Integration of QoS Metrics, Rules and Semantic Uplift for Advanced IPTV Monitoring

de Fréin[†], Ruairí and Olariu, Cristian and Song, Yuqian and Brennan, Rob and McDonagh, Patrick and Hava, Adriana and Thorpe, Christina and Murphy, John and Murphy, Liam and French, Paul

[†]KTH - Royal Institute of Technology, Stockholm,
Sweden
web: https://robustandscalable.wordpress.com

in: Journal of Network and Systems Management. See also BibTeX entry below.

BibTeX:

```
@article{deFrein14Integration,
    author={de Fr{\'e}in$^\dagger$, Ruair{\'i} and Olariu, Cristian and Song, Yuqian and Brennan, Rob and McDonagh, Patrick
    and Hava, Adriana and Thorpe, Christina and Murphy, John and Murphy, Liam and French, Paul},
    title={Integration of QoS Metrics, Rules and Semantic Uplift for Advanced IPTV Monitoring},
    journal={Journal of Network and Systems Management},
    year={2014},
    volume={23},
    number={3},
    pages={673--708},
    issn={1573-7705},
    doi={10.1007/s10922-014-9313-9},
    url={http://dx.doi.org/10.1007/s10922-014-9313-9},
}
```

J Netw Syst Manage (2015):JONS-743-R2 manuscript No.
(will be inserted by the editor)

Integration of QoS Metrics, Rules and Semantic Uplift for Advanced IPTV Monitoring

**Ruairí de Fréin · Cristian Olariu · Yuqian Song ·
Rob Brennan · Patrick McDonagh ·
Adriana Hava · Christina Thorpe · John Murphy ·
Liam Murphy · Paul French.**

Received: August 14, 2012 / Resubmitted: January 29, 2014 / Accepted: March 16, 2014

Abstract Increasing and variable traffic demands due to triple play services pose significant Internet Protocol Television (IPTV) resource management challenges for service providers. Managing subscriber expectations via consolidated IPTV quality reporting will play a crucial role in guaranteeing return-on-investment for players in the increasingly competitive IPTV delivery ecosystem. We propose a fault diagnosis and problem isolation solution that addresses the IPTV monitoring challenge and recommends problem-specific remedial action. IPTV delivery-specific metrics are collected at various points in the delivery topology, the residential gateway and the Digital Subscriber Line Access Multiplexer (DSLAM) through to the video Head-End. They are then pre-processed using new metric rules. A semantic uplift engine takes these raw metric logs; it then transforms them into World Wide Web Consortium (W3C)'s standard Resource Description Framework (RDF) for knowledge representation and annotates them with expert knowledge from the IPTV domain. This system is then integrated with a monitoring visualization framework that displays monitoring events, alarms, and recommends solutions. A suite of IPTV fault scenarios is presented and used to evaluate the feasibility of the solution. We demonstrate that professional service providers can provide timely reports on the quality of IPTV service delivery using this system.

Neither the entire paper nor any part of its content has been published or has been accepted for publication elsewhere. It has not been submitted to any other journal.

R. de Fréin & C. Olariu
TSSG, Waterford Institute of Technology, Waterford, Ireland
Corresponding author: R. de Fréin, email: *rdefrein@gmail.com*, Ph.: +353 51 83 4002, fax: + 353 51 34 1100
URL: `http://www.tssg.org/about/people/dr-ruairi-de-frein/`. C. Olariu: *colariu@tssg.org*

R. Brennan & Y. Song
Trinity College Dublin, Ireland
R. Brennan: *rob.brennan@cs.tcd.ie*, Y. Song: *yuqians@scss.tcd.ie*

P. McDonagh, A. Hava, C. Thorpe, J. Murphy & L. Murphy
School of Computer Science and Informatics, University College Dublin, Dublin, Ireland
P. McDonagh: *patrick.mcdonagh@ucd.ie*, A. Hava: *adriana.hava@gmail.com*, C. Thorpe: *christina.thorpe@gmail.com*, J. Murphy: *j.murphy@ucd.ie*, L. Murphy: *Liam.Murphy@ucd.ie*

P. French
IBM Tivoli Division, Kinsale Road, Cork, Ireland
P. French: *paulfrench@ie.ibm.com*.

Keywords Network and System Monitoring · Measures · Quality of Service · IPTV · Monitoring · Semantic Uplift

1 Introduction

Monitoring IPTV services presents significant research challenges and business opportunities, particularly if such monitoring can be used for Customer Experience Management (CEM). Cisco [1] predicts that network traffic volumes in the order of tens of exabytes are not that far off [2]; given that 90% of the bits transmitted on the Internet will be video related and that the number of consumers of these bits will soon exceed one billion, monitoring customer experience will be crucial to safeguard revenues. IPTV's deployment investment to address this opportunity is two-fold, consisting of both improvements in the underlying infrastructure as well as provisioning and managing the vast data centers needed to provide IPTV [3]. Recent work has explored strategies for coordinating the allocation of resources for multiple virtual IPTV providers to maximize revenue [4] and routing strategies to manage network resources when multiple IPTV services are overlaid on the same network [5]; however, the fundamental problem –indeed an integral part of satisfying customer expectation– lies in evaluating the quality of the IPTV service being provided and then giving guidance on how delivery can be improved. In this paper, we take a first systemic view of IPTV monitoring: we consider what metrics should be collected, where metrics should be collected, and how these metrics should be presented to a Network Manager (NM) in a semantically enriched way via standard Real-Time Control Protocol (RTCP) reports. The challenge our system addresses is how to allow the NM to drill-down and investigate IPTV anomalies and outages in more semantically enriched detail and also suggest a corrective action that is meaningful.

IPTV faces stiff competition from technologies for digital TV delivery [6] –which are described by the DVB standards and are considered to have high reliability– as the networks they use are either dedicated to the service or operate in licensed spectrum, which is specifically allocated for this purpose. In comparison, IPTV faces the challenge of having to deliver its traffic over the same connection as home Internet traffic [3]. While steps can be taken to improve IPTV Quality of Service (QoS), IPTV is vulnerable to issues such as dynamic traffic loads and equipment failures which can deteriorate the quality of the content delivered [7]. Depending on the type of encoding process and delivery parametrization, data loss can have considerable impact on viewing quality [8]. IPTV refers to the transport of any video signal and is not limited to broadcast TV. Other common IPTV services are video on demand, pay per view events, premium channels and network personal video recorders. Over The Top (OTT) delivery provides an interesting QoS challenge. Thus there is an emphasis on multicast traffic, but unicast traffic must also be carried for some services such as network personal video recorders.

We propose a monitoring system that uses semantically enriched IPTV performance metrics to interpret events and propose a corrective action. This allows the NM to be cognizant of customer experience, and even more importantly, responsive to service failure [9, 10]. We describe the state-of-the-art in the fields of two components of our consolidated monitoring system: IPTV performance measurement and semantic uplift.

1.1 Related Work: IPTV Monitoring

To maximize customer viewing quality given the challenges outlined above, the monitoring platform should provide the NM with comprehensive coverage of all potential sources of fault. The NM's objective is to maximize both QoS and Quality of Experience (QoE).

QoS measures how well the network transports IPTV content from the video head-end to the customer's playback device [11, 12]. Once a mechanism for identifying IPTV traffic and evaluating QoS is in place, inference techniques are required by the NM to locate and identify faults. In this paper, we adopt a semantic uplift and a rules-based method to aid inference. One approach for event detection is to have a set of predefined thresholds in place which, when breached, triggers an event or rule. Guidelines for threshold selection are given in [11]. An approach based on RTP with RTCP feedback outlined by Begen, Perkins and Ott in [12], may allow service providers to rapidly identify and isolate problems. The purpose of QoS monitoring is to use network events to initiate event resolution procedures: the NM may choose to re-route, or re-configure the traffic rules at the device in question, modify the video stream, or apply some combination of higher-layer loss-recovery mechanisms and protocols [13].

The QoE measurement measures service delivery from the customer's perspective [14, 15]: it is influenced by factors spanning the service plane; most of which are not subject to frequent change [11]. An accurate QoE measurement is of interest to service providers as it indicates how IPTV performs compared to the customer's expectation of how it should perform [10]. The relationship between QoE and QoS is tightly coupled. However, depending on the level of monitoring detail that the NM requires, monitoring may involve measuring delivery performance using just network measurements. One such example is the Media Delivery Index (MDI) [16]. One of the metrics included in the MDI is Media Loss Rate (MLR), which measures the amount of content lost during service delivery; typical targets are of the order of 10^{-3} packets/second or lower [17].

The NM may also choose to monitor the quality of received video after transporting it across the network. This could involve the use of reduced-reference (some video content information required) or no-reference (no video content information required) metrics. When monitoring is performed at this level, the NM may have to select a subset of sites in which to collect this information, as collecting data from all sites may lead to volumes of the data that are too large. In this situation, data aggregation may be also performed.

For completeness we describe the state-of-the-art in IPTV monitoring approaches. Kang, Kim and Hong describe a method and system architecture for monitoring and analyzing multimedia service traffic in [18]. The authors extract information on dynamic sessions, such as the dynamically selected protocol and port numbers, and they use this information to determine if previously unknown traffic is multimedia traffic. In short, they acquire session level information, which enables them to overcome problems associated with only using the port numbers of UDP and TCP to identify the application of the traffic. MMdump [19] captures a packet by referencing port numbers–it misses fragmented packets even though they may be multimedia service packets [18]. MMdump is used to investigate the characteristics of multimedia service traffic over RTSP and H.232 (via a parsing module). It operates by parsing control messages to extract the dynamically assigned port numbers–the parsing module then dynamically changes a packet filter to allow packets associated with these ports to be captured. However, MMdump incurs a burden as it requires frequent compilation and changes of the packet filter. In more recent approaches, the aim is to provide light-weight video quality metrics, in order to avoid solutions that require detailed knowledge of video characteristics. For example, Tao, Apostolopoulos and Guerin, propose a loss distortion model that accounts for the impact of network losses on video quality as a function of application specific parameters (video codecs, loss recovery rate, etc.). They then contribute a light weight video quality monitoring solution that is suitable for large-scale deployments as it does not require parsing and decoding of the transmitted video bit streams [20].

1.2 Related Work: Semantic IPTV monitoring

Once IPTV and network metrics have been collected, interpreting their meaning in the context of a greater dynamic system is crucial for a successful problem resolution. Semantic networking was proposed by Noirie, Dotaro, Carofiglio, Dupas, Pecci, Popa and Post in [21] with the aim of allowing the network to acquire knowledge about traffic flows so that this information could be used for self-configuration and self-management. In comparison with traditional passive monitoring approaches, semantic monitoring aims to understand the meaning of a traffic flow given its context: the relationship between flows. However, semantic monitoring depends on the availability of semantic network descriptions (domain models) and semantic representations of the dynamic network behaviour. Several authors have described semantic network modelling approaches, but the semantic uplift framework in this paper addresses the creation and updating of semantic representations of dynamic network behaviour.

López de Vergara, Guerrero, Villagrá and Berrocal in [22], describe and summarize several ontology-driven network management and monitoring projects. They detail how semantic technologies are applied and explain their advantages and drawbacks. In this paper, they find that semantic technologies are explicit, formal, and share-able, which means that ontology-based modelling and reasoning can be composed with other semantic techniques to express formal network monitoring and management logic to improve current approaches.

Current network analysis tools restrict analysis to the (low) level of individual facts and provide limited constructs to aid users in *bridging the semantic gap*–effective analysis of raw data from networked systems requires bridging the semantic gap between the data and the user's high-level understanding of the system. In a novel semantic framework (described in [23]), the raw network data represents facts about the system's state, and analysis involves identifying a set of semantically relevant behaviours, which represent "interesting" relationships between these facts. The objective of this framework is to enable semantic analysis at a level closer to the user's understanding of the system or process. The key to this framework is to provide: 1) a formal language for modelling high-level assertions over networked systems data as behaviour models; and 2) an analysis engine for extracting instances of user-specified behaviour models from raw data. This framework emphasizes reuse, composibility and extensibility of abstractions by using semantic techniques. Another approach [24] constructs a task ontology framework for diagnosing an IPTV network error with a generic vocabulary, which populates a representation of the domain knowledge and enables a knowledge-driven analysis procedure. Due to the growing complexity of IPTV networks, further work–such as the semantic uplift process presented in this paper–is needed to enrich the semantic meaningful information obtained from heterogeneous data sources, by leveraging domain expertise.

