Proactive Network Management of IPTV Networks

R. K. Sinha, K. K. Ramakrishnan, R. Doverspike, D. Xu, J. Pastor, A. Shaikh, S. Lee
AT&T Labs - Research, New Jersey, USA

Abstract: Consumer communications and entertainment services, including broadcast TV and VoIP require service providers to meet stringent availability and latency constraints. When a packet technology, such as IP, is used to transport these services, this also poses stringent packet loss requirement on the network. This aspect of IPTV, where impairments have consumer-visible impact and potential public relations consequences, creates new challenges in protocol design, as well as network management. The key to operating an effective network is to expand beyond the typical "reactive" network management approach to be able to anticipate and manage potential network problems. This paper describes network management techniques deployed in a production IPTV network with over 2 million customers.

1. Introduction

Multiple service providers have deployed IPTV for a few years now for distribution of both live TV (so-called "linear" TV) as well as on-demand delivery of video content. Unlike an Internet Service Provider (ISP) who provides residential broadband Internet service on a *best-effort basis*, it is critical for an IPTV service provider to ensure high quality of the service. Quality has two key aspects: providing sufficient network availability to ensure that the impact on video perception resulting from end-to-end packet loss is tolerable for the viewer and delay (latency) experienced by a user viewing the TV content is properly managed. For example, if restoration from a failure occurs within 50 milliseconds, there is no user-perceived impairment as a result of the failure. Many of the issues of running and managing the network *end-to-end* (i.e., from the national video source to the user set-top box) are similar to what an ISP applies to other services. However, here the high sensitivity of the IPTV application to impairments creates a new network management challenge for providers. The key to running an effective IPTV network is to go beyond "reactive" network management to *anticipate and manage potential network problems*. This requires both careful multi-layer protocol design and anticipatory network management.

The IPTV networks with a large customer base are primarily in South Korea and in Europe, especially France, Germany, Spain, and Italy, but they span a relatively small geographical area. In this paper we discuss the philosophy, protocol design, network management techniques, and tools we use in one of the largest commercial IP networks that provides residential TV services via IPTV technology and IP multicast, Voice-over-IP (VoIP), and broadband Internet services.

In this environment a key need is a coordinated design and operation across all the protocol layers. The IPTV system design we describe incorporates a protocol design that is aware of information across multiple network and protocol layers. For example, individual link outages between routers/switches are recovered (within 50 milliseconds) using a version of MPLS Fast Reroute (FRR) [6]. As such, these outages are essentially transparent to the higher layers. The requirement to recover link failures within 50 milliseconds arises because the higher layer mechanisms (FEC, retransmission based recovery of lost packets) have a capability to handle burst losses of relatively small magnitudes and with reasonable delays. Higher delays may result in unacceptable user-perceived impairment.

In addition to reacting to single outages, we have developed an enhancement to the basic protocol design that better anticipates more complex series of outages. A typical outage may require one or more hours to repair or normalize. Therefore, because of the stringent requirements on reliability for the IPTV network, subsequent outages (resulting in a network state with multiple concurrent outages) are not rare enough to be ignored. This is especially true when one considers that network outages occur from two major sources: 1) network component failures or fiber cuts and 2) planned network maintenance or reconfiguration activity. For example, a link or router may be taken down for maintenance (e.g., to reconfigure the topology for growth/expansion of the network). During that period there is a small (but not negligible) probability of a component failure. E.g., analysis of link outages from a large commercial IPTV deployment over a four month period revealed that in 17% of cases, at least 2 links were down concurrently, and in 2% of the cases, 3 or more links were down concurrently. Thus, we see it is important to plan for potential outages after an initial outage [9].

