The Global Data Plane Visualization & Monitoring Application: A White Paper

Neil Agarwal
University of California, Berkeley

Abstract

The Global Data Plane (GDP) hopes to revolutionize the deployment architecture for the Internet of Things and assist the Internet as it adapts to the future [1, 7, 8]. The GDP proposes raising the level of abstraction to a data-centric design focused around the distribution, preservation and protection of information. This paper proposes a solution to monitoring and visualizing the GDP. By integrating the infrastructure of the GDP into the design, we have developed an application that provides a real-time status of the availability of GDP components and services.

1 Overview

1.1 Introduction

By 2020, over 30 billion connected devices will join the Internet with a majority of them falling under the category of the Internet of Things (IoT) [5]. The influx of IoT devices brings a new set of requirements, challenges unique to the IoT. In order to fulfill the demands of the future, we must rethink the architecture of the Internet and how we use the Internet in supporting these new applications.

At UC Berkeley, researchers at the Swarm Lab have proposed a novel approach to meet the scale and unique requirements that the Internet of Things will bring to the Internet. The proposed new middleware, referred to as the Global Data Plane, hopes to revolutionize the deployment architecture for the Internet of Things.

As the GDP undergoes development and research, ensuring the stability and reliability of such a system becomes increasingly more significant. In order for the GDP to scale, there is a need for an application to monitor and detect availability and status of GDP components. This paper describes such an application for monitoring and visualizing the Global Data Plane with an intended audience of GDP administrators and developers.

1.2 Paper Organization

Section 2 provides a general overview of the Global Data Plane. Section 3 explains the need for monitoring the GDP and requirements for a monitoring solution. Section 4 presents the visualization and monitoring application. Section 5 discusses the implications of this application and what it means to network monitoring, the IoT and the Internet.

2 The Global Data Plane

2.1 The Internet of Things

By the end of 2017, over 8.4 billion connected things will be in use and expected spending on endpoints and services will reach almost $2 trillion [2]. The generation of smart and connected devices has arrived and it is expected to stay for a long time to come. Coined the "Internet of Things" by Kevin Ashton in 1998, the idea of connecting objects or "things" to the Internet has since erupted into a pivotal transformation redefining the role that technology plays in society [4]. With IoT devices flooding the consumer market targeting a variety of applications such as wearables or home security, we must ensure that the Internet infrastructure will be capable of handling such an influx of traffic.

2.2 Architectural Concerns

Currently, devices within the Internet of Things are being connected directly to the cloud, a seemingly natural and simple way of constructing a distributed application [7]. Yet, relying on web service calls to access the cloud and securely transfer data between devices and storage within the realm of IoT is not an effective solution. Current web services and cloud interfaces do not meet requirements for supporting the huge volume of Internet of Things [8].

When IoT applications come into the picture, different properties and characteristics become much more significant. For example, ambient data collection IoT applications require certain levels of privacy and security. Even encrypted IoT traffic can be analyzed to reveal potentially sensitive data [3]. By identifying individual device flows and examining changes in traffic rates, a passive network observer gains access to an array of behavioral information of the user. Furthermore, real time applications, such as video streaming services, have low latency requirements which are not easily supported in an open cloud environment. The list of pitfalls with today's approach to IoT continues when considering scalability, quality of service guarantees, durability management, and bandwidth saturation [8].

2.3 A New Approach

The status quo necessitates the construction of a uniform global data plane built specifically for the IoT that is able to meet the diversity of requirements of smart devices. This is the primary motivation for UC Berkeley's Global Data Plane (GDP) [1]. The GDP provides a middleware for IoT applications. Masking the heterogeneity of underlying communication paradigms, network/storage devices, and physical connections, GDP raises the level of abstraction to focus on data [8]. The main architecture of the Global Data Plane takes the form of interconnected log streams. Each IoT device is represented by one or more single-writer, multi-reader log files. The GDP connects such log files with applications, creating a global data plane for the Internet of Things. By focusing on routing the flow of data, the need to create stovepipe solutions for each type of connection and gateway disappears [8]. The GDP hopes to provide a standardized middleware for the Internet of Things.