More generally, Hoag and Hayes-Roth, in [25], present an approach that applies semantic reasoning techniques to network management and resource allocation in order to avoid overbuilding. An ontology-based formal definition of different management behaviour specifications (integrated with management information definitions) in which Semantic Web Rule Language (SWRL) rules are defined directly over the ontology elements to allow for logical reasoning, are presented in [26]. These cases demonstrate the advantages of using semantic-based approaches for dealing with heterogeneous data sources, a necessary requirement for modern monitoring systems.

For IPTV networks, Lee and Kim in [24] contribute organized and conceptualized task ontologies for network quality diagnosis. Fallon, Huang and OSullivan in [27] present a knowledge-driven approach that applies cluster analysis on reported quality metric values. The authors map the analysis result to various domain ontologies for the service in order to analyze and optimize multimedia services in telecommunication networks. Determining how to adapt a semantic approach to the highly dynamic and distributed data sources of an IPTV network is challenging.

The knowledge plane, which contains a set of domain knowledge models, is the key component in semantic networking. A typical semantic networking approach may dynamically evoke internal and external knowledge models, an approach which remedies the shortcomings of some traditional policy-based approaches. Prior to 2009, a lot of work was published in the network management community on utilizing ontological approaches to express network management models [22]. The focus of this work was primarily on techniques for specification translation of traditional information models or the application of ontology-based approaches to information model inter-operability issues. Since then, significant developments have been made in defining linked data representations of information models [28] and evolving Directory Enabled Networks-new generation (DEN-ng) so that it "combines information models with ontologies" [29]. Both of these approaches are complementary. More recently, Seo, Kwon, Kang and Hong in [30] have proposed an IPTV performance indicator hierarchy that extends the DEN-ng information model along with an architecture that uses an ontology and Semantic Web Rule Language to manage Service Level Agreements (SLA), and to detect SLA violations in particular. We direct the interested reader to this paper which provides an excellent overview.

Frutos, Kotsiopoulos, Vaquero Gonzalez and Rodero Merino in [31] model QoS semantically to enhance the service selection process by annotating SLA templates with semantic QoS metrics. Moreover, the measurements and knowledge provided by different network monitoring approaches and platforms may be modelled and integrated with a syntactic ontology-based solution [32]. To capture domain knowledge, SARA [33] is designed to gather heterogeneous data from different resources and to organize data according to high-level or abstract semantic attributes through rules specified by domain experts. These semantic attributes support non-expert users exploring an information domain across heterogeneous sources.

1.3 Contributions and Organization

In Section 2, we give a problem specification for the IPTV monitoring problem. In Section 3, we describe what IPTV metrics can be collected and contribute a rule set for each metric. A hierarchy of rules facilitates the problem-inference process. In Section 4, we develop an exemplar semantic uplift engine and show how it is integrated with domain expert knowledge to consume and perform inference on raw log data. This system combines IPTV performance monitoring with a semantic uplift engine and visualization widgets. In Section 5, we demonstrate how each of these components interacts. We describe how this system diagnoses single and multiple points of failure for the representative problem scenarios introduced in Section 2.

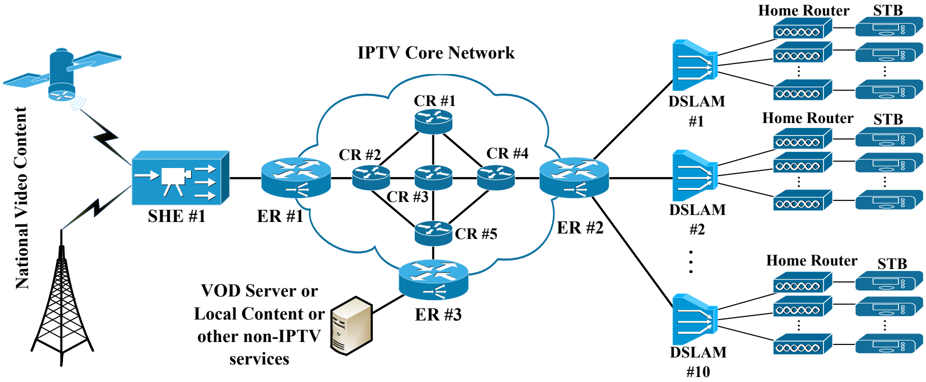


Fig. 1 Exemplar IPTV delivery topology consisting of a Super-Head-End, servers, network elements and the user's Set-Top Box. Subscriber IPTV flows traverse the core network and edge routers and are delivered to Set-Top-Boxes (STB)s through the DSLAMs and home routers.

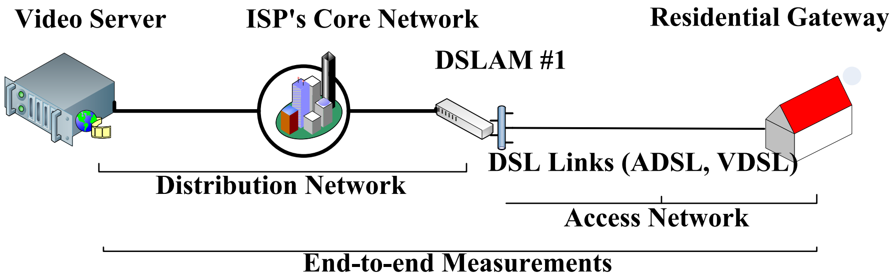


Fig. 2 Categorizing IPTV problem regions: Various interfaces and network element monitoring agents in Fig. 1 are categorized as belonging to the access network or distribution network. Problem localization diagnosis may require End-to-End measurements.

2 IPTV Monitoring Evaluation Test-bed

We start by specializing the unconstrained monitoring topology in Fig. 1 to the reduced set of elements in Fig. 2. Ideally, we perform monitoring at various interfaces illustrated in Fig. 1, servers and network elements for user impacting impairments from a Super-Head End to a Set-Top-Box. A simplified IPTV network can be viewed as having two distinct parts: the Distribution Network (DN) and the Access Network (AN). The DN carries the IPTV traffic from the Video Server (VServer) to the Digital Subscriber Line Access Multiplexer (DSLAM) via the ISP's Core Network (CN). The AN (DSLAM to Residential Gateway (RG)) aggregates the traffic from multiple DSL connections for transmission on the ISP's CN; the AN is also responsible for distributing traffic to each customer's DSL connection from the CN. This simplified architecture is depicted in Fig. 2. To collect metrics, monitoring agents are implemented in each of these network elements: RG, DSLAM and VServer. Once monitoring data is collected, it is written to a monitoring CSV file. This simplified IPTV topology is used in the remaining sections of this paper to describe how a subset of the challenges above are addressed. In Section 5.2, we describe how various components in the test topology are emulated using the network simulator NS-3.

Now that our test topology is defined, we use it to determine what types of outages we can incur and then detect in the network. A discussion of the types of metrics we can collect and problems we can detect (based on this specialized topology) is given in the next section. All that remains is our formal problem statement. Given heterogeneous sources of IPTV performance metrics, how can we monitor the network in such a way as to allow the NM drill-down, investigate outages and anomalies in semantically enriched detail and then suggest corrective action?

3 IPTV Per User and Network Metrics and Rules

Managed devices in IPTV networks exchange device specific metrics with the network management system. We describe a simplified agent-based IPTV deployment scenario that describes which nodes require monitoring agents, what metrics can be collected from these agents, and what IPTV performance evaluation rules can be inferred from them. Tables 1, 2 and 3 describe the metrics that are collected from the RG, the DSLAM and the video server. The metric-set collected from each network element was defined by choosing the set of metrics that was most widely available across all sets of network elements to give a service level indication.

3.1 Using Metrics to Inform Rules

The role of each major component within the delivery network architecture (VServer, DSLAM and GW) is now defined. Based on these component definitions, metrics are categorized as End-to-End, AN, or DN metrics. A set of rules is defined for each category of metrics (this set is by no means comprehensive). These rules can be used to ascertain the health of the IPTV service. The rules can be extended to provide a root-cause analysis capability.

3.1.1 Definition –Residential Gateway

The RG is responsible for the distribution of all traffic within the home. It is also responsible for forwarding the customer's traffic to and from the Internet Service Provider's network via the access network (e.g., DSL or Hybrid Fiber Coax). RG are generally equipped with a single WiFi interface and multiple Ethernet ports. The GW metrics comprise identification information, information about the content and general quality-related information. This information could be extracted from the node itself or recorded through packet analysis. This may require an extension to some of the GW functionalities. If a GW has the ability to filter out IPTV flows for independent monitoring of its traffic, it allows for a much more accurate monitoring of the IPTV service.

3.1.2 Metric & Rules –Residential Gateway

GW Uptime: This metric relates to the router itself. Regardless of the conditions of the network interfaces, downtime on a GW will terminate service delivery.

```
Rule: If GW.UPTIME < Monitoring Interval, GW has rebooted, trigger alarm
```

Table 1 Gateway Metrics

Name	Description
GATEWAY	This identifies the CSV as belonging to a gateway node
UniqueID	A unique ID for the GW (gateway-x, $0 \geq x \leq 100$)
Codec	This is a string indicating the codec in use (e.g H264 or MPEG2)
Bit-rate	Fixed value of 1.5 corresponding to 1.5Mb/s SD video
Uptime	Records the gateway uptime (OS resource value)
IPTVPLR	Records the PLR (iperf stream value)
Latency	End-to-end latency between the GW and VServer
Jitter	Records jitter (iperf value)
iptvMOS	Mean Opinion Score ($1 \geq MOS \leq 5$). This is weighted to have an average of 4.75

Table 2 DSLAM Metrics

Name	Description
DSLAM	This identifies the CSV as belonging to a DSLAM node
UniqueID	A unique ID for a DSLAM (random variable on 64 bits in ns3)
port_id	An integer value of the port number used for CSV reporting (typically 48 for current DSLAMs)
line_status	A string value representing the current line status (Up/Down/Test, Up = normal, Down = broken, Test = under repair)
average_up_line_rate	A value in Mb/s representing the rate of data flow from DSLAM to GW
average_down_line_rate	A value in Mb/s representing the rate of data flow from home GW to DSLAM
port_severely _errored_seconds	A value in seconds representing the total amount of time the port has spent experiencing transmission errors (when t = 2 an alert is triggered)
port_unavailable _seconds	A value in seconds representing the total amount of time the port was unavailable (line_status = Down/Test). When t = 3 an alert is triggered
port_high_ber	A value in seconds representing the total amount of time the port was affected by high bit error rate (when t = 15 an alert is triggered)
line_noise_margin	A float representing the noise margin on the DSL line (triggered if $<$ 10 dB)
line_resyncs	A value representing the current number of DSL resyncs performed by the line in the last monitoring interval (when resync = 2 an alert is triggered)

Table 3 Video Server Metrics

Name	Description
VServer	This identifies the CSV as belonging to a VServer node
UniqueID	A unique ID for the VServer (server-x,$0 \leq x \geq 100$)
PLR	The Packet Loss Rate for the VServer's outgoing interface
Latency	Records end-to-end latency between the VServer and GW
AccSuccRate	Records the access success rate for the VServer (mean of 98%)
AvStrmSetup	Average stream setup time, (mean = 150ms)(2PING $\mapsto$ GW + Processing Time)
CurResUse	Current resource usage (OS statistics)

PLR (Packet Loss Ratio) per Interface: This metric is the primary indicator of IPTV service quality. Any loss of video data will have an impact on the customer's QoE. Loss events should be kept to a minimum in order to ensure maximum QoE. Any loss events should be noted and reported.

```
Rule: If GW.INTERFACE.PLR > 0, trigger warning
```

Latency: This metric records the latency between the GW and the VServer. Latency is very important in the broadcast IPTV scenario; it defines the channel switching delay when a customer selects an new channel.

```
Rule: If AVERAGE (GW.LATENCY) > LATENCY.THRESHOLD, trigger warning.
TR-126 defines this threshold as 200ms.
```

Jitter: The inter-arrival times of packets sent from the VServer to the GW must be kept relatively fixed in order to ensure smooth playback. A buffer reduces jitter; however, this buffer affects the response time of the server.

```
Rule: If AVERAGE(GW.JITTER) > JITTER.THRESHOLD, trigger warning.
TR-126 defines this threshold as 50ms.
```

Video Mean Opinion Score (MOS): MOS ascertains the quality of the received video that is presented to the customer. Measurement of the video MOS is not always feasible due to the associated monitoring complexity; yet, if the video MOS scores are available they provide the ISP with a very accurate indication of IPTV service quality. A large number of different metrics are available for selection, but for an operational deployment either non-reference or reduced-reference metrics are typically used. The corresponding scores for each of these metrics can then be converted to a MOS. If direct access to the Set-Top Box (STB) is not available, the STB may calculate and forward the MOS to the GW; in this paper we assume that this is the case.

```
Rule: If AVERAGE(GW.MOS) < MOS.THRESHOLD, trigger warning.
Thresholds may account for subscriber/content type.
```

3.1.3 Definition –Digital Subscriber Line Access Multiplexer

The Digital Subscriber Line Access Multiplexer aggregates individual DSL links onto the ISPs back-haul network. In addition, it forwards traffic from the ISPs back-haul network to the appropriate DSL link to the GW in the customer's home. There are a large number of factors which can affect the delivery of traffic to/from the customers GW, such as line attenuation or excessive traffic demands. All parameters must be monitored in order to ensure that the DSL connection between the customer and the ISPs back-haul network is capable of supporting IPTV delivery with a high-level of QoE.