Given this approach, our network management framework must also be designed to collect and analyze information across multiple protocol layers. This information allows us to monitor and react to issues we observe in real-time plus, anticipate and avoid potential problems that may occur due to subsequent outages and the potential onset of network congestion. For example, most network operators consider FRR to be a "distributed" and automatic "self-healing" network capability and, thus, are usually unaware of the actual state of the network. The latter awareness is especially critical in a nationwide IPTV backbone because, to offer IPTV services efficiently enough to compete with

more traditional broadcast technologies, such as cable TV providers, it employs IP multicast [4] for distribution of broadcast TV content. This results in a highly efficient (low capacity/low cost) network, but one that requires very careful restoration planning/design and management. For example, when link-based FRR responds to an outage by rerouting traffic over a path of alternate routers and links, the multicast tree can undergo significant change. As described earlier, if a subsequent outage occurs, because of the highly efficient design, there is a potential for path overlap and hence congestion, if the network management systems, tools and, ultimately, personnel are unaware of this condition.

We have approached this problem by designing a tool, ***Birdseye***, that visualizes both the backbone network, status of its links (down, planned maintenance, up, in FRR backup mode), its multicast tree, and link congestion. This alerts the network manager about the complete routing and performance of the underlying network. In addition, at the highest layer, user perceived video quality of the video is monitored from a centralized Network Operation Center (NOC) by tunneling into the Video Hub Office (VHO) that serves each metro area. If problems are reported or observed, yet everything appears acceptable in the backbone network, then the network manager can home into other causes, e.g, VHO servers or metro network.

In Section 2 we describe the network and the restoration protocol. We describe the various data sources for Birdseye in the Section 3. Finally, in Section 4, we describe several network states to illustrate the operations of Birdseye and how it alerts the operators.

2. Description of the network

AT&T's IPTV network distributes video, internet, and VoIP to over two million customers in the continental US (as of Jan 2010). Figure 1 (copied from [3]) shows a simplified network architecture.

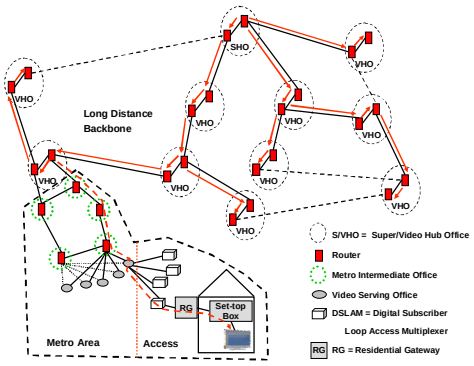

Figure 1. Simplified network architecture

The backbone, consists of a super *hub office* (SHO) and a large number of VHOs, one at each metro. Video content is gathered from the national content providers at the SHO and is distributed to VHOs using PIM-SSM [1]. An Internal Gateway Protocol (IGP) such as OSPF [5] distributes topology and routing changes. Each VHO in turn feeds its metropolitan area. Our focus in this paper is on the backbone portion of the network. The metro network, broadband Internet, and VoIP services, are beyond the scope of this paper. See [3].

2.1 Protocol Design for Failure Restoration

Figure 2 (as shown in [9]) is an example of how the network interfaces are set up to be able to quickly recover from failures. Each rectangular box represents a router at a VHO with the 'root' being the SHO. Figure 2 shows the network operation when there is no failure. Focusing on router pairs E and C, we observe four Internal Gateway OSPF adjacencies between this router pair: two unidirectional (or ***directed***) pseudowires (dashed lines between nodes E and C) and two unidirectional physical-layer "PHY" links (solid lines between nodes E and C). These pseudowires are associated with a primary path and a corresponding backup path (which are typically MPLS label switched paths (LSPs)). The primary path for each pseudowire is its corresponding PHY link and the backup path routes over other PHY links that are disjoint from the primary path. The OSPF link costs (which we generically refer to as ***weights***) on the pseudowire links are lower than those for the PHY links. This causes the OSPF shortest path algorithm to primarily route over the pseudowire links rather than the PHY links in a non-failure state. By routing over the pseudowire links, the protocol enables that OSPF shortest paths and PIM multicast tree are unaffected as long as the pseudowire remains up.

