Poster: Towards Robust, Extensible, and Scalable Home Sensing Data Collection

Mohammed Elbadry*, Mengjing Liu*, Yindong Hua*, Zongxing Xie, Fan Ye
{mohammed.elbadry,mengjing.liu,yindong.hua,zongxing.xie,fan.ye}@stonybrook.edu
Electrical and Computer Engineering
Stony Brook University, NY, USA

ABSTRACT

Home-based health monitoring systems are important to many conditions (e.g., aging, chronic diseases). The absence of suitable data collection infrastructure is a fundamental barrier to the development of related algorithms and systems. In this poster, we present Proteus, a robust, extensible and scalable data collection infrastructure, to enable small research teams to manage large deployments. We identify the desired features and achieve them by combining mature technologies and new components:*i)* extensibility with new, diverse sensor types and data formats with a few lines of coding (LOC) efforts; *ii)* scalability in managing sensor/edge devices to automate many deployment, management tasks; *iii)* resilience to system failures and network outage. Experiments on a prototype show zero data loss or system error for one sensor node running 10 days, and 99.95% of data received for 32 emulated sensors sending data at 200 Mbps, 20 and 100 fold reductions in node setup efforts and LOC for new sensor types. The preliminary results show Proteus is promising for large-scale longitudinal deployment of home-based health monitoring.

ACM Reference Format:
Mohammed Elbadry[*], Mengjing Liu[*], Yindong Hua[*], Zongxing Xie, Fan Ye. 2023. Poster: Towards Robust, Extensible, and Scalable Home Sensing Data Collection. In *ACM/IEEE International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE '23), June 21–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3580252.3589431

1 INTRODUCTION

For many health conditions (e.g., aging, chronic diseases), continuous, multi-modality sensing data collection and analysis at home is critical. The lack of suitable data collection infrastructure is a fundamental barrier to home-based health monitoring at scale. There are lots of efforts for a small research team to manage a dozen real home deployments. Moreover, heterogeneous sensing hardware and modalities, constant faults in sensor nodes, and the need of maintaining and updating system and models at scale present challenges to a continuous, multi-modal home-based data collection system.

In this poster, we introduce Proteus[1], a robust, extensible and scalable infrastructure which facilitates in-home, longitudinal data collection. We identify, combine mature technology pieces, engineering practices, and new design components, including : *i)* a data-agnostic pipeline using pub sub; *ii)* A continuous integration and deployment (CI/CD) pipeline, status monitor and version control at edge nodes/servers, with automated networking setup and node registration upon bootstrapping, dockerized in a portable package; *iii)* complementary edge and cloud storage backup and watchdog mechanisms for network and sensor node failures. Preliminary experiences show that we can achieve twenty-fold reduction in manual configuration efforts of 8 edge nodes from 80 minutes to 4 minutes. The stress test at a limited scale show that Proteus works reliably with 32 edge nodes, sending data at an aggregate rate of 200 Mbps, with negligible data loss, sufficient for most home health monitoring applications. Additionally, we observe zero data loss or system error for one edge node running for 10-days continuously. These preliminary results show that Proteus is promising for longitudinal data collection.

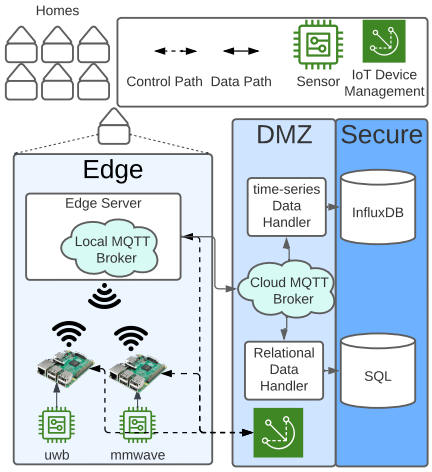

Figure 1: Proteus Design Overview

2 SYSTEM DESIGN

Figure 1 shows the complete infrastructure with data path and control path. We assume there exists home WiFi and Internet access, and most of the data are time series and events that can be stored in relational and time series databases. On edge, edge sensor nodes collect data directly from the sensors and push data to an edge MQTT broker located on the edge server. The edge server further pushes the data to a secure cloud. On the edge server, edge analytics and models can be deployed to process sensitive data that

[1]This work is supported in part by NSF grants 1951880, 2119299.

CHASE '23, June 21–23, 2023, Orlando, FL, USA

ACM ISBN 979-8-4007-0102-3/23/06.
https://doi.org/10.1145/3580252.3589431

[*]These authors contributed equally to this work

cannot leave the home. Once data reach a secure pub/sub MQTT broker in a DMZ (cloud frontend facing external, untrusted networks), two microservice handlers push time series and relational data to respective databases. The control path provides updating and monitoring services on edge remotely.