3.1.4 Metrics & Rules –Digital Subscriber Line Access Multiplexer

Note the term *port* is used here to identify a single DSL connection to a customer's residential gateway. The metrics, their names and values are based on a review of DSLAM hardware documentation.

Port Status: A value used to represent the current status of the port (Up = ready to transmit, Down = unable to transmit and Testing = testing mode and is unavailable to transmit).

```
Rule: If DSLAM.PORT.PORTSTATUS == DOWN|TESTING, trigger alert.
```

Line Status: A value used to represent the current status of the connection with the GW. All values are enumerated: Down: No connection to the GW; Downloading: Sending updated firmware to the GW; Data: Connection established, passing data; Test: In test state; Unknown: Connection with the GW failed due to an unknown error.

```
Rule: If DSLAM.PORT.LINESTATUS == DOWN | DOWNLOADING | TEST | UNKNOWN, trigger alert.
```

Line Uptime: A value to record how long the connection with the GW has been up.

```
Rule: If DSLAM.PORT.LINEUPTIME < Monitoring Interval, trigger alert.
```

DSL Max Attainable Up Line Rate: A value to represent the maximum attainable upstream line rate on a port.

```
Rule: None.
```

DSL Max Attainable Down Line Rate: A value to represent the maximum attainable downstream line rate on a port.

`Rule: None.`

DSL Up Line Rate: A value to represent the current upstream line rate on a port.

`Rule: If DSLAM.PORT.DSLUPLINERATE < DSLAM.PORT.DSLMAXATTAINABLEUPLINERATE, trigger alarm.`

DSL Down Line Rate: A value to represent the current downstream line rate on a port.

`Rule: If DSLAM.PORT.DSLUPLINERATE < DSLAM.PORT.DSLMAXATTAINABLEUPLINERATE, trigger alarm.`

Port In/Out Errors: Values to represent the current number of in or out transmission errors.

`Rule: If DSLAM.PORT.IN(OUT)ERRORS > Threshold, trigger alarm.`

Port Severely Errored Seconds (SES): A value representing the number of seconds experiencing severe errors on a port.

`Rule: If DSLAM.PORT.SES > Threshold, trigger alarm.`

Port Unavailable Seconds (UAS): A value representing the amount of time in seconds that a ADSL line is unavailable for a port.

`Rule: If DSLAM.PORT.SES > Threshold, trigger alarm.`

Port Loss of Signal Seconds (LOS): A value to represent the amount of time in seconds when a loss of signal has occurred.

`Rule: If DSLAM.PORT.LOS > Threshold, trigger alarm`

Seconds declared as a high bit error rate: A value to represent the amount of time in seconds that have had a high Bit Error Rate (BER) for a port.

`Rule: If DSLAM.PORT.HIGHBER > Threshold, trigger alarm`

3.1.5 Definition –Video on Demand Server

The video server (VServer) is responsible for the preparation of source content for transmission across the DN and AN. Content can be broadcast content, which is multicast across the network, or video-on-demand content, which is been selected for viewing by a particular user. Content is prepared for transmission using the following steps: selection of encoding details (codec, bit-rate, framerate, GOP size selection), the actual encoding process, packetization and multiplexing. Broadcast video content uses the MPEG Transport Stream (MPEG TS) as it allows for multiple channels to be multiplexed and delivered together. After packetization into the the MPEG TS, lower layer headers such as those for Real Time Protocol (RTP) and Internet Protocol (IP) are added. If Constant Bit Rate (CBR) traffic is required, the NM can chose to add null data to the MPEG TS to increase the bandwidth of the stream to a required value. In the case of on-demand content, the Transmission Control Protocol (TCP) is widely used to ensure robust delivery. This would not be possible in the broadcast case due to multicast being employed.

3.1.6 Metrics & Rules –Broadcast or VoD Server

Video Server Packet Loss: Represents the current loss rate on the video server’s outgoing link(s) to the DN. A value greater than zero indicates a severe problem with either the server or its link to the DN; such a scenario must be remedied immediately. Losses further down the path to the customer may be tolerated to some extent, but losses/errors at the server (especially Broadcast TV) will affect a large number of users.

Table 4 End-to-End Metrics: Acceptable packet loss rates are taken from [11]

Video Encoding	Limit			
SD MPEG-2	3.0 Mb/s 5.85e-6	3.75Mb/s 5.46e-6	5.0Mb/s 5.26e-6	-
SD H.264 AVC/VC-1	1.75Mb/s 6.68e-6	2.0Mb/s 7.31e-6	2.5Mb/s -5.85e-6	3.0Mb/s 5.85e-6
HD MPEG-2	15Mb/s 1.17e-6	17Mb/s - 1.16e-6	18.1Mb/s 1.17e-6	-
HD H.264 AVC/VC-1	8Mb/s 1.28e-6	10Mb/s -1.24e-6	12Mb/s 1.22e-6	-

```
Rule: If VSERVER.LINK.PLR > 0, trigger alarm.
```

Video Server Latency: A value to represent the latency between a VServer and a connected customer's STB. Excessive latency will decrease the QoE due to excessive wait times for channel change or on-demand transactions.

```
Rule: If VSERVER.LINK.LATENCY > threshold, trigger warning.
```

Video Access Success Rate: A value to represent the current video access success rate, i.e. what percentage of requests to access a particular video or channel lead to successful transmission of the video/channel.

```
Rule: If VSERVER.ACCESSSUCCESSRATE < threshold, trigger warning.
```

Average Stream Setup Time: A value to represent the average time taken to setup an on-demand (or broadcast group join). This is calculated in the present paper as the time taken from the initial setup request to the time taken for the first packets to be transmitted to the customer.

```
Rule: If VSERVER.AVERAGESETUPTIME > threshold, trigger warning.
```

Video Server Resource Utilization Rate (%): Represents the current resource utilization rate expressed as a percentage of available resources. Resources can be individually measured in terms of the CPU, memory, or disk access. We measure the video server resource utilization rate as a combination of all three parameters.

```
Rule: If VSERVER.RESOURCEUTILISATIONRATE> threshold, trigger warning. The value for
'threshold' requires some understanding of the number of sessions that can be concurrently run.
```

3.1.7 End-to-End Metrics & Rules

Packet Loss Rate PLR (%): Acceptable packet loss rates –according to TR126– are listed in Table 4. Values can either account for all traffic or *ideally* for IPTV traffic only.

3.2 Hierarchical Rules to Interpret Metrics

The rules used for each resource type can be combined hierarchically to build more complex rules for the ANs, DNs and the End-to-End networks. In a hierarchy of rules, higher level rules can be applied to achieve problem isolation (e.g., to a single DSLAM). However, these rules only apply to the cases where there are multiple customers facing problems with the service quality. If only one user is experiencing an issue, the metrics collected from the appropriate resource types identify the source of the problem. We give some examples of higher level rules; these rules are expressed less formally than the ones seen previously. Fault resolution is typically performed edge –the closest point to the customer where a fault report has been triggered– inwards. Hierarchical Rules (HR) serve to isolate the fault by identifying a fault from both the customer-side and the Vserver-side.

HR1: A number of customers (each served by the same DSLAM) are receiving poor QoS (either through loss, excessive latency, higher jitter, low video access rates), but the VServer(s) they are receiving content from is/are reporting no issues. We conclude that we can identify the DSLAM as being the source of the service degradation.

HR2: A number of customers (served by a collection of different DSLAMs, but from the same VServer) are receiving poor QoS. The individual DSLAMs are reporting no issues; we can therefore identify the common VServer as being the source of the degradation.

HR3: A collection of customers (served by a collection of different DSLAMs and VServer) are receiving poor QoS. If the DSLAMs and servers are reporting no issues, we identify the individual GWs as being the source of degradation by further investigation of their metrics.

HR4: A collection of customers (served by a collection of different DSLAMs and VServer) are receiving poor QoS. If the DSLAMs and servers are reporting no issues and further investigation of their individual gateways yields no indications of problems, we are unable to isolate the source of the issue.

HR5: A collection of customers (served by the either the same or different DSLAM, but the same VServer) are receiving poor QoS. If both the individual GWs and DSLAMs are reporting no issues and the VServer's PLR on its link to the distribution network is below the threshold, we must then investigate the VServer's resource utilization rate. If this yields no insight, investigation must be made into the quality of the encoded video stored on the VServer.

4 Semantic Uplift Engine

The semantic uplift engine, a plug-in to our IPTV monitoring visualization framework is now described. This component leverages modelled domain expert knowledge for real-time uplift of heterogeneous raw log data sources of IPTV service monitoring information and knowledge. The engine is part of the monitoring visualization framework for the IPTV delivery network monitoring agents. This framework takes raw node metric logs (CSV files) from heterogeneous components of the IPTV network, annotates them with domain concepts represented in W3C's standard Resource Description Framework (RDF), and aggregates them with modelled domain expert knowledge for anomaly diagnosis and analysis. This uplifted information is displayed using a variety of visual widgets within the monitoring framework.

Fig. 3 illustrates how the semantic uplift engine fits into the IPTV delivery network topology. The semantic uplift engine consumes node metric CSV files generated by the monitoring agents at each point of interest in the network described in Section 3. Note that deployment of the uplift engine depends on the availability of a distributed collection infrastructure, such as that provided by the IBM Tivoli performance management suite [34].

Knowledge models are at the core of a semantic approach to monitoring; hence, we discuss the application of our domain expert knowledge model to the IPTV delivery network. An outline of the semantic monitoring visualization framework is also provided. This focuses on the details of the semantic uplift process including data mapping, event detection, event aggregation and anomaly detection and analysis.

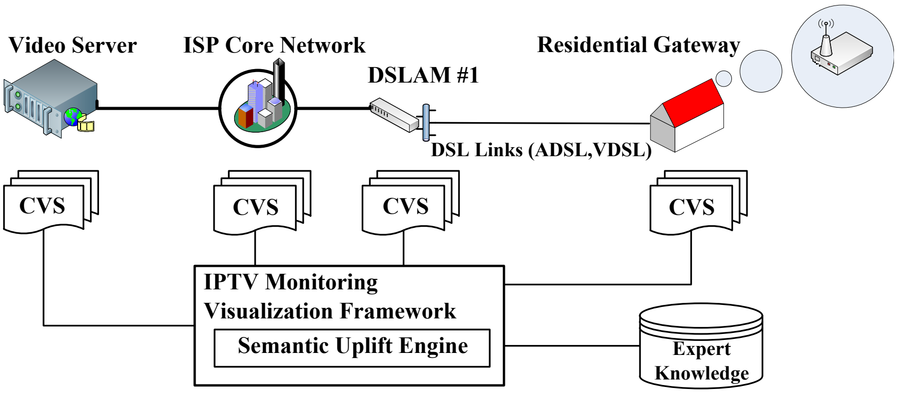


Fig. 3 Mapping of the Semantic Uplift Engine to the IPTV Delivery Network in Fig. 2: CSV files generated by network element agents are consumed by the semantic uplift engine leveraging expert knowledge.

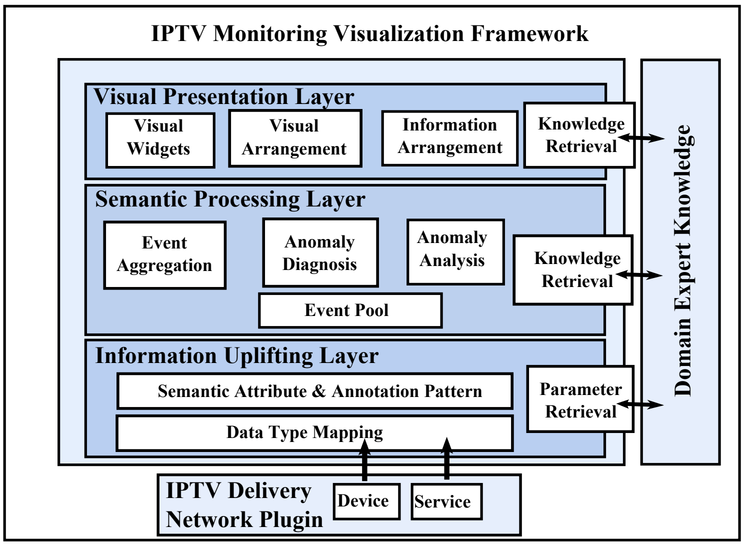


Fig. 4 IPTV Monitoring Visualization Framework: Network event data is first enriched via semantic attribute and pattern annotation, events are processed using event aggregation, anomaly diagnosis and analysis, and finally, presented using visual widgets.