Figure 3 illustrates the case of a successful FRR restoration, where both of the PHY links between E and C (say both directions are impacted) fail. Upon this failure being detected, both PHY links are taken out of service and the two pseudowires are switched from their primary path to their backup paths. This is an entirely local decision made by end point routers E and C. The use of the backup path involves minimal protocol exchange with the entities over the path and thus the time to restore from a failure is primarily driven by the time to detect the failure of the physical link. Note the path from the Root to node A now switches to the backup path at node E (E-A-B-C), reaches node C, and then continues on its previous (primary) path to node A (C-B-F-A). The switching of pseudowire to backup paths occurs well before the OSPF timers expire (as a result of missing Hellos). In fact, the Hello messages to maintain the adjacency now continue to flow over the backup path. *OSPF is unaware of the actual routing over the backup path.* Therefore, when the OSPF Link State Advertisements (LSAs) are broadcast, although they show that the PHY links are down, the states of the pseudowire links remain unchanged and (because of the lower weight of the pseudowire compared to PHY link) there is no change to the OSPF shortest path tree and thus the multicast tree.

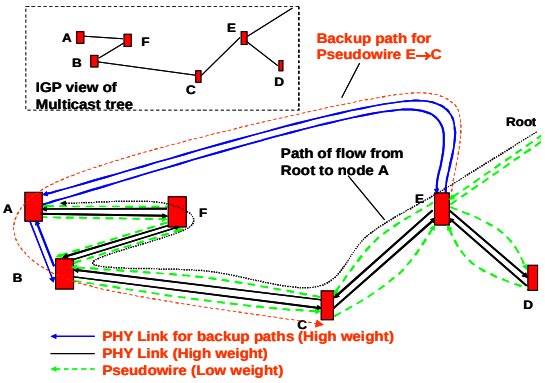

Figure 2. A network segment with pseudowires, PHY links, and a sample FRR backup path

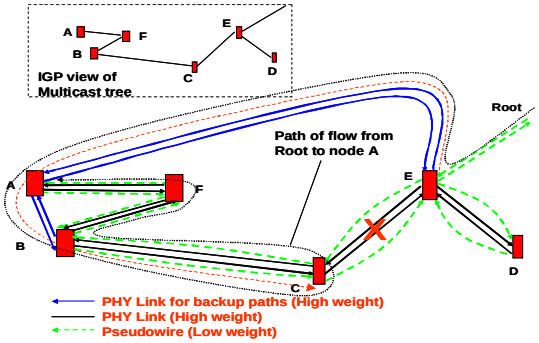

Figure 3. Single link failure: FRR backup path is put into work and OSPF is unaware of the failure

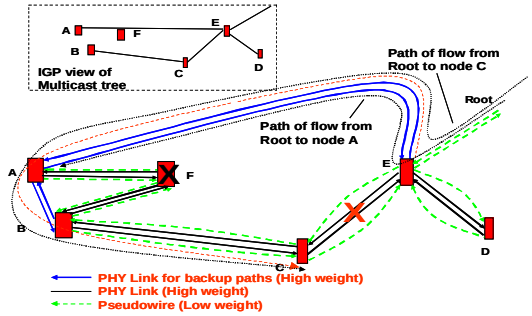

Figure 4. Multiple failures with one link and one router failing: FRR backup path overlaps with multicast traffic

Figure 4 shows a double failure situation where node F also fails before PHY link E-C is repaired (it could also be the case that node F is being considered for maintenance during the time that the link E-C is still awaiting repairs). In this case FRR is not sufficient to repair the failure and thus a new set of OSPF shortest paths and a new multicast tree needs to be computed, as shown. There is a subtlety in this case. From OSPF's perspective, there is no difference between the case where the pseudowire between E and C goes over the PHY link or over the backup path. In this case, while the backup path between E and C is active, the flows from the Root to node A in the new multicast tree and the backup path of E→C used in the Root to node C flow overlap on the link E→A. This has the potential for congestion related losses, which is undesirable.

Reference [9] proposes a cross-layer approach to reconfigure the multicast tree after the FRR based restoration from the initial failure so that this kind of overlap is avoided, thus preventing congestion on the link E→A.

