Network Monitoring Architecture based on Home Gateways

Claudio Casetti, Yan Grunenberger, Frank den Hartog, Anukool Lakhina, Henrik Lundgren, Marco Milanesio, Anna-Kaisa Pietilainen, Renata Teixeira, Shuang Zhang

Future Network and MobileSummit 2013 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2013
ISBN: 978-1-905824-36-6

Poster Paper

Network Monitoring Architecture based on Home Gateways

Claudio Casetti[1], Yan Grunenberger[2], Frank den Hartog[3], Anukool Lakhina[4], Henrik Lundgren[5], Marco Milanesio[6], Anna-Kaisa Pietilainen[7], Shuang Zhang[3], Renata Teixeira[7]
[1]*Politecnico di Torino, Italy* [2]*Telefonica, Spain* [3]*TNO, Netherlands* [4]*Guavus, India*
[5]*Technicolor, France* [6]*Eurecom, France* [7]*UPMC, France*

Abstract: The "Future Internet Gateway-based Architecture of Residential netwOrks (FIGARO)" project proposes to tackle the new challenges arising from the shift of the Internet use from technology centric to user/content centric with a novel network architecture centered on the residential gateways. Many use cases for the FIGARO architecture such as home automation, distributed content management, content delivery optimizations, network performance monitoring and troubleshooting require advanced network monitoring functionality on the residential gateway. In this paper, we discuss the requirements and design of the FIGARO gateway-centric network monitoring architecture.

Keywords: Network performance, performance monitoring, home networks

1. Introduction

Residential access networks are seeing steady deployment. Over the past decade, Internet usage has grown by more than 3.5 times, to about 1.6 billion users, about 300 million of which are broadband subscribers[1]. We expect that this growth will continue and that the home will become a central place for end-users to access the Internet and to store and share their personal content. To implement this vision, the FIGARO project[2] proposes a Future Internet architecture that is structured around residential networks. Given that the home gateway connects the home with the rest of the Internet, we see it as the ideal place to implement the functionalities of this new architecture. A gateway-centric approach enables efficient network management in terms of monitoring of network, application, and services; as well as automatic troubleshooting and network optimizations. The FIGARO architecture also enables a number of new functionalities such as home automation control, distributed content management, and content delivery optimizations. All these new functionalities require the development of active and passive monitoring techniques that can run continuously and online inside a large number of home gateways.

Monitoring in the FIGARO context poses new challenges. First, gateways have limited CPU, memory, and storage. Even if in FIGARO we are designing a new generation of gateways, we cannot significantly increase the cost of each gateway. Monitoring and troubleshooting tools have to run online and fast; in particular, tools based on passive monitoring have to process every packet efficiently. Most existing monitoring and troubleshooting tools run offline or online but in dedicated and powerful machines. When used in a resource-limited environment like the gateway these techniques overload the

[1] `http://www.internetworldstats.com/dsl.htm`
[2] `http://www.ict-figaro.eu`

CPU and jeopardize the performance of the whole system. FIGARO therefore is developing novel monitoring techniques that work only with access to the home gateway without overloading it. Second, given that the gateway has limited storage, FIGARO will require efficient techniques to summarize data without loss of "essential information". The notion of essential information depends on the use case and FIGARO integrates different types of content with diverse requirements. Hence, FIGARO will define and implement an interface for other modules to query monitored data (either directly or via alarms). Finally, many of the FIGARO use cases require collecting data from a large number of gateways. The FIGARO monitoring architecture must scale to possibly millions of home gateways.

Although there is much literature treating network measurement and monitoring techniques, monitoring of home and access networks is still in its infancy and FIGARO's approach of centralizing all monitoring capabilities at the gateway is unique. In this paper, we present the requirements and design of the FIGARO network monitoring architecture, overview the key monitoring modules, and discuss some of the on going work in the project.

2. Requirements

The FIGARO network monitoring component has two main tasks: (i) to collect and to store monitoring data from various vantage points in the residential network, and (ii) provide an interface for other components of the FIGARO architecture to query this data. The supported vantage points are illustrated in Figure 1: Home Network (green, HN) usually Ethernet or WLAN; Home Network Gateway (blue, Gateway); Internet Access Link (red, AL) typically DSL or Cable; and Link(s) to Neighbor(s) (blue, NL), which is usually WLAN. The various FIGARO use cases have various requirements in terms of the actual metrics and how the data is accessed. The monitored metrics are grouped in three broad classes:

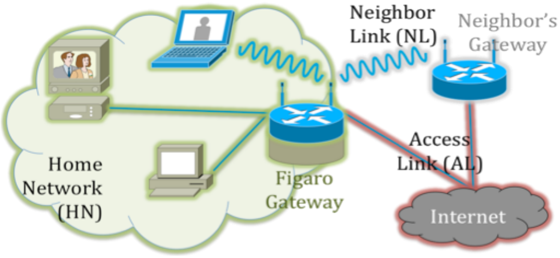

Figure 1: Monitoring vantage points.

- **Network performance metrics**, *e.g.*, network and device reachability, available bandwidth, achieved troughput, MAC layer goodput, access link characteristics, wireless link characteristics, frame and packet error rates and latency of various links.
- **Traffic characteristics**, *e.g.*, application/service usage mix and performance, connection failures and video stream information.
- **Gateway status**, *e.g.*, available storage, available links and clients, home network topology, CPU and memory.