4.1 IPTV Monitoring Visualization Framework

The framework in Fig. 4 has been developed as a general-purpose tool for consuming both logs/events and knowledge provided by a domain expert to produce both visualizations and user-centric explanations of network conditions. It applies semantic techniques to knowledge representation and event processing. The framework is divided into three processing layers: the information uplifting layer (network event data conversion and enrichment), the semantic processing layer (event analysis), and the visual representation layer (presentation). The information uplift layer supports diverse annotation patterns and processes.

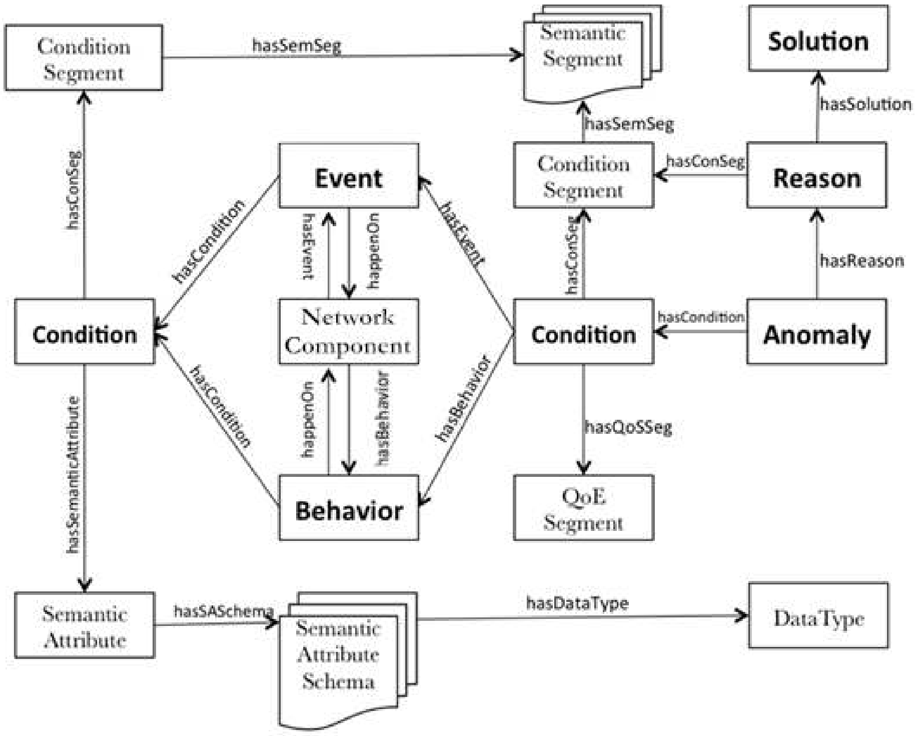


Fig. 5 Domain Expert Upper Knowledge Model

4.2 Domain Expert Knowledge Model

The insights and knowledge of domain experts are captured and modelled in the domain expert upper knowledge model in Fig. 5. It acts as a bridge between expert insights, logs, analysis and network visualizations supporting monitoring. Rather than being a single overarching IPTV knowledge model, the domain expert knowledge model is an upper or meta-model that enables the easy integration of multiple domain specific models; for example, for individual devices or services. Crucially, it defines a framework for linking human expert insights about these models or systems and system artifacts such as device logs or events. It also allows experts to encode knowledge about system states, behaviors and potential network or service anomalies. This is significant because these upper model concepts can then be used to span multiple device or network models enabling cross-model reasoning. In the meta-model, the semantic attribute and semantic segment are the two key concepts used to enable efficient processing and combination of domain expert insights based on heterogeneous network component models.

Semantic Attributes are used to annotate raw log or event data from the network. They support heterogeneous data collection because a domain expert can define multiple sources of evidence of network conditions as equivalent (for example, evidence of low effective bandwidth via events from multiple services). Once annotated, these events can all be treated equivalently by the ontology-level semantic reasoning, and therefore, the gap between raw log data and the formal domain expert knowledge model is bridged. Semantic attributes are encoded in Resource Description Framework knowledge models. They encapsulate an expert's subjective insight into the IPTV network. They consist of a concept definition, a set of constraints and links to both the raw log data or metrics and a specialized knowledge model for the device, network, or service. For example, the semantic attribute "antenna_noise_bad" could be defined as occurring when a "high" WiFi antenna noise is recorded in a specific type of access point log file, where "high" is defined by the expert-specified constraint "more than -80dBM". It also links to the "antenna_noise" concept in a detailed wireless device metrics knowledge model.

Semantic Segments are used in the meta-model to represent a combination of semantic attributes, domain ontology classes and the corresponding logic to capture network state transitions, anomaly detection or resolution. This logic goes beyond the typical use of structured knowledge (ontologies) by enabling generic rules or temporal logic to be combined with traditional semantic technologies. This provides a highly abstracted description of logical rules and conditions for semantic entities, which are, for example, automatically decomposed into atomic SWRL rules and SPARQL queries in the semantic processing layer of the monitoring framework.

In addition, the domain expert knowledge model provides OWL classes to support problem identification, diagnosis and analysis. They are: *Condition, Semantic Entity (Event, Behavior, and Anomaly), Reason,* and *Solution*. They represent conditions that could be a trigger for another event, behaviour, or anomaly. The *Event* class is used to describe the network performance status and sudden changes in state. The *Behavior* class indicates the behavior that happened on/between network components, like "data transferring between a *router* and *gateway*". The *Anomaly* class is used to represent events or behaviors that affect the Quality of Experience (QoE) for users. The *Reason* class is used to relate expert-defined reasons to an anomaly of a given type. The *Solution* class is used to describe expert-defined solutions for combinations of reasons and anomalies. The problem identification, diagnosis and analysis classes are always associated with either a single or a combination of several Semantic Attributes (and via these attributes to raw log or event data). We now discuss the data type mapping, information uplift, semantic attribute annotation, semantic entity annotation and semantic processing steps in more detail.

4.3 Data Type Mapping

Fig. 6 illustrates how domain experts define the meaning of elements in system log data. The Data Type Mapping process maps an entity in the domain expert knowledge model in Fig. 5 to elements in the log data or metrics. The run-time mapping process is instantiated by a number of mapping schemes that are encoded by the domain expert. The outputs of the run-time mapping process are a set of resource models, which refer to domain knowledge models in order to make them understandable by the information uplift engine. These semantic attributes inform the enhancement of low-level data with semantics, where each contains the following: a semantically meaningful concept; patterns related to the data type; parameters related to the pattern; links to the domain knowledge model; and links to the raw data.

For example in an IPTV delivery network, one customer consumes IPTV via his home gateway "gateway66", that is connected to a DSLAM with id "dslam43" that receives an IPTV stream from a video server called "vserver". We model the "gateway66", "dslam43" and "vserver" as instances of the classes Gateway, DSLAM and VSERVER respectively. The semantic uplift engine consumes the real-time metrics of these instances and annotates them with domain semantic attributes in order to identify if there is an anomaly and

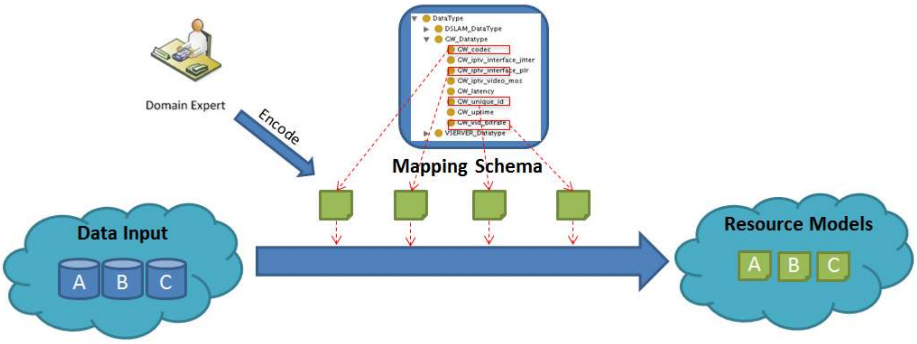


Fig. 6 Domain Expert-based Data Type Mapping

if it affects IPTV. In this case, for "gateway66", the semantic uplift engine retrieves the corresponding parameters and schemas related to the Gateway class and assigns semantic attributes to the real-time data. If the latency of "gateway66" is higher than 200ms, it triggers the threshold and the semantic attribute "latency high" is assigned to the data. The event "IPTV QoS Low" is triggered by the semantic attribute "latency high" on "gateway66".

4.4 Information Uplift

The information uplift approach extracts information by annotating semantic meanings onto the captured characteristics of identified network stream log data and models the extracted information in an appropriate representation, that references the domain expert knowledge model. As shown in Fig. 7, the information uplifting approach is divided into two processes: the semantic attribute annotation process and the semantic entity annotation process.

4.4.1 Semantic Attribute Annotation

The semantic attribute annotation process aims to extract meaningful information from snapshots of a real-time data stream. Although this real-time data is fed into the semantic attribute annotation process based on highly heterogeneous metrics, the data types (e.g., "packet_loss_rate") of metrics are aggregated and mapped to corresponding data type elements in the knowledge model. Hence related semantic attributes can be applied to the same data type to simplify the annotation process. This process supports diverse information extraction and annotation patterns for semantic attributes, which are pieces of semantic encodings captured from domain experts. When processing the real-time data streams, the pattern detection algorithms are applied to aggregate and detect data value changes that capture the characteristics of the data stream by dividing the data into discrete intervals of moderately varying behaviour or time-stamped change points where there are abrupt changes of the steady state metric values. The appropriate semantic attributes are associated with these characteristics in the raw log streams or metrics. Information extraction techniques are applied to capture the characteristics of the stream data. As an example, in Fig. 8, in a given time interval, heterogeneous log data from devices and services in the network is collected and aggregated. The pattern detection algorithm 'A', is applied to detect the changes of the steady state metric values of the real time data stream. This algorithm also divides the data

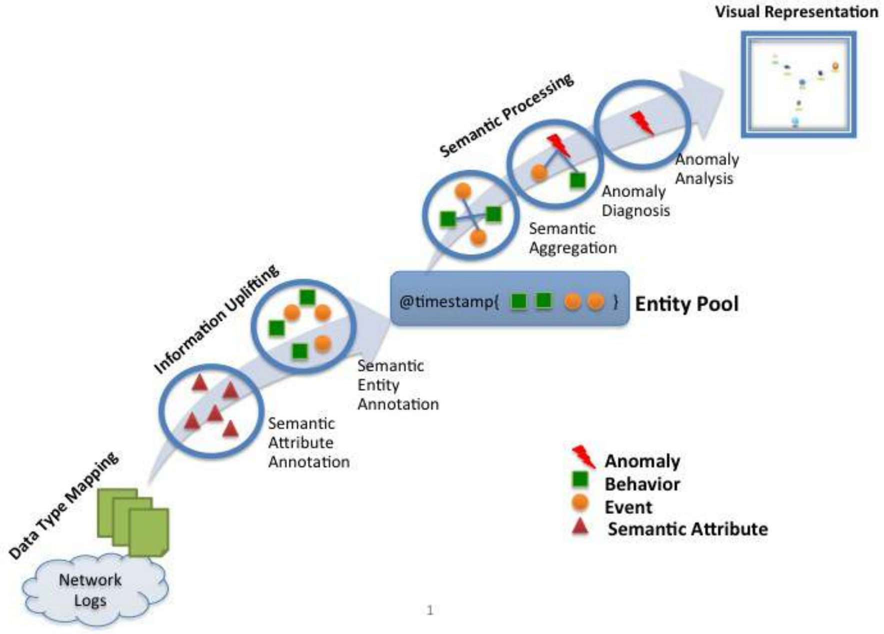


Fig. 7 The Phases of the Semantic Uplift Approach.