3. Integration of Multiple Data Sources

Our web based tool Birdseye synthesizes and interprets data from a variety of data sources described below. Our overall philosophy is to:

(a) Display the network in its idealized, failure-free state to operators and network designers. Furthermore, provide important information for network management decisions, such as link capacities and the failure-free traffic routing.

(b) Maintain real-time state from the tools described in section 3.2. These include real-time topology discovery, link utilization, Border Gateway Protocol (BGP) reachability information using BGP routing tables, and a database of links planned to go under scheduled maintenance.

(c) Parse differences between the failure-free topology and the real-time topology to determine the key topology changes an operator should be aware of. We also alert the operator to link congestion and potential pitfalls of scheduled maintenance activities.

For item (a), we note that because of the need to interpret the planner's intent and occasionally inconsistent data, calculating the idealized topology in real-time is not as readily obvious as one might first think; therefore, it is obtained by applying logic to the data obtained from the topology discovery tool, historical data, and auxiliary tables that represent planned network topology changes.

3.1 Creating a failure-free network view

We use the NetDB tool [2] to build a failure-free topology of the network by analyzing router configuration files. Note that as Figures 2 through 4 illustrate, the term "topology" in packet networks is, in fact, often a logical concept for the purpose of layer-3 routing. Various links can be created between routers and given routing weights. These links can be physical (encountering no intermediate routers) or logical in nature. For example, NetDB compiles all the interfaces in the example network with various attributes defined in the configurations. Table 1 lists some of the interfaces used in the example network in Figure 2. Interfaces names starting with 'P' correspond to PHY links

and those starting with 'SDP' correspond to MPLS pseudowires. The first row shows the interface P1 on router E with IP address 10.1.1.10 and OSPF weight 1000. First, we infer links between interfaces based on the IP addresses and subnet masks (e.g., between P1 in E and P1 in C, and between SDP2 in E and SDP2 in C). This set of inferred links, shown in Table 2, serves as the base topology. As we discussed earlier, note that the OSPF weight of PHY links, 1000, is higher than that of the pseudowires (which have an OSPF weight 10), which causes OSPF routing to prefer paths using the pseudo wire (SDP) rather than the physical link (P).

Table 1. Interface table for the example network

Router	Interface	IP	Subnet Mask	OSPF Weight
E	P1	10.1.1.10	30	1000
E	SDP2	10.2.1.6	30	10
C	P1	10.1.1.9	30	1000
C	SDP2	10.2.1.5	30	10
E	P2	10.1.1.18	30	1000
A	P2	10.1.1.17	30	1000
A	P1	10.1.1.1	30	1000
B	P1	10.1.1.2	30	1000
B	P2	10.1.1.5	30	1000
C	P2	10.1.1.6	30	1000
…				

Table 2. Set of inferred links between E and C

Source	Dest	Source Intf	Dest Intf	Source IP	Dest IP	OSPF Weight
C	E	P1	P1	10.1.1.9	10.1.1.10	1000
C	E	SDP2	SDP2	10.2.1.5	10.2.1.6	10
E	A	P2	P2	10.1.1.18	10.1.1.17	1000
A	B	P1	P1	10.1.1.1	10.1.1.2	1000
B	C	P2	P2	10.1.1.5	10.1.1.6	1000

Another part of the router configuration specifies the interface IP for each hop in the primary and secondary paths of pseudowires. Using Tables 1 and 2, these IP addresses can be translated into a set of links. E.g, the primary path for pseudowire on interface SDP2 in E uses the PHY link to C with interface P1. On the other hand, the secondary path takes a three-hop route, first going from P2 in A(10.1.1.17), to P1 in B(10.1.1.2), and finally to P2 in C(10.1.1.6). In our system, based on a regular configuration feed from a production system, we construct the topology and the data tables on a daily basis.

3.1.1 Dealing with incomplete data

We list the two main reasons that the network data from NetDB can sometimes be incomplete.