2.4 GDP Conceptual Architecture

The GDP is composed of GDP Routers, GDP Clients, and GDP Log Servers. IoT devices connect to the GDP as clients, either as a reader or a writer of data. For example, a sensor would connect to the GDP as a GDP writer, publishing produced data to a log located on a GDP Log Server. An actuator would connect to the GDP as a GDP reader, subscribing to a GDP log. Every GDP client and GDP Log Server is connected to a GDP Router. The GDP Routers form a layer of communication between GDP clients and GDP Log Servers, routing data to and from each entity through a publish/subscribe mechanism. Each GDP client and each GDP log is represented by a unique 256-bit GDP identifier, configured independent of the location of the GDP client or log. This GDP identifier is a hash of the metadata which includes the public key. In addition to allowing the GDP to employ location-independent routing, this provides a means of verification for all data sent and received through the Global Data Plane. See Figure 1 for a conceptual overview of the GDP.

From the perspective of an IoT device or an application, the GDP provides a uniform infrastructure for storing and accessing data as well as an ability to communicate with other IoT devices/applications [8]. However, hidden under the simplicity of the GDP interface is a complex layer of interconnected devices, logs, and routers. Access control, authentication and integrity checks, and encryption provide secure communication between each entity. Log replication provides durability and high availability of data. The level of complexity magnifies when the scale of the Internet of Things is considered and millions of devices join the GDP.

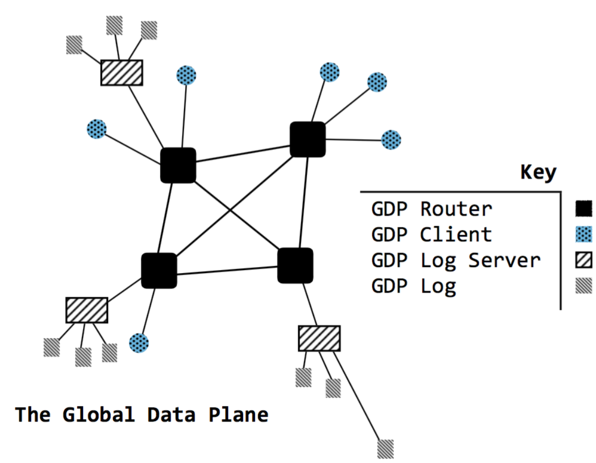

Figure 1: GDP Overview

3 Monitoring the GDP

3.1 The Problem

Maintaining the integrity and performance of the GDP requires a real-time understanding of its state and performance. Currently, there is a lack of a centralized and accessible means of monitoring the GDP. The underlying architecture of the GDP comprises a complex interconnected array of entities, requiring intensive individual and manual monitoring. As the GDP scales to millions of entities, an automated mechanism of monitoring becomes a requirement.

Currently, all operational statistics of the GDP are transient. Activity data from moments ago is not available. The lack of historical data becomes a bigger issue when performance analysis needs to be done over periods of time. Such data would enable discovery of unusual and perhaps, undesired trends and would provide system administrators with the information necessary to properly operate the GDP, maintain its availability and meet service level agreements (SLAs).

The GDP currently lacks the infrastructure for complete and thorough system monitoring. There is a conclusive need for the collection, accessible storage, and visualization of real-time performance statistics.

3.2 Solution Requirements

In order for a GDP monitoring solution to be effective, it must meet the following requirements:

- Provide real-time status of each GDP component
- Offer a means of retrieving historical data
- Scale with the growth of the GDP
- Be highly available
- Integrate with existing implementation of GDP
- Provide mechanisms for monitoring the monitoring solution itself

4 The Global Data Plane Visualization & Monitoring Application

We present the Global Data Plane Visualization & Monitoring Application (GDPVMA) - a project that demonstrates a resilient solution for monitoring the GDP.