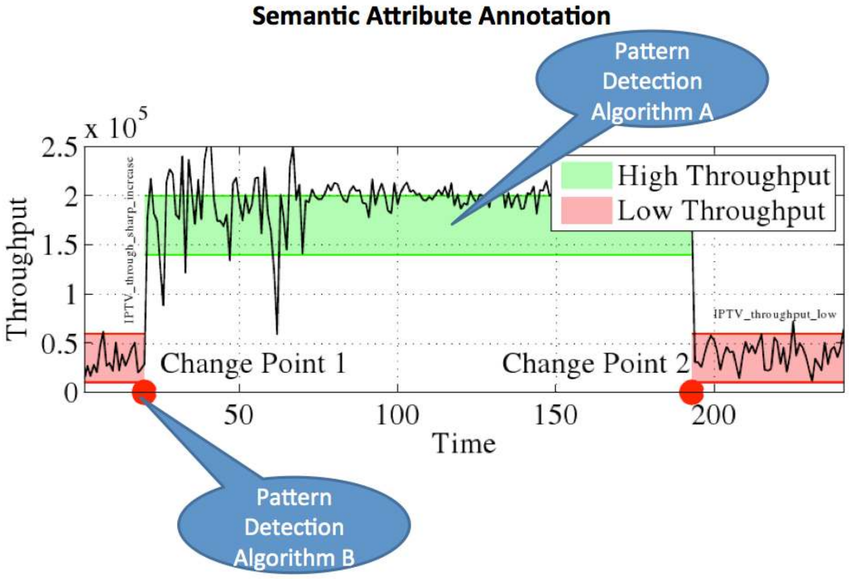


Fig. 8 The pattern detection algorithms applied on the real time data stream.

stream into discrete intervals and points. Another pattern detection algorithm 'B', aggregates the data value and captures the characteristics in each discrete interval. Then these captured characteristics are annotated with domain-expert defined semantic attributes like "throughput_high" or "throughput_low".

The information is extracted by annotating characteristics of the log data stream, and modelled as semantic attributes, corresponding to the expert-defined semantic attribute schema. According to the captured characteristics, there are several types of the real-time annotation process that can be used to generate the annotated semantic attribute stream (P) with corresponding time stamps:

- **Discrete Annotation:** This process detects change points in the data stream. These change points are considered to be a sequence of semantic meaning points (S) that are annotated forming a semantic attribute stream (P), i.e., $P = \{S_1, \ldots, S_m\}$, where $S_i = (s,t)$ is a pair with the semantic meaning (s) at time-stamp t.
- **Continuous Annotation:** This process is used to annotate a piece of data with its corresponding meaning. It annotates the data intervals with the data status (S), i.e., $P = \{S_1, \ldots, S_m\}$, where $S_i = (s, t_1, t_2)$ is a triple indicating the data has the status (s) in a period (t_1, t_2).

The annotated semantic attribute streams are maintained for the further extraction of meaningful information that enables the next annotation process, namely, semantic entity annotation.

4.4.2 The Semantic Entity Annotation Process

During the semantic entity annotation process, the semantic attributes describing log entries are linked to higher-level semantic entities like events and behaviours in the domain defined by domain experts. This enables a dynamic picture of the network to be built up from the annotated semantic attribute stream, allowing features such as the network topology status changes to be available in a more meaningful way for the visual representation to non-expert users.

Through these information extraction and annotation patterns, semantically meaningful information is extracted from the raw data. Based on the annotated semantic attributes, all related entities in the domain knowledge model are checked one-by-one in an event diagnosis loop, in which the information is iteratively annotated with events from low-level to high-level. This checking process is performed based on the rules encoded by the domain expert in the semantic entity schema. For example, a particular semantic attribute could be considered to be a low-level annotation. If there is another entity whose condition is based on this initial annotation, this can refer to higher-level events; in short, events are annotated in a level-by-level manner. All annotated events are kept in an entity pool. In the semantic entity annotation process, the entity pool constantly checks the semantic annotation loop until there are no more new events (and no rules to fire) and at that time, the uplift of the data in this time interval is finished. The semantic entities in the entity pool are then maintained for use in other approaches. There are several types of annotation processes for this pattern-driven annotation stream (P):

- **High-level Meaning Annotation:** This process aims to annotate the high-level event (S) onto the low-level semantic attribute stream. The high-level semantic meaning (s) with the corresponding low-level semantic meanings ($s_1, \ldots, s_n$) are determined according to the expert encoded semantic segments, i.e. $P = \{S_1, \ldots, S_m\}$, where $S_i = (s, \{s_1, \ldots s_n\})$.

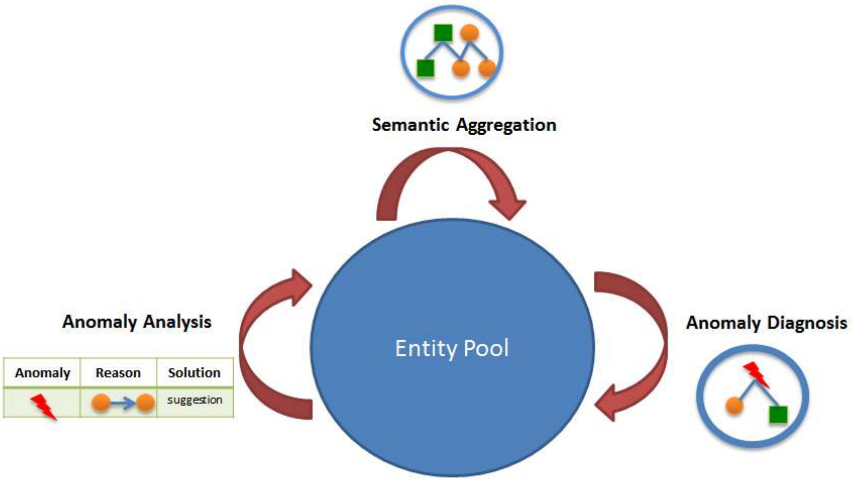


Fig. 9 The semantic processing approach

- **Behaviour Annotation:** This process annotates the behaviour (b) onto the raw data stream that is based on a semantic segment of behaviour events (S), i.e. $P = \{S_1, \ldots, S_m\}$, where $S_i = (b, t_1, t_2)$ is a triple with the behavior (b) that occurred in a period (t_1, t_2). For example, "Play" is a behavior for an IPTV service. When the IPTV service is playing, the semantic attribute "playing" is dynamically annotated onto the log data flow.

4.5 Semantic Processing

The semantic processing approach aims to apply further knowledge-driven aggregation, diagnosis and analysis to uplifted information either in response to user interactions with the visual widgets or for deeper semantic analysis (for example, to determine the root-cause of events or to support multi-level problem descriptions in an analytic view). All annotated semantic entities are maintained in an entity pool with a semantic structure, which means the entities are linked to each other according to the relationship extracted from the knowledge model and encodings. Fig. 9 shows an example of semantic entities relevant to a network resource model in the entity pool. The blue link indicates the semantic relationship between two linked entities. This linked structure enables the further semantic reasoning through these entities. Time stamps are associated with entities to enable temporal semantic reasoning during diagnosis and analysis.

The semantic processing approach enables drill-down analysis across the monitoring domain to support the higher-level monitoring objectives of non-expert users. Semantic entities uplifted and modelled from heterogeneous log data are linked to enable semantic aggregation, anomaly diagnosis and anomaly analysis if required. Thus, this semantic processing approach is executed in three steps:

4.5.1 Semantic Aggregation

In the information uplifting approach, the semantic entities are uplifted and modelled based on particular network resources. As further information uplift, the semantic aggregation process reviews all of the entities extracted from different domains currently in the entity pool to ensure that they include references to appropriate higher-level entities such as "IPTV_service_quality_low" that is associated with the semantic entities annotated on devices, connections, behaviours and target groups on the delivery path.

4.5.2 Anomaly Diagnosis

The anomaly diagnosis process detects and indicates if an anomaly has happened among the current uplifted semantic entities. Events that cause network health degradation or that affect the quality of user experience are labelled as an anomaly. The diagnosis process is a knowledge-driven process that builds an anomaly model based on the dependency of annotated semantic entities across different network monitoring domains by using the semantic reasoning.

4.5.3 Anomaly Analysis

The anomaly analysis process facilitates a drill-down analysis across the knowledge domains to determine the anomaly's root-cause reason. It models the analysis process step-by-step to support non-expert users' understanding of network problems.

- If an anomaly is chosen to be analyzed, its anomaly model is loaded into the semantic processing layer.
- All entities in the anomaly model are expanded from high-levels to low-levels according to each entity's dependency.
- This expansion is built as a decision tree that may refer to the entities from different knowledge domains.
- A recursive approach is taken to check the root-cause by following some predefined consequence (in one proof-of-concept consequence, the check process starts from the last "unusual" node and traces back to the source of the delivery path). The first root-cause with the least conflicts is considered to be the root-cause with the highest possibility.

If an anomaly is detected in the anomaly diagnosis process then a root-cause analysis process is applied to it. For example, an IPTV quality degradation anomaly is defined as being potentially caused by a root-reason "Core_Router_offline" status for the source device. The aggregation, diagnosis and analysis result is also semantically modelled to represent what is happening; what will happen; what caused the problem; and the available solutions. The results of the semantic aggregation, diagnosis and analysis are represented in a display-independent schema for consumption by the visual representation. Thus, a wide range of widgets can be developed to enable human-centric visual arrangements.

4.6 Handling multiple events

Handling multiple events that are generated at almost the same time is a basic requirement of any monitoring system. We take the following approach: CSV files are generated by the monitoring agents at the various points in the network indicated in Section 3. The semantic uplift engine consumes these CSV files in the manner outlined above. They are represented here as a continuous times series of CSV arrival times denoted by

$$x(t) = \begin{cases} 1 & \text{if a CSV file is received at time } t \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

During the uplift process and fault diagnosis, we consider a set of CSV files gathered during a time interval of length T seconds. Windowed segments of this time series are treated separately during run-time using a moving window function $\mathbf{I}_{[-\frac{T}{2},\frac{T}{2}]}(t)$. For example,

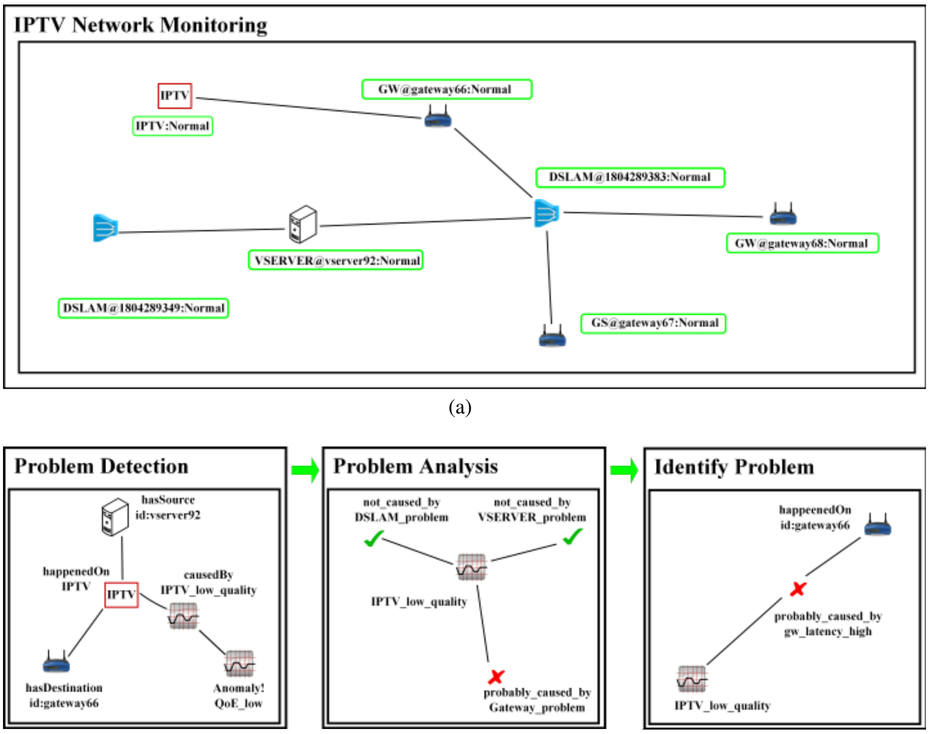
$$\hat{x}_n(t) = x(t) \times \mathbf{I}_{[-\frac{T}{2},\frac{T}{2}]}(t - nT) \tag{2}$$

The n^{th} signal segment that is positioned at time nT, that is $\hat{x}_n(t)$, has finite support. The number of CSV files consumed and used in the uplift and diagnosis process is bounded by the choice of the time interval, T. Windowing in this manner assumes that the network's statistics are locally stationary, so it follows that uplift and fault diagnosis performed on events that occur at approximately the same time are caused by outages that are potentially correlated. A large time interval choice, T, may cause unrelated events to be considered in our uplift and analysis; a small time interval T may mean that key events are not accounted for in our diagnosis. Multiple events, occurring during a short time interval can be treated as having a number of underlying factors. Event support/segmentation is captured by the indicator function, $\mathbf{I}_{[-\frac{T}{2},\frac{T}{2}]}(t) = 1$ when $|t| < (T/2)$ and 0 otherwise, and is important as it focuses fault diagnosis and problem isolation routines. Each signal segment, $\hat{x}_n(t)$ indicates the times of arrival of the events that occur in the neighbourhood of time $t = nT$, which is bounded above and below by $T/2$. The set of events associated with these arrival times, $\hat{x}_n(t)$, is denoted $\mathscr{X}_n$. The events occurring in the time interval centered on nT, e.g., $\mathscr{X}_n$, are semantically enriched from a low-to-high level, in the manner described above. The key point is that only events in the set $\mathscr{X}_n$ are annotated. Then the annotations are evaluated to see if the associated events could have affected the QoE. If the QoE is affected, we deem an anomaly has occurred, and we show it (and all other anomalies) in a real-time dashboard. This whole process is carried out within the time interval T. The process is repeated on the next time window $(n+1)T$ of events $\mathscr{X}_{n+1}$, with arrival times during the segment $\hat{x}_{n+1}(t)$.

