, not by widening the layers, but by defining the data that should be imported into a management database. The architecture provides cross-layer visibility as a service to other applications and users that depend on this information.

References

[1] "Update: CSX train derailment, thread on NANOG mailing list, `http://www.merit.edu/mail.archives/nanog/2001-07/msg00367.html`," July 2001.

[2] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "A simple network management protocol (SNMP)," RFC 1157, Internet Engineering Task Force, May 1990.

[3] John Strand, Angela Chiu, and Robert Tkach, "Issues for routing in the optical layer," in *IEEE Communications Magazine*, Feb. 2001.

[4] SMARTS Inc., "`http://www.smarts.com`,".

[5] Ramana Kompella, Jennifer Yates, Albert Greenberg, and Alex C. Snoeren, "IP fault localization via risk modeling," in *Proc. Networked Systems Design and Implementation*, May 2005.

[6] Srikanth Kandula, Dina Katabi, and Jean Philippe Vasseur, "Shrink: A tool for failure diagnosis in IP networks," in *Proc. ACM SIGCOMM MineNet Workshop*, Aug. 2005.

[7] G. Li, D. Wang, R. Doverspike, C. Kalmanek, and J. Yates, "Economic analysis of IP/Optical network architectures," in *Proc. Optical Fiber Communication Conference*, Mar. 2004.

[8] D. Katz, K. Kompella, and D. Yeung, "Traffic engineering extensions to OSPF version 2," RFC 3630, Sept. 2003.

[9] H. Smit and T. Li, "Intermediate system to intermediate system (IS-IS) extensions for traffic engineering (TE)," RFC 3784, June 2004.

[10] J. Lang, "Link management protocol (LMP)," in *Internet Draft, draft-ietf-ccamp-lmp-10.txt*, Oct. 2003.

[11] E. Mannie, "Generalized multi-protocol label switching (GMPLS) architecture," RFC 3945, Oct. 2004.

[12] Panagiotis Sebos, Jennifer Yates, Dan Rubenstein, and Albert Greenberg, "Effectiveness of shared risk link group auto-discovery in optical networks," in *Proc. Optical Fiber Communication Conference*, Mar. 2002.