The rest of this section is organized as follows. The first subsection outlines the premise behind the concept of "Using the GDP to debug the GDP", where we integrate the design of our monitoring application into the GDP architecture. The second subsection presents an overview of the monitoring and visualization application.

4.1 Using the GDP to Debug the GDP

The GDP Visualization and Monitoring Application first tackles the challenge of monitoring network topology. As the GDP is composed of a number of routers and clients, understanding and monitoring the connections between each entity becomes important. Recall that the layer of communication existing in GDP is supported by GDP Routers. Each GDP client or GDP log is hosted by a single GDP Router - the job of the GDP Router is to relay messages to and from other GDP Routers, or GDP clients and logs that it hosts. In order to determine where it must forward a message, a GDP Router maintains knowledge about the GDP clients and GDP logs hosted by all other GDP Routers. This knowledge is maintained through the GDP Routing and Maintenance Protocol, a layer of communication between GDP Routers [6]. Upon changes in its connections with GDP routers, GDP clients or GDP logs, a GDP Router immediately notifies all other GDP Routers via this protocol. "Using the GDP to debug the GDP" begins with the idea of listening in on communication over the GDP Routing and Maintenance Protocol. This provides a dynamic push-based mechanism to determine changes in network topology without the need to modify any of the GDP components.

To facilitate the process of intercepting messages across the GDP Routing and Maintenance Protocol, we suggest the creation of a new entity - the GDP Monitor, a subclass of the GDP Router. Functionally a GDP Router, it will be notified of all messages sent across the GDP Routing and Maintenance Protocol. As a GDP Monitor, it will also have the enhanced capability to parse and persist those messages.

The second application of "Using the GDP to debug the GDP" occurs in the processing and storage of messages captured by the GDP Monitor. Embedded in the GDP Monitor is a GDP client that publishes data captured by the GDP Monitor to a designated GDP log, dubbed the Monitoring GCL (GCL stands for GDP Channel/Log). The Monitoring GCL provides intermediate storage of monitoring data. Then, another GDP client embedded in the GDPVMA subscribes to this Monitoring GCL. The publish/subscribe functionality of the GDP notifies a GDP client each time its subscribed GCL log is appended. Leveraging the infrastructure of the GDP for the monitoring process enables segregation between the data collecting and the data storing process. Using the GDP itself for self monitoring allows retention of the push-based notification characteristic of the GDP Routing and Maintenance Protocol.