Handling multiple events in this manner requires the choice of two parameters: 1) the time interval and 2) whether or not overlapping windowed segments are used. There is not one parametrization that is optimal for all IPTV deployments; however we give some heuristic methods to guide their selection. The time interval is a heuristic that may be set by using some multiple of the average length of an outage episode detected in the network. Secondly, overlapping windowed segments incur higher processing costs on the monitoring system; however, they yield superior resolution on the visualization dashboard. Choosing how much to overlap involves a trade-off between increased resolution and the computational capacity of the monitoring deployment.

4.7 Visual Representation Layer

In the Visual Representation Layer, several user-friendly widgets are built in Adobe Flex to reduce the level of expertise required to understand and monitor the network based on the aggregated, uplifted, and enriched log information retrieved from the semantic processing layer. It is important to note that the information retrieved is independent of any particular visualization widget, so the visualization layer can embed additional expertise-driven logic to select or personalize the most appropriate presentation widget for a given combination of information and user. This separation of domain-specific expertise from visualization specific expertise improves on the traditional approach of embedding domain reasoning, and associated domain-level assumptions in the presentation layer.



(a)

(b) Scenario 1: Anomaly analysis performed when there is excessively high latency at the GW which causes poor QoE.

Fig. 10 Topology used to illustrate problem detection, analysis and identification for the four test scenarios. Comprehensive analysis of monitored events on a simulated IPTV network is visualized by the NM using the Network Topology, Problem Analysis and Diagnosis, and Real-time Event Monitoring Widgets.

The network topology widget is depicted in Fig. 10(a). It illustrates the system in operation, monitoring the events on a simulated IPTV service delivery network. Fig. 10(a) displays the networks, nodes and services being monitored in the network, their relationships and the status of each node. A green node name label indicates normal operation, whereas a red label indicates a problem has been detected at that node. For example, if a Gateway changes label colour to red in Fig. 10(a), the problem analysis widget illustrated in Fig. 10(b) helps the NM infer the cause of the problem. The analysis in Fig. 10(b) draws upon the expert knowledge of the domain embedded in the semantic processing layer. Fig. 10(b) illustrates what aspect of the node has an error. It helps identify the likely root cause of the problem. The widget in Fig. 10(b) visualizes the inference path used to derive its conclusion. The objective of this form of display is to provide additional context for the user when troubleshooting network problems. Although it is not shown in this figure, it is also possible to associate suggested corrective actions with problems in the domain knowledge model. The widget in Fig. 10(b) is opened by double-clicking on the problem node in the network topology widget in Fig. 10(b). The real-time event monitoring widget is not illustrated here. It shows simple visual alerts (orange-brown spots) on a time axis at the instances when problems have been detected in the system. There are also a number of other widgets available for playback of raw log data, for visualization of service execution rates

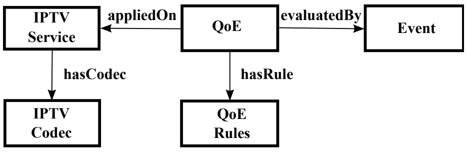


Fig. 11 Representation of QoE Rules in IPTV Domain Knowledge Model

and so on. In addition, widgets for collecting expert knowledge for the purpose of defining semantic attributes in the system exist. Fig.11 illustrates the representation of QoE Rules in IPTV Domain Knowledge Model.

5 Experiments & Evaluation

We evaluate the feasibility of the monitoring system using the four problem scenarios proposed in Section 5.1. Each scenario is defined in detail below. We describe the emulation environment used to create the test network and events.

5.1 Failure Scenarios

To evaluate the feasibility of the proposed monitoring system, we propose four problem scenarios. These scenarios demonstrate if the solution can deal with single and multiple points of failure in IPTV delivery. They address a subset of the challenges listed above. In particular, the NM is interested in determining whether or not the integrated solution delivers knowledge of an outage in a per user manner and allows the NM to assess the impact of an outage on a per user level.

Scenario 1: Excessively high latency at a Gateway is causing poor QoE.

Scenario 2: A high number of severely errored seconds at a DSLAM is causing poor QoE.

Scenario 3: A high latency and number of severely errored seconds cause poor QoE at a DSLAM and Gateway.

Scenario 4: The resource utilization rate breaches a threshold at the Video Server that causes poor QoE.

5.2 IPTV Emulation Model

Hardware emulation is used to approximate each network element's behaviour. Emulation is a well-established capability of many network simulators (NS-2, NS-3, Qualnet and OPNET); the wide-spread acceptance of NS-3's ability to accurately emulate the functionality of real network elements underpins the accuracy of these experimental results. Emulation is used to avoid the physical constraints and CAPEX associated with building a real test IPTV network.

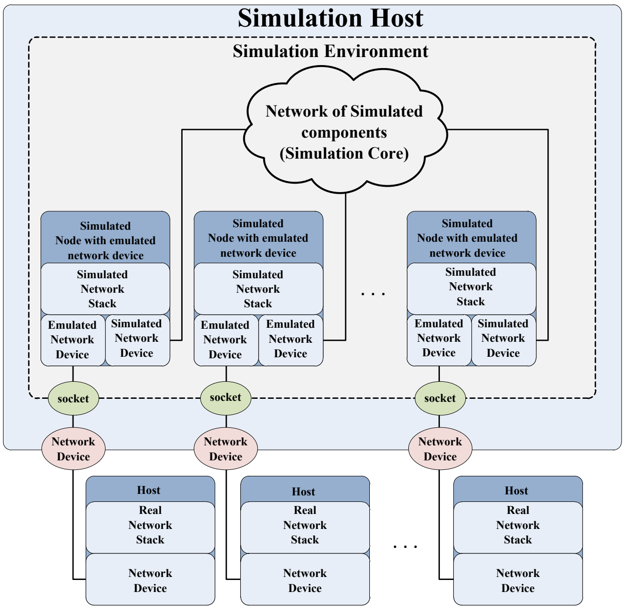
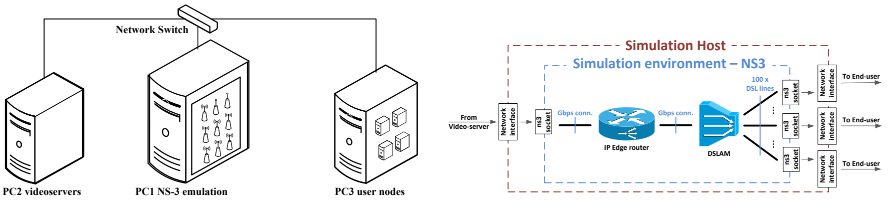


Fig. 12 Emulation Concept: the Simulation Core simulates the test topology which consists of Edge Routers and DSLAMs in this paper. Some nodes in the Simulation Environment are connected to the Simulation Core and to real network devices via sockets.



(a) IPTV Emulation Topology

(b) Network topology emulated in NS-3

Fig. 13 IPTV Emulation Topology: In Fig. 13(a), PC1 acts as the Simulation Host which runs NS-3 (More detail is given in Fig. 13(b)). PC2 is the video streaming server. PC3 plays the role of the end-users.

DSLAMs and IP Edge Routers are simulated, which allows us to mimic the types of problems identified in the different regions of the exemplar topology in Fig. 2. The combined emulation-simulation configuration is summarized in Fig. 12. The Simulation Host computer hosts the simulation and has real world connectivity through real network devices. In the Simulation Environment (NS-3) a Simulation Core simulates the desired topology of *simulated-only* nodes. The Simulation Core is comprised of components common across all protocols, hardware, and environmental models: the Simulation Core is used to build-up the entire simulation engine. This part of the system has no connection with the real components. On the other hand, some nodes in the Simulation Environment are connected to both the Simulation Core and to real network devices installed on the real Simulation Host. The binding to the real devices is made using *sockets*. Furthermore, the real devices are then connected to Real Hosts.

The IPTV simulation/emulation system is presented in Fig. 13(a). All components are interconnected using Ethernet cables using a single network switch. Each computer hosts a part of the IPTV system. PC1 is the Simulation Host running NS-3 as the Simulation Environment. PC2 plays the role of the video streaming server. PC3, a computer with multiport Gigabit Ethernet capability, plays the role of the hosting machine where end-users are visualized using Virtualbox. For simplicity we illustrate one DSLAM and Edge Router in Fig. 13(b).

The topology simulated by the NS-3 environment is illustrated in Fig. 13(b). Traffic pushed by the Video Server is forwarded through the ingress point of the Simulation Environment. From there, a simulated Gigabit per second (Gbps) line forwards the traffic to an IP Edge Router. The IP Edge Router serves a DSLAM through a Gbps connection.

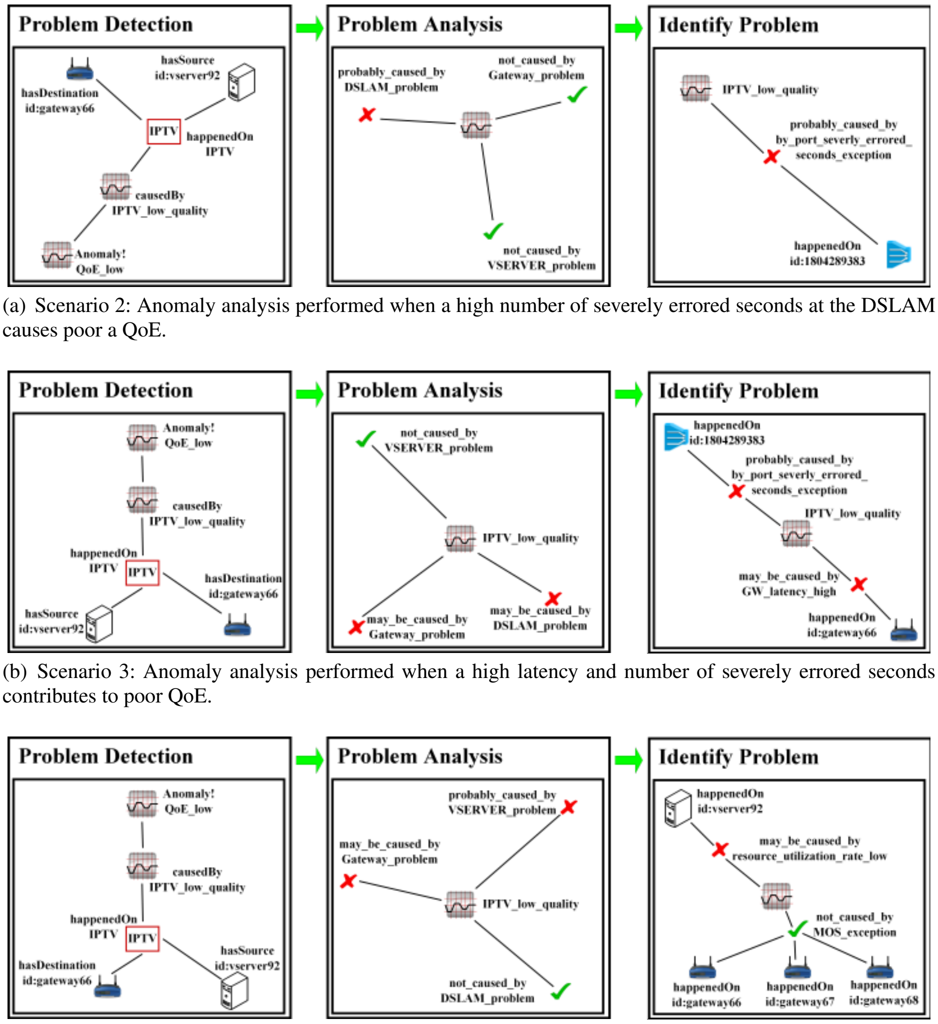
For simplicity, we have used only one DSLAM here, but the simulation can be scaled-up to include multiple DSLAMs. However, the size of the topology simulated in NS-3 is limited by the computational resource capacity of the machine hosting the simulation environment. In this work, the machine performed packet forwarding fast enough to keep up with the real-time flow of the packets. A computational resource analysis showing at which traffic load the NS-3 simulation is not able to forward traffic in real-time, is out of the scope of this paper. It is this constraint that limits the size of the simulation/emulation experimental set up. We have implemented our own DSLAM model in NS-3, which is able to collect and push monitoring reports to the central entity. Up to 100 DSL lines are served by the DSLAM. We have implemented an ADSL model with upload and download data rates of 10 megabits per second (Mbps) and 2 Mbps, respectively. The ADSL lines forward traffic to the end-users via the NS-3 egress points. In our simulation scenario we use 100 ADSL lines. To avoid having 100 Network Interfaces to connect the 100 end-users to the DSLAM, we point each egress socket to the same real Network Interface on the Simulation Host. As such, all ADSL connections are multiplexed over one single 1 Gbps wired Ethernet connection. The capacity of the simulated ADSL connections ensures that the multiplexed traffic of one link does not interfere with another link.