The data query interface must support on-demand queries for both historical and current raw and aggregated values of various metrics, and access to continuous stream of real-time updates on various metrics.

3. System Design

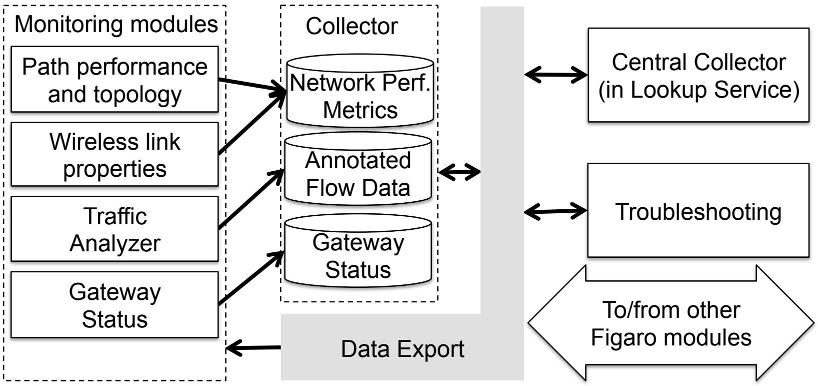

Figure 2: Monitoring framework functional graph.

Figure 2 presents the functional split of the FIGARO monitoring framework. Various *Monitoring Modules* feed data to a *Collector* that stores the data in a scalable fashion. Other FIGARO modules can query on-gateway data from the monitoring module using the *Data Export* function. A *Central Collector* (located outside of the home, possibly at a cloud or NOC) stores monitoring data long term. In this section we will discuss the design and implementation of these functions.

3.1 Monitoring Modules

There are four main monitoring modules in the current architecture.

Path performance and topology This category groups tools that use active probing to infer properties of network paths inside the home network and the access link. We focus on available bandwidth, latency, reachability, and topology (in particular the topology of the home network).

Wireless link properties Many home networks today use 802.11 to interconnect devices inside the home and, in FIGARO, we also assume that a device in a home network will connect to a neighbors gateway over WiFi. This group of tools will measure the WiFi links (both inside the home and to the neighbor). In particular, we collect the MAC layer goodput, achieved throughput, frame error rate; the wireless physical layer bit rate; the channel busy time; SNR and RSSI; and the IP layer packet error rate.

Traffic analyzer This category passively monitors the traffic that traverses the gateway to track active applications and their performance. The traffic analyzer maps packets into flows (using the typical 5-tuple to identify flows: source and destination IP, source and destination port, and protocol). Then, it extracts for each flow: the application that generated the flow, typical flow statistics; and performance metrics of the flow (for instance, achieved throughput). This module also generates alarms when application performance is poor.

Gateway status The last category samples the resource consumption (in particular, the available storage capacity and the CPU and memory utilization) and the status of the gateway (Internet access synchronization rate and the on-periods).

3.2 Collector

The Collector system is designed to ingest data from different sources - both streaming or batched, and from socket or file interfaces. The Collector can be optionally configured to: (i) filter or subsample data, (ii) merge multiple streams of data by a user-defined operation into a new combined stream, (iii) organize (or bin) the data along time boundaries, and finally, (iv) the data stream can be aggregated into summary projections based on a priori interest in specific fields of the data. Such summary projections can be used to preprocess the data to be stored to reduce the raw data volume. The above described steps are organized as a pipeline-based streaming architecture which allows the data streams to be continuously processed first, and then inserted into a database, enabling low-latency processing.

Data is stored in a circular disk buffer on-gateway so that old data is overwritten by fresh incoming data. The size of the circular buffer is configurable. However, because the on-gateway storage is limited, the Collector also sends old data records into an off-gateway data center / cloud (which we call the Central Collector), where the data will be archived and analyzed without the compute limitations of the gateway. The on-gateway Collector sends data to the Central Collector periodically, either when the circular buffer is full or after a configurable timeout, whatever happens first.

3.3 Data Export

The Data Export will interact with the on-gateway Collector and the Central Collector to provide all the different views to the data (on-demand - continuous, historical - current, raw - aggregate). The Data Export provides two basic access mechanisms: (i) Publish/subscribe: publish the raw stream continuously to any module that subscribes to it; and (ii) data export will also allow modules to perform SQL queries on data stored in a given gateway or in the long-term storage in the Central Collector.

A FIGARO module can use the first mechanism to subscribe to the raw stream when it needs recent data or a continuous data stream. Modules can unsubscribe when complete or stay subscribed to the feed forever; for example, the Central Collector in the cloud is built by continuously subscribing to the raw stream from each gateways Collector. A module can use the second mechanism when it needs a specific event or query a specific period of time in the past. Some measurements can be done on-demand. In this case, the Data Export explicitly triggers the measurement and subscribes FIGARO module that requested the measurement to the resulting data stream. The module will unsubscribe itself when done. This design gives the flexibility to the modules to ingest data and keeps the Collector stateless and simple.

4. On-Going Work

The FIGARO Future Internet architecture builds on residential gateways. Performance monitoring is a key component of the FIGARO architecture. In this paper we have presented the requirements and design of gateway-based monitoring framework. The on-going work in the FIGARO project includes research on efficient passive online application identification techniques, online prioritization of traffic based on QoE and application classification and development of troubleshooting tools to identify the root cause of network performance degradations (*e.g.* to detect if the problem is at the home, on the access link or in the network provider, or service provider network).