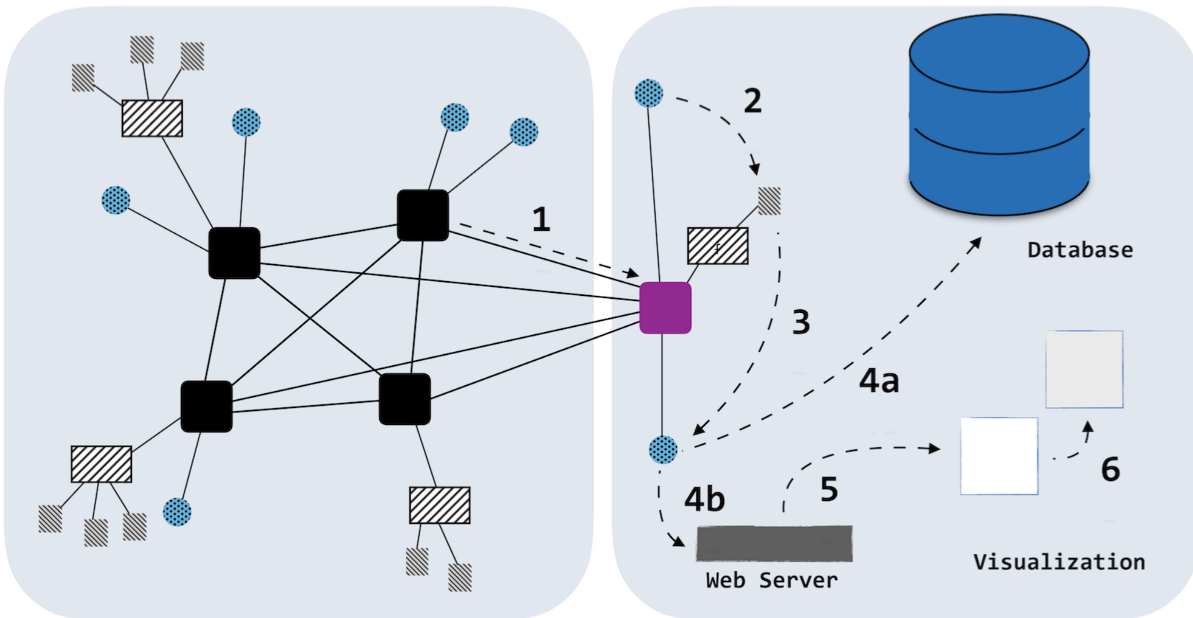

Figure 2: An Overview of the GDPVMA. See section 4.2 for description of steps.

4.2 Overview of Solution

The GDPVMA, a web application, renders a real-time visualization of the Global Data Plane. This visualization provides users the ability to understand the relationships between GDP entities and easily view the state of each component in real time. See Figure 4 for a screenshot of the web visualization.

The solution workflow, as depicted in Figure 2, comprises the following steps:

1. The GDP Monitor continuously receives messages being sent across the GDP Router and Maintenance Protocol.
2. An embedded GDP client in the GDP Monitor publishes these messages to the Monitoring GCL.
3. Another GDP client subscribing to the Monitoring GCL in the GDPVMA is notified of an addition to the Monitoring GCL
4. This GDP client performs 2 actions:
   (a) Ingests the data into a relational database
   (b) Sends the data to the web server
5. The web server receives updates and forwards them to all current instances of the web visualization in browser through WebSockets
6. Upon receiving the message, the web visualization displays an animation and updates the display accordingly, providing real-time updates to the users.

Note that using a storage form other than the GDP logs to store monitoring data such as a relational database enables a much more efficient aggregation and analysis of the data. While the format of a GDP log may assist in maintaining event streams, a relational database allows us to maintain a real-time representation of the current status of the Global Data Plane.

The GDPVMA also includes mechanisms of maintenance and routine testing. Twice a day, a testing service runs integration tests on the infrastructure of the GDPVMA and sends reports to the system administrators upon the completion or failure of each test. These tests ensure the proper end-to-end functioning of the monitoring process. See Figure 3 for sample testing report messages.

5 Conclusion and Future Work

5.1 Summary

As the Internet of Things is becoming the Internet of the present, there is a need to rethink IoT development and deployment architecture. The Global Data Plane (GDP) offers a novel solution for elevating the layer of abstrac-

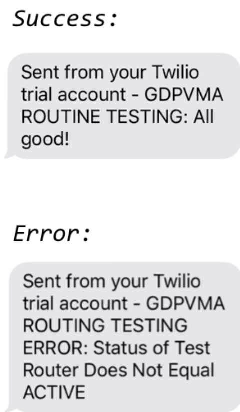

Figure 3: Sample Routine Testing Notifications

tion to a data plane, interconnecting log streams and producing a homogeneous platform for IoT devices. As the GDP proposes to standardize deployment of the IoT, the issue of monitoring and maintaining the GDP becomes significant. The Global Data Plane Visualization & Monitoring Application (GDPVMA) was built as a robust and real-time solution to monitoring and visualizing the GDP.

5.2 Areas for Future Development

The GDPVMA can be developed further to integrate Secondary Routers, GDP Routers that exist behind NAT firewalls. Moreover, an ability to organize the GDP in a hierarchical manner, perhaps federations, through the visualization would provide users, developers, and maintainers of the GDP a much nicer and more accessible view of the GDP.