(a) NetDB gets a snapshot of router configuration files. If any links or routers have an outage at that point, they will be missing from the NetDB data. We overcome this by taking a union across the snapshots of the last several days. We use multiple heuristics to deal with inconsistencies in the data from different extracts. E.g., if a given link has (OSPF) weight 100 in one snapshot and 100,000 in another snapshot, we infer that the weight of 100,000 meant that the link was temporarily "costed out" for maintenance and therefore 100 is the correct weight. On the other hand, if weights are 100 and 150 in two different snapshots, we resolve in favor of the most recent snapshot.

(b) There is a delay for a newly added link to get populated into NetDB. To deal with this delay, we maintain an auxiliary table of links in the real-time topology that are not in the failure-free topology. We add this auxiliary set of links to the failure-free topology. Once we see any of these links in the NetDB topology, we purge them from the auxiliary table.

3.2 Data sources for real-time network state

We now describe the data sources used for collecting the real time topology, link utilization, number of BGP routes, and list of links scheduled to undergo maintenance.

3.2.1 Tracking Real Time topology from OSPFMon

OSPFMon [7] is used for collecting OSPF LSAs from the network. It establishes a partial adjacency with one or more routers to receive LSAs. Apart from processing the streaming LSAs to identify and log network events (e.g., router up/down, link up/down etc.), OSPFMon generates periodic snapshots of the network topology (list of routers, list of links between them etc.) in real-time. The snapshots are generated when a network event occurs. Most other carrier-based network management implementations access messages or alarms provided by an upstream network management interface, such as an Element Management System or Simple Network Management Protocol (SNMP) or Command Line Interface. We have found through experience that this introduces a layer of interpretation and ambiguity, plus reliance on the switch vendor's network management features. In contrast, one can view the OSPFMon data collection platform as another network element actually attached to the network. Therefore, we see the ***exact*** same control plane messages that the switches see. To our knowledge, OSPFMon is unique in this regard.

The real-time topology (snapshot from OSPFMon) is then compared to the failure-free topology (after rectification of NetDB and auxiliary tables). Links present in the

failure-free topology, but missing in the real-time topology are deemed to have failed.

To avoid overwhelming the system when conditions in the network is changing very rapidly, OSPFMon implements a throttling mechanism, wherein no more than one snapshot is generated per 30 second window. Furthermore, a new snapshot is generated even when the network is stable – i.e., no event has occurred -- for a period of six hours.

3.2.1.1 Dealing with Loss of Updates

There are two ways for Birdseye to obtain real-time topology data from OSPFMon: the LSA event stream or the topology snapshots. The LSA event stream has the advantage that Birdseye gets what it indeed needs – incremental changes to the network topology. The disadvantage is that 1) we need to implement a reliable interface (i.e., with handshaking) to the OSPFMon platform to ensure that no messages are missed and 2) must piece-by-piece compile the incremental information into a full topological view. We decided to use the snapshots because every event in the network creates a new topology snapshot which has information for *all* links. In addition, snapshots are generated periodically so that Birdseye can "catch up" with any missing data. Of course, one limitation of this approach is the delay in visualization (and alert) caused by the throttling mechanism mentioned earlier. However, after a year of actual network management experience, network operators have found this interval acceptable.

3.2.2 Monitoring Link utilization

The Multi-Router Traffic Grapher (MRTG) [8] polls the router nodes (typically every 5 minutes) using SNMP to collect link utilization information. Link congestion is an important indicator to operators who monitor the network because congestion can impact video service similarly to a link outage.

Links can get congested when, as outlined in Section 2.1, a backup path's flow overlaps with the multicast flow. Another situation of interest is when switch pairs have parallel links between them. MRTG is used to monitor how the link load balancing algorithm in the switch is operating, how the multicast traffic is mixing with the unicast traffic, and how the High Definition (HD), Standard Definition (SD), and Picture-In-Picture (PIP) streams are being load-balanced. By analyzing link utilization, Birdseye alerts the operator whether congestion is because of load imbalance or due to multicast traffic overlap from an outage.