5.3 Context for Scenarios

In these experiments, an anomaly or problem has occurred when an IPTV subscriber is experiencing a low QoE, due to fluctuations in the IPTV flow in the network test-bed. The topology in Fig. 10(a) is used to generate the problem scenarios. The subscriber IPTV flows traverse the simulated network from the video server and are routed to home users through the DSLAMs and the home gateways. We emulate an IPTV delivery network with a video server using iperf. Two DSLAMs are simulated and three home routers are created using NS-3 models. Each node in the network collects their respective metrics and generates metric CSV files. The network log data is then reported to the information uplift engine. The uplift engine performs the necessary steps to correctly identify the source(s) of the problem, the particular metric threshold(s) that were breached, and then, the uplift engine uses this information to suggest a solution to the NM.

5.3.1 Scenario 1: Excessively high latency at the GW is causing poor QoE

Using a simulation script, we cause a single point of failure to occur in the network. Naturally, from the perspective of failure detection and location, we assume the NM is unaware of the time and location of the failure. The uplift engine inspects the CSV entries from the


Problem Detection
hasSource id:vserver92
hasDestination id:gateway66
IPTV
happenedOn IPTV
causedBy IPTV_low_quality
Anomaly! QoE_low
Problem Analysis
probably_caused_by DSLAM_problem
not_caused_by Gateway_problem
not_caused_by VSERVER_problem
Identify Problem
IPTV_low_quality
probably_caused_by by_port_severly_errored_ seconds_exception
happenedOn id:1804289383
Problem Detection
Anomaly! QoE_low
causedBy IPTV_low_quality
happenedOn IPTV
IPTV
hasDestination id:gateway66
hasSource id:vserver92
Problem Analysis
not_caused_by VSERVER_problem
IPTV_low_quality
may_be_caused_by Gateway_problem
may_be_caused_by DSLAM_problem
Identify Problem
happenedOn id:1804289383
probably_caused_by by_port_severly_errored_ seconds_exception
IPTV_low_quality
may_be_caused_by GW_latency_high
happenedOn id:gateway66
Problem Detection
Anomaly! QoE_low
causedBy IPTV_low_quality
happenedOn IPTV
IPTV
hasDestination id:gateway66
hasSource id:vserver92
Problem Analysis
probably_caused_by VSERVER_problem
may_be_caused_by Gateway_problem
IPTV_low_quality
not_caused_by DSLAM_problem
Identify Problem
happenedOn id:vserver92
may_be_caused_by resource_utilization_rate_low
not_caused_by MOS_exception
happenedOn id:gateway66
happenedOn id:gateway67
happenedOn id:gateway68

(a) Scenario 2: Anomaly analysis performed when a high number of severely errored seconds at the DSLAM causes poor a QoE.

(b) Scenario 3: Anomaly analysis performed when a high latency and number of severely errored seconds contributes to poor QoE.

(c) Scenario 4: Anomaly analysis performed when the resource utilization rate has breached its threshold value at the Video Server causing poor QoE.

Fig. 14 Analysis of IPTV topology outage scenarios for single and multiple points of failure: The problem detection, analysis and identification processes are illustrated for each of the scenarios from left to right. This structured approach allows for drill-down which leads to a suggested course of remedial action.

gateway, DSLAM and video server. This procedure examines the metric values in the CSV files to ascertain which metrics have breached their threshold(s). Problem detection, analysis and identification are described below. This process is visualized in the analysis panel of the visual interface in Fig. 10(b).

Detection: The uplift engine receives metrics from the network nodes to detect if the customer is receiving poor quality of IPTV service. Poor QoS may be attributed to packet loss, excessive latency, higher jitter, or low video access rates, each of which may effect the QoE of the IPTV consumers. In this particular scenario, a threshold is exceeded and the “IPTV_low_quality” event is uplifted and triggered as an anomaly event by the uplifted semantic attribute. Problem detection is illustrated in the left-most sub-figure of Fig. 10(b).

Analysis: Problem analysis is performed by tracing back along the IPTV flow’s delivery route and examining the maintained events and semantic attributes in the event pool of the relevant nodes. It is then determined if the anomaly event was triggered by the uplifted semantic attributes of any of the nodes in this path. The “IPTV_low_quality” anomaly event was caused by a problem at the subscriber’s gateway, which is indicated by a cross in the center sub-figure of Fig. 10(b).

Identification: Examination of the candidate problem node indicates that the IPTV traffic on “gateway66” is suffering from high latency (right-most sub-figure of Fig. 10(b)). This problem is given as the most likely root-cause for the “IPTV_low_quality” anomaly detected above. This completes the problem drill-down process for this scenario.

Solution: The recommended action is to push configuration changes to the network as per operational guidelines.

5.3.2 Scenario 2: High number of severely errored seconds at the DSLAM

Similarly to Scenario 1, a simulation script generates a single point of failure. The uplift engine performs its inspection procedure for each CSV entry. The problem is detected, analyzed and identified and this process is visualized in the analysis panel of the visual interface in Fig. 14(a).

Detection: The IPTV flow from the DSLAM to the Gateway experiences a large number of severely errored seconds that cause the end user to experience low QoE. The threshold associated with the number of errored seconds is exceeded and the “IPTV_low_quality” event is uplifted and triggered as an anomaly event by the uplifted semantic attribute.

Analysis: The detection process informs the NM that the quality decrease happened on the DSL link between between “DSLAM1804289383” to “gateway66” in the left-most panel of Fig. 14(a). The information uplift engine suggests that the “IPTV_low_quality” anomaly event may have been caused by a problem on the DSLAM in the right-most panel of Fig. 14(a).

Identification: Once the uplift engine has identified the source of the problem further analysis –in this case on the customer’s connection between the DSLAM and their gateway– indicates that the customer is experiencing a high number of severely errored seconds in the rightmost panel of Fig. 14(a).

Solution: The DSL profile in question is changed to a more stable profile, one with a lower bit-rate.

5.3.3 Scenario 3: High latency and high number of severely errored seconds

In this scenario, the simulation script causes two nodes to be responsible for the degradation in the QoE experienced by the end users –a multi-point of failure scenario. As part of its inspection process, the uplift engine notes that there are metrics in two different nodes that are reporting problems. The detection, analysis and identification processes are illustrated in the visual interface depicted in Fig. 14(b).

Detection: The left-most graph in Fig. 14(b) depicts that the IPTV flow is experiencing an “IPTV_low_quality” anomaly.

Analysis: We assume a human is tasked with implementing the remedial action suggested by our monitoring system. A natural approach is to present the problem amelioration step in an order that focuses on the node that serves the greatest number of customers –a DSLAM. The reason for this is two-fold. First, the DSLAM serves a greater number of customers (in the range of 24 to 48 customers), whereas the gateway only serves one customer. From a service delivery (and financial) point of view, priority should be given to the problem that has the potential to affect the greatest number of customers. In this case, the breached threshold is only localized to one single link, but it is possible to envisage a situation where a problem affects the DSLAM as a whole. Secondly, due to the direction of the flow, it may be that the problem in the GW is a symptom of the problem in the DSLAM. Solving the DSLAM issue first may solve the problem in the GW.

Identification: Using high-level rules, the information engine prompts the NM that this problem is *probably* caused by the "port_severely_errored_seconds_exception" on the DSLAM and *may be* caused by the "GW_latency_high on the gateway". The rule, *probably*, has higher priority than *may be*.

Solution: The remedial action associated with this scenario prompts the NM to reconfigure the GW and DSLAM in line with their operational guidelines. Note, however, that the configuration for DSLAM has a higher priority than the GW.

5.3.4 Scenario 4: Resource utilization rate has breached threshold at the Video Server

The simulation script causes one or more of the GWs to report a low QoE. In addition, the video server reports that its resource utilization rate –the outgoing bandwidth as a percentage of its outgoing link's capacity– has passed its threshold value. This scenario is illustrated in Fig. 14(c).

Detection: The detection process illustrates that the IPTV flow is experiencing an "IPTV_low_quality" anomaly.

Analysis: The uplift engine identifies the nodes responsible for this and the corresponding metrics. In this case, the MOS at the GW and resource utilization rate at the video server are responsible. Based on this indication from the monitoring system, the problem in the video server is tackled first.

Identification: Even though several gateways are suffering a "MOS_exception", they may not have caused the QoE anomaly to send a trigger. The "resource_utilization_rate_low problem" on "videoserver92" may have caused the "IPTV_low_quality" problem. This analysis is presented visually in the right-most panel of Fig. 14(c).

Solution: The system recommends that the NM reduce the bit-rate of the videos being transmitted from the video server.

5.4 Semantic Uplift Process Scalability

One common concern for adopting a semantic-based approach is its performance, especially in a large, complex network scenario. In this experiment, our domain experts defined in the region of 100 semantic attributes and semantic segments. After 3 hours of simulation, we observed that each network node has an average of 20 related semantic entities and that the number of stored semantic entities is mainly affected by three factors: the size of the domain knowledge models, the length of the historical tracing window, and the number of nodes and services in the current network model. It is important to establish how the event processing time grows (scales) with the number of semantic entities maintained in the entity pool. Thus a simple test harness was established where controlled numbers of semantic entities could be added to the entity pool and a new event processing task was created and its execution

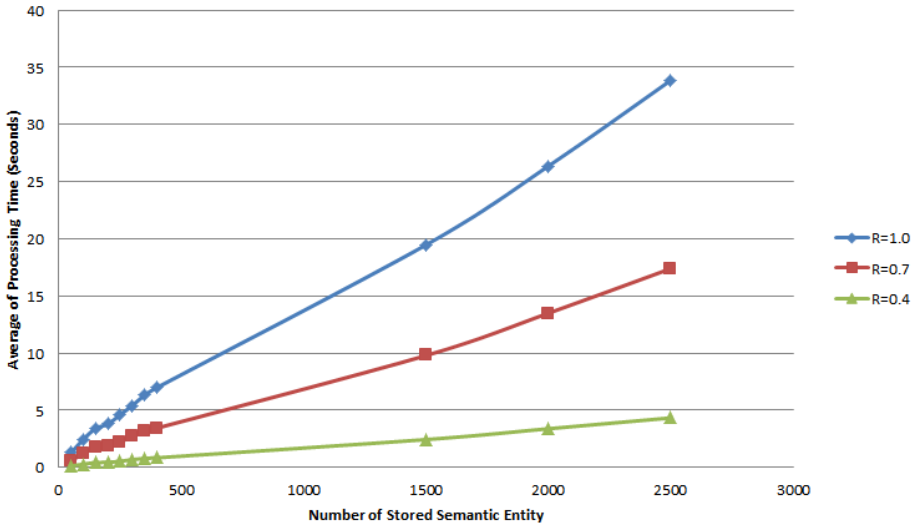


Fig. 15 Average Processing Time based on Number of Stored Semantic Entities

time recorded. This test was conducted for a fixed network size. Initial investigations showed that that the ratio (R=NSS/NSA) between the Number of Semantic Segments (NSS) and the Number of Semantic Attributes (NSA) in the entity pool also affect the processing speed. According to the experimental result in Fig. 15, balancing the ratio R is an effective way to improve the system performance. This ratio in turn is based on the rate at which the network logs are sampled to establish semantic segment values. In our prototype system, we set the time interval for initiating the inspection process for the entity pool at 10 seconds, which was sufficient for network scalability in this test environment. Further study is required to establish the relationship between network size or domain model expressivity and event processing time.