6 Acknowledgments

I would like to thank Nitesh Mor for introducing me to the Swarm Lab and guiding me for the last year and a half. Thanks for the countless emails, meetings, and lessons. Also, thank you to Eric Allman, Ken Lutz, and Professor John Kubiatowicz for their support and wisdom. I would also like to thank the rest of the Swarm Lab Global Data Plane group.

References

[1] `https://swarmlab.eecs.berkeley.edu/projects/4814/global-data-plane`.

[2] Gartner newsroom. `http://www.gartner.com/newsroom/id/3598917`, Feb 2017.

[3] Noah Apthorpe, Dillon Reisman, and Nick Feamster. A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic. `http://www.internetsociety.org/sites/default/files/Poster-SmartHomeNoCastle.pdf`, 2017.

[4] Kevin Ashton. That 'internet of things' thing. *RFID Journal*, Jun 2009.

[5] Alex Blanter. Internet of things: Changing the world. *A.T. Kearney*, Nov 2015.

[6] Nikhil Goyal, John Wawrzynek, and John D. Kubiatowicz. Global data plane router on click. `http://www2.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-234.html`.

[7] Edward A Lee, Björn Hartmann, John Kubiatowicz, Tajana Simunic Rosing, John Wawrzynek, David Wessel, Jan Rabaey, Kris Pister, Alberto Sangiovanni-Vincentelli, Sanjit A Seshia, et al. The swarm at the edge of the cloud. *IEEE Design & Test*, 31(3):8–20, 2014.

[8] Ben Zhang, Nitesh Mor, John Kolb, Douglas S. Chan, Ken Lutz, Eric Allman, John Wawrzynek, Edward Lee, and John Kubiatowicz. The cloud is not enough: Saving iot from the cloud. In *7th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 15)*, Santa Clara, CA, 2015. USENIX Association.