3.2.3 BGP routing table information

In addition to video distribution, IPTV networks are typically shared as an access network providing internet connectivity to the customers as is the case in most triple play service offerings. In this network the Internet service also utilizes the same network connectivity from the residential gateway at the customer premise to the VHO. The internet-bound streams are differentiated from video and voice and are treated as best-effort traffic. This stream diverges at the VHO and is routed to an ISP edge router, also called a Provider Edge (PE)-router. To monitor the health of this service, we look at the BGP sessions at each VHO for signature alarms, such as when BGP session with the route reflector goes down or when the link to the access routers goes down. In addition, we also monitor the number of routes present at the access routers and look for abnormal deviations from set thresholds.

3.2.4 Link maintenance information

Finally, a good example of proactive network management is where the Birdseye system maintains an interface to an Operations Support System (OSS) that schedules network maintenance activity. Maintenance activity is often planned far in advance and is often agnostic to real-time network state. The links or switches that are planned to be affected are displayed on the Birdseye graph visualization within a prescribed time frame of their due date. If a link is down and a maintenance procedure is instigated, this could cause a serious network problem, much akin to a multiple network failure. In fact, on several occasions network operators averted significant service outages in AT&T's network because they were alerted to these potential network conflicts by Birdseye. These few examples more than pay for the research and justify the approach described in this paper.

4. The Birdseye Network Visualization Tool

We illustrate how the Birdseye tool harnesses the multiple information flows across the layers and provides key information needed for proactive network management. Space does not permit us to describe all of its features and capabilities, so we will focus on a few key ones.

The Birdseye visualization tool is deployed in the AT&T Network Operation Center and is actively used by network operators to monitor and perform proactive network management for the U-verse™ backbone network. To demonstrate some of the features described above we created a hypothetical topology and hypothetical outages. The actual U-verse backbone is significantly larger, more complex and has a substantially richer connectivity than this hypothetical topology. Figures 5 and 6 are actual screen shots from the tool for this hypothetical network and outages. To save space we show three concurrent link failures (between routers in Cleveland-Cincinnati, New York-Wash. D.C., and Albuquerque-Knoxville) and a planned maintenance event (Raleigh-Wash. D.C., shown by a "hard hat" symbol) on a single network snapshot.

Clearly, if one were to instigate maintenance activity while the New York-Washington D.C. link is down, the Washington D.C. metro would become isolated from the backbone and those customers would lose all national streaming content. We guide the operator with appropriate warnings to avoid/postpone such a maintenance event.

To visualize an FRR event, the physical link from Albuquerque-Knoxville (shown in dashed green) has failed and the corresponding pseudowire was successfully rerouted to its backup path (shown in dashed red). As the visualization shows, the backup path for the pseudowire routes through routers in Dallas and Atlanta. The highlighted backup path alerts operators to cancel any

scheduled maintenance on those links because they are carrying multicast traffic. These links are also vulnerable because their subsequent failure would fail the Albuquerque-Knoxville pseudowire and result in a slower OSPF/PIM reconvergence (compared to FRR).

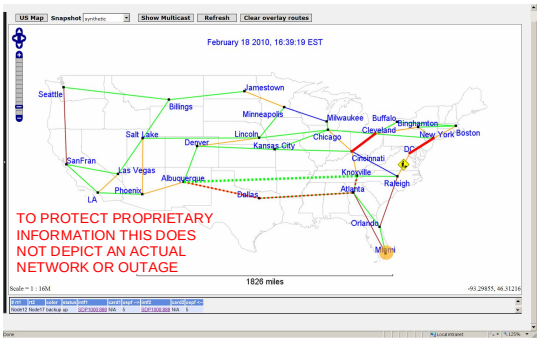

Figure 5: Geographical based visualization of Topology with three concurrent link failure