6 Discussion and Conclusions

Investing in IPTV infrastructure alone will not guarantee improved IPTV customer satisfaction. In this paper, we make a significant contribution towards engineering a monitoring system for IPTV. This contribution joins up and improves on much of the monitoring work that has been done in isolation on various monitoring components in the literature. We identify which metrics should be collected and which network elements should be monitored so that the NM is cognizant of a range of IPTV events and their affect on subscriber QoE. A method for IPTV semantic uplift is contributed that annotates IPTV events so that the NM may perform trouble-shooting based on event identification information in a manner that is enriched by the event's context. An additional contribution lies is the suggested corrective actions given by the system that are inferred from the network element metrics, and rules-set defined for the system.

A detailed IPTV problem specification is given. IPTV monitoring events are categorized according to their point of origin and IPTV events are either access network, distribution network, or End-to-end events. This hierarchy of event types motivates the hierarchical rules that are used to suggest corrective action to the NM. To evaluate the efficacy of this system and to explore the IPTV problem-space, we have defined four problem scenarios that are representative of various aspects of the category hierarchy of the events discussed here. We evaluate this IPTV monitoring solution using these scenarios and a new IPTV emulation-simulation framework. In short, the system achieves problem resolution using a three-step process: detection, analysis and identification. A suite of visualization widgets aid the analysis process. Problem drill-down is achieved by clicking-down through the hierarchy of network elements presented. More importantly, these experiments suggest that this framework may be extended to consider more metrics and rule types.

In future work, we will emulate the core network with edge routers using OpenFlow. The number of DSLAMs to be emulated will be increased and a larger number of home routers will be created using VirtualBox. The aim of this work will be to evaluate the scalability of the approach taken here both in terms of metric collection protocols and the visualization widgets. In addition, the use of OpenFlow will move our experimental test-bed work closer to a real-world scenario. Additional videos and screen shots of the system in operation and the work done to date are available at www.fame.ie. To make the simulation/emulation environment reproducible, the IP routing table used are available on request. To conclude, in this paper, we addressed the fundamental problem of IPTV service delivery and how the NM can be cognizant of (and act on) a low subscriber QoE. We presented a system for IPTV monitoring and presented what metrics should be collected, where metrics should be collected, and how these metrics should be presented to a Network Manager (NM) in a semantically enriched way.

Acknowledgements The authors would like to thank the anonymous reviewers and the associated editor, Ajita John, for their detailed comments and suggestions. The revised manuscript has benefited in terms of presentation and clarity. This work has received support from Science Foundation Ireland via grant 08/SRC/I1403 FAME SRC (Federated, Autonomic Management of End-to-End Communications Services - Strategic Research Cluster) and via grant 11/TIDA/I2024.

References

1. Cisco visual networking index, Global IP Traffic Forecast,2011-2016, 2011.
2. A. Begen, T. Akgul, and M. Baugher. Watching Video over the Web: Part 2: Applications, Standardization, and Open Issues. *Internet Computing, IEEE*, 15(3):59–63, May 2011.
3. Y. Xiao, X. Du, J. Zhang, F. Hu, and S. Guizani. Internet Protocol Television IPTV: The killer application for the next-generation Internet. *Comms Mag., IEEE*, 45(11):126–34, 2007.
4. S. Balasubramaniam, J. Mineraud, P. Perry, B. Jennings, L. Murphy, W. Donnelly, and D. Botvich. Co-ordinating Allocation of Resources for Multiple Virtual IPTV Providers to Maximize Revenue. *Broadcasting, IEEE Transactions on*, 57(4):826–39, Dec. 2011.
5. S. Balasubramaniam, J. Mineraud, P. McDonagh, P. Perry, L. Murphy, W. Donnelly, and D. Botvich. An Evaluation of Parameterized Gradient Based Routing with QoE Monitoring for Multiple IPTV Providers. *Broadcasting, IEEE Transactions on*, 57(2):183–94, Jun. 2011.
6. 301 192: Digital Video Broadcasting (DVB). ETSI, EN (2004).
7. J. Evans, A.C. Begen, J. Greengrass, and C. Filsfils. Toward Lossless Video Transport. *Internet Computing, IEEE*, 15(6):48–57, Nov. 2011.
8. J. Greengrass, J. Evans, and A.C. Begen. Not All Packets Are Equal, Part I: Streaming Video Coding and SLA Requirements. *Internet Computing, IEEE*, 13(1):70–5, Jan. 2009.
9. Chuanfei Luo, Jun Sun, and Hongkai Xiong. Monitoring and Troubleshooting in Operational IP-TV System. *Broadcasting, IEEE Transactions on*, 53(3):711–18, Sep. 2007.
10. K. Kerpez, D. Waring, G. Lapiotis, J.B. Lyles, and R. Vaidyanathan. IPTV service assurance. *Communications Magazine, IEEE*, 44(9):166–72, Sep. 2006.
11. Triple-play Services Quality of Experience Requirements. Tech. rep., Broadband Forum (2006). Arch. & Trans. Work. Grp.

12. A.C. Begen, C. Perkins, and J. Ott. On the use of RTP for monitoring and fault isolation in IPTV. *Network, IEEE*, 24(2):14–9, Mar. 2010.
13. R. Doverspike, Guangzhi Li, K.N. Oikonomou, K.K. Ramakrishnan, R.K. Sinha, Dongmei Wang, and C. Chase. Designing a Reliable IPTV Network. *Internet Computing, IEEE*, 13(3):15–22, May 2009.
14. Hyun-Jong Kim and Seong-Gon Choi. A study on a QoS/QoE correlation model for QoE evaluation on IPTV service. In *Advanced Communication Technology. The 12th International Conference on*, volume 2, pages 1377–82, Feb. 2010.
15. A. Takahashi, D. Hands, and V. Barriac. Standardization activities in the ITU for a QoE assessment of IPTV. *Communications Magazine, IEEE*, 46(2):78–84, Feb. 2008.
16. J. Welch and J. Clark. RFC4445: A Proposed Media Delivery Index (MDI). Network Work. Grp., 2006.
17. IPTV QoE: Understanding and interpreting MDI values. Agilent Technologies, 2006.
18. H.J. Kang, M.S. Kim, and J.W.K. Hong. A method on multimedia service traffic monitoring and analysis. *Self-Managing Distribute Systems*, 2867:93–105, 2003.
19. J. van der Merwe, R. Cáceres, Y.H. Chu, and C. Sreenan. MMDump: a tool for monitoring internet multimedia traffic. *ACM SIGCOMM Comput. Commun. Rev.*, 30(5):48–59, Oct. 2000.
20. Shu Tao, J. Apostolopoulos, and R. Guerin. Real-Time Monitoring of Video Quality in IP Networks. *IEEE/ACM Transactions on Networking*, 16(5):1052–65, Oct. 2008.
21. Ludovic Noirie, Emmanuel Dotaro, Giovanna Carofiglio, Arnaud Dupas, Pascal Pecci, Daniel Popa, and Georg Post. Semantic networking: Flow-based, traffic-aware, and self-managed networking. *Bell Labs Technical Journal*, 14(2):23–38, 2009.
22. J.E. López de Vergara, A. Guerrero, V.A. Villagrá, and J. Berrocal. Ontology-based network management: Study cases and lessons learned. *J. Net. Sys. Man.*, 17(3):234–54, 2009. Sp. Iss.: Ontological Approaches for Net. and Serv. Man.
23. A. Viswanathan, A. Hussain, J. Mirkovic, S. Schwab, and J. Wroclawski. A semantic framework for data analysis in networked systems. *Proc. 8th USENIX Symp. Net. Sys. Design and Implementation*, 2011.
24. H. Lee and T. Kim. IPTV network error diagnosis framework based on task ontology. *Asia Pacific Industrial Engineering and Management Society Conf.*, pages 1898–1904, 2009. Kitakyushu, Japan.
25. J.C. Hoag and F.A. Hayes-Roth. Semantic reasoning for adaptive management of telecommunications networks. In *Systems, Man and Cybernetics. IEEE International Conference on*, volume 1, pages 127–31, Oct. 2006.
26. Antonio Guerrero, VíctorA. Villagrá, Jorge E. López Vergara, and Julio Berrocal. Ontology-Based Integration of Management Behaviour and Information Definitions Using SWRL and OWL. In Jürgen Schönwälder and Joan Serrat, editors, *Ambient Networks*, volume 3775 of *Lecture Notes in Computer Science*, pages 12–23. Springer Berlin Heidelberg, 2005.
27. Liam Fallon, Yangcheng Huang, and Declan OSullivan. Towards automated analysis and optimization of multimedia streaming services using clustering and semantic techniques. In Rob Brennan, II Fleck, Joel, and Sven Meer, editors, *Modelling Autonomic Communication Environments*, volume 6473 of *Lecture Notes in Computer Science*, pages 12–23. Springer Berlin Heidelberg, 2010.
28. M. Feridun and A. Tanner. Using linked data for systems management. *Network Operations and Management Symposium (NOMS), IEEE*, pages 926–9, 2010.
29. J. Strassner, J.N. De Souza, D. Raymer, Srini Samudrala, S. Davy, and K. Barrett. The design of a new policy model to support ontology-driven reasoning for autonomic networking. *Latin American Network Operations and Management Symposium (LANOMS)*, pages 114–25, 2007.
30. S.S. Seo, A. Kwon, J.M. Kang, and J.W. Hong. OSLAM: Towards ontology-based SLA management for IPTV services. *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 1228–34, 2011.
31. Henar Muoz Frutos, Ioannis Kotsiopoulos, Luis Miguel Vaquero Gonzalez, and Luis Rodero Merino. Enhancing Service Selection by Semantic QoS. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvnen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Simperl, editors, *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 565–77. Springer Berlin Heidelberg, 2009.
32. J. López de Vergara, J. Aracil, J. Martínez, A. Salvador, and J. A. Hernández. Application of ontologies for the integration of network monitoring platforms. In *Proceedings of the 1st European Workshop on Mechanisms for Mastering Future Internet*, Jul. 2008. Salzburg, Austria.
33. C. Hampson and O. Conlan. Supporting personalized information exploration through subjective expert-created semantic attributes. In *Semantic Computing. IEEE International Conference on*, pages 384–89, Sep. 2009.
34. IBM Tivoli Performance Management Suite. IBM.

Ruairí de Fréin received his PhD in Applied Mathematics, from University College Dublin in 2010. He is a Research Fellow in the TSSG, Ireland, working on mathematical models of communications networks, with a particular focus on the application of Machine Learning to Network Monitoring. He is undertaking a Marie Curie Research Fellowship with Amadeus SAS, Sofia Antipolis, France. His research interests include Scalable IPTV and Femtocell Monitoring.

Cristian Olariu received his B.Eng. in 2008 from Politehnica University of Timisoara, and his Ph.D. in 2013 from Waterford Institute of Technology. He is currently a post-doctoral research fellow with University College Dublin. His interests are in the field of VoIP, QoS for VoIP, and Wireless Mesh and Cellular Networks.

Yuqian Song is a Ph.D. candidate in KDEG, Trinity College Dublin, Ireland. His research focuses on using domain expert knowledge to enable the real time uplift of meaningful information from raw data to support non-expert users in understanding and monitoring network systems.

Rob Brennan is a senior research fellow at Trinity College Dublin. His research interests include semantic inter-operability, intelligent distributed systems and the application of linked data to systems management. He has contributed to 3GPP, TMF, IETF and OMG standards. His Ph.D. (2005) is from Dublin City University.

Patrick McDonagh received his B.Sc. in Computer Science from University College Dublin in 2009. He subsequently received his Ph.D. in 2012 from University College Dublin. His research is focused in the areas of methods of monitoring and managing carrier-grade services in wired and wireless access networks.

Adriana Hava received a B.Sc. from Politehnica University of Timisoara and currently is a Ph.D. student in Performance Engineering Laboratory from University College Dublin. Her areas of interest include QoS routing and load balancing solutions for video deliveries in wireless mesh networks.

John Murphy is an Associate Professor in Computer Science and Informatics at University College Dublin. He has published over one hundred peer-reviewed journal articles or international conference full papers in performance engineering of networks (mobile and optical) and distributed systems (component and enterprise) and supervised to completion seventeen PhD students.

Liam Murphy is Professor of Computer Science and Informatics at University College Dublin, where he is Director of the Performance Engineering Laboratory. Prof. Murphy has published approximately 150 refereed journal and conference papers on various topics, including dynamic and adaptive resource allocation algorithms in computer/communication networks, and software performance.

Paul French is a Senior Technical Staff Member at IBM Tivoli Software, Cork, Ireland. He is currently an LTE Network Management solution architect at IBM Ireland.