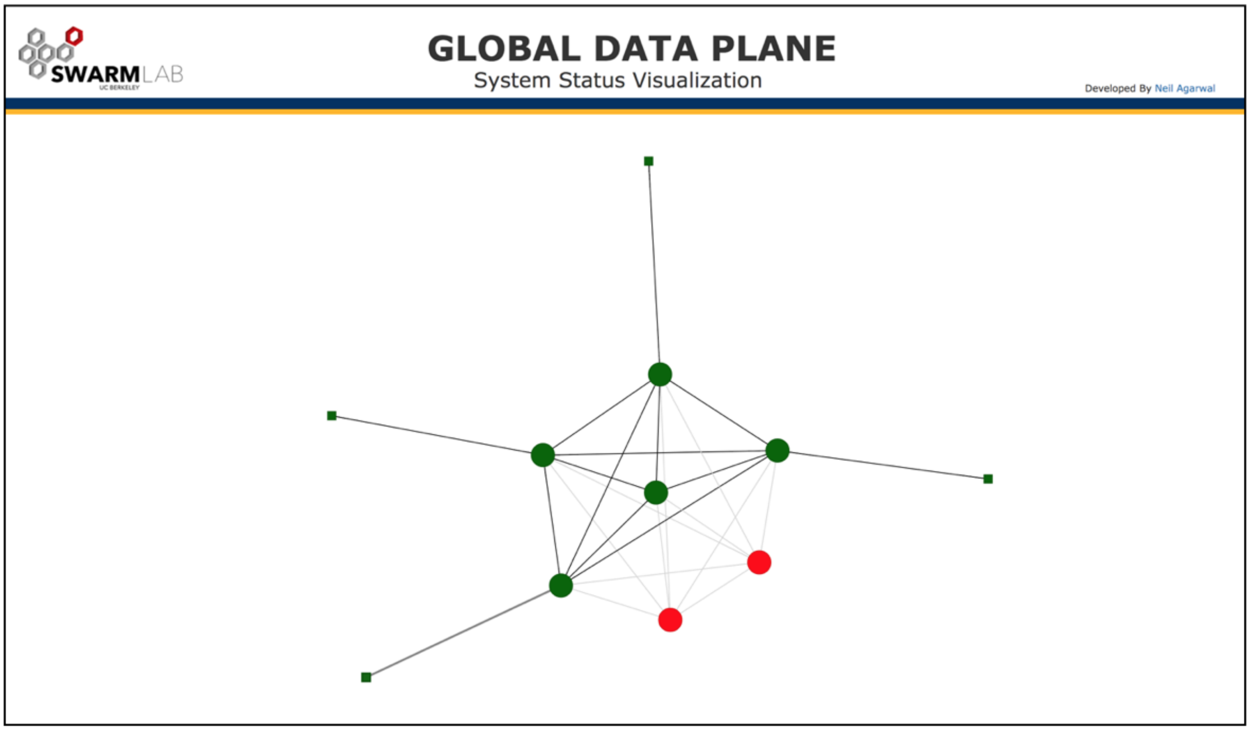

Figure 4: A screenshot of the Web Visualization

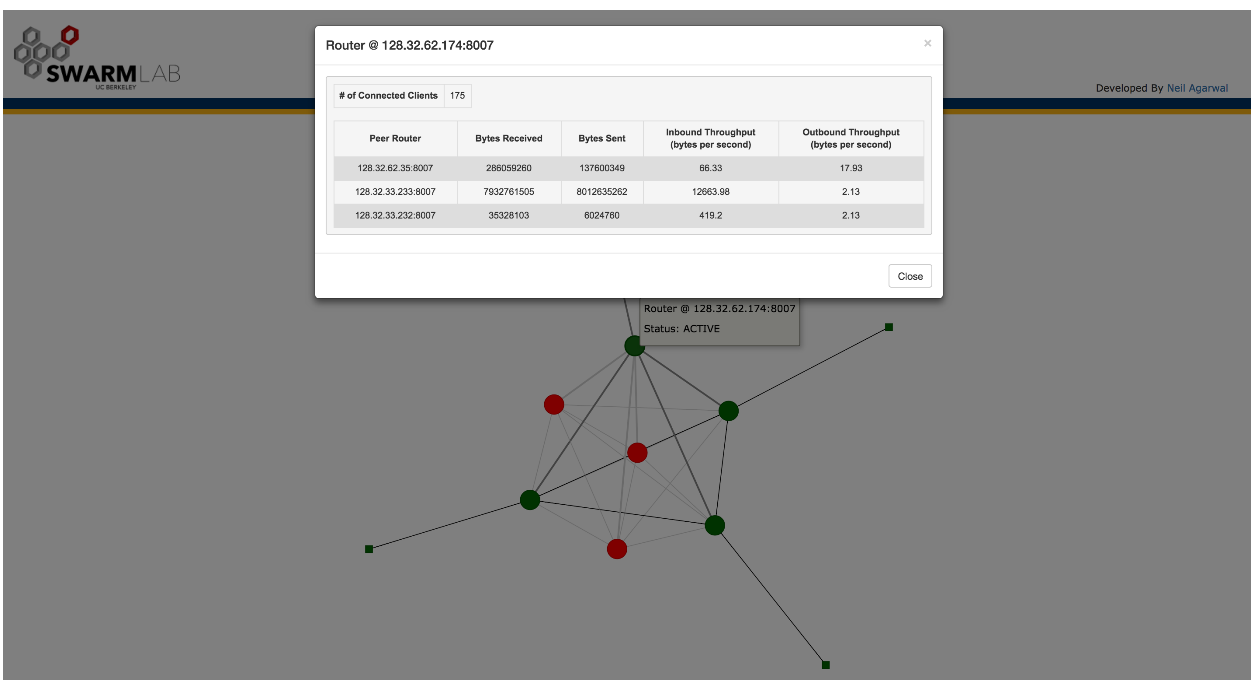

Figure 5: Another screenshot of the Web Visualization