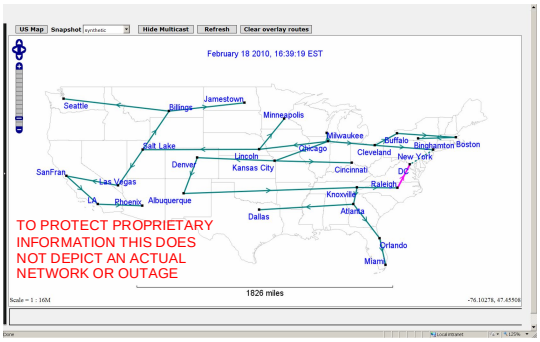

Figure 6: Resulting multicast tree derived from network state

For the other two failed physical links (Cleveland-Cincinnati and New York-Wash. D.C.) the pseudowires have also failed (shown in thick red) because links on their backup paths have failed and thus FRR was unsuccessful. However, as described, the network indeed reconfigures its routing when the pseudowires are not restored and the resulting multicast tree represents this rerouting. Figure 6 displays the resulting multicast tree, derived using the surviving links, rerouted pseudowires, and link weights of this hypothetical topology.

Note that the root of the tree is in Chicago and this multicast view shows how the tree has deviated from its ideal (non-failure) state. In this case, the only change is that Wash. D.C. is now a leaf node with Raleigh as the parent, instead of being a child node of New York. Note that if one carefully examines the routing and FRR paths in Figures 5 and 6, to get to Dallas, multicast packets flow through routers in Kansas City, Denver, Albuquerque, Dallas, Atlanta, Knoxville, back to Atlanta and finally to Dallas. This illustrates the routing on a national geographical scale analogous to the example of Figure 3.

The orange circle over Miami in Figure 5 is an alert to the operators about BGP connectivity problems affecting customer broadband internet service and Voice-over-IP (VoIP) telephony service in the Miami metro area.

With any of these alerts, selecting the alert brings out additional information. Finally, the network operators would also carefully monitor the link utilization and load balancing in such a situation, as provided by the MRTG tool described earlier.

5. Summary

The key to operating an effective IPTV network, where even small impairments can have consumer-visible impact and potential public relations consequences, is to expand beyond the typical "reactive" network management approach to be able to anticipate and manage potential network problems. We described our network management approach used to manage AT&T's backbone IPTV network, with over two million subscribers. By carefully distilling network data (historical, real-time, and anticipation of potential problems) from multiple AT&T-developed systems (such as OSFPMon, NetDB, and maintenance scheduling) and public tools such as MRTG into a comprehensive visualization tool, called Birdseye, we provide an indispensible platform for network operators to achieve the key objectives for the service.

6. References

[1] S. Bhattacharyya, ed., An Overview of Source-Specific Multicast (SSM), IETF RFC 3569, July 2003; www.ietf.org/rfc/rfc3569.txt.

[2] D. Caldwell, A. Gilbert, J. Gottlieb, A. Greenberg, G. Hjalmtysson, J. Rexford, The cutting EDGE of IP router configuration, ACM HotNets Workshop, 2003.

[3] R. Doverspike, G. Li, K. Oikonomou, K.K. Ramakrishnan, R.K. Sinha, D. Wang, C. Chase, Designing a reliable IPTV network, IEEE Internet Computing, May/June 2009, pp 15-22.

[4] D. Estrin et al., Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, IETF RFC 2362, June 1998; www.ietf.org/rfc/rfc2362.txt.

[5] J.T. Moy, OSPF Anatomy of an Internet Routing Protocol, Addison-Wesley, 2000.

[6] P. Pan, G. Swallow, and A. Atlas (Editors), "Fast reroute extensions to RSVP-TE for LSP tunnels," IETF RFC 4090, 2005.

[7] A. Shaikh, and A. Greenberg, OSPF Monitoring: Architecture, Design and Deployment Experience, USENIX NSDI, 2004.

[8] S. Shipway, Using MRTG with RRDtool and Routers, Cheshire Cat Computing publication, April 2009.

[9] M. Yuksel, K. K. Ramakrishnan, R. Doverspike, R. K., Sinha, G. Li, K. Oikonomou, and D. Wang, Cross-Layer Techniques for Failure Restoration of IP Multicast with Applications to IPTV, COMSNETS 2010, Jan 2010.