Data-agnostic Pipeline To support multiple types of data concurrently with minimal code changes in the pipeline, we leverage MQTT pub/sub transport on edge and cloud to make the pipeline independent of data types. We design a per-packet message format and data handlers which describe and read the data format of new types to write the data appropriately to the database.

Scalable Deployment We automate cloud and edge deployment procedures leveraging AWS Greengrass. On edge. Deploying code to dozens of sensors takes a few minutes. To set up a sensor device, it needs to have the appropriate OS image, register a unique ID with AWS Greengrass, and store the unit's identification in database for record. We minimize manual labor by batch copying the OS images and creating a self-registration routine on first boot. On cloud, we design modular code that supports scalable microservices.

Resilience Our system has multiple embedded devices and multi-hop network transport, and each may fail. For resilient operation, we need to monitor the system health with minimal manual labor. To achieve this, we build in *i)* watchdogs on edge to detect abnormal conditions, such as sensor disconnection, network interruption, edge server going offline, etc., to allow quick recovery when errors occur. *ii)* a database at the edge server to store data locally in case publishing to the cloud fails. *iii)* automatic monitoring systems aggregating enterprise software (e.g. Dashboards included with AWS Greengrass, MQTT Broker, and InfluxDB), developer logs, and LED chips that use different lighting patterns to indicate the status of sensor devices to users.

3 EVALUATION

We implement our infrastructure's cloud components on a HIPPA-compliant cloud using virtual machines running Red Hat Enterprise Linux 8.4.1. As for the edge, we use Raspberry Pi 3Bs plus, Ultra-Wide Band (UWB) sensor for sensor nodes and gaming laptops (Intel Core i7, RAM 16GB) as edge servers. Further, we deploy InfluxDB and MySQL to store time series data and relational data.

Scalable deployment Through automated self-registration and batch OS flash image copying, we reduce the setup time of 8 edge nodes from 80 minutes (10 minutes per unit to register on AWS Greengrass) to 4 minutes. To push data from edge sensor to cloud, the developer has to parse the data from the sensor and pass it to our API (~4-5 LOC). Whereas without Proteus, the developer will have to deploy edge server broker, handler to pass data to cloud (~50-80 LOC), cloud broker, database handler (~100-200 LOC), and redundancies to ensure data reliability across each hop (~200-300 LOC). Proteus achieves 100-fold reduction in LOC.

Data loss and Latency We run multiple edge nodes concurrently for an extended period of time to test for data loss and latency at each hop of the infrastructure. Each edge node sends 1 MQTT message per second, including 80 UWB base-band data frames, where the message data size is 0.78MB. The data flow and 4 hops of the infrastructure are shown in Figure 2. In addition, we run multiple threads on the gaming laptop to simulate multiple edge node connections to stress test the system for data loss and latency.

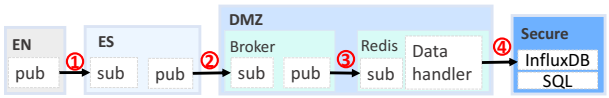

Figure 2: The hop-by-hop data flow from the edge node to the cloud database.

Table 1: Data Loss Rate Hop by Hop

# EN	R	ES	Sub	Redis	DB	EDB	T
4	25.0	0	0	0	0	0%	7
7	43.7	1.00%	0.03%	0	0.10%	11.00%	72
16	102.7	0	0	0	0.10%	1.40%	2
32	205.4	0	0	0	0.05%	7.46%	2

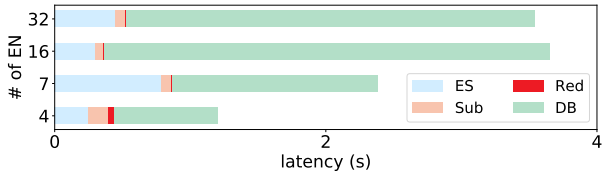

Figure 3: Average Latency Hop by Hop. With 32 sensor node connections sending data at 200Mbps, the end-to-end latency is 3.53 seconds, sufficient for home sensing

We summarize the test results of data loss in Table 1, including the ratio of data stored in the edge storage as a percentage of all data (EDB). Experiments show that Proteus can support data collection with 32 edge nodes sending data at 200 Mbps overall, with negligible data loss (0.05%). With further auto-scaling in cloud, it is promising to support more home deployments. EDB in Table 1 demonstrates the fraction of data backup up to 11% in the event of network failures, proving that backup storage is necessary. Figure 3 shows the average hop-by-hop latency estimated in the experiments with real edge nodes as well as simulated. In the 3-day run test with 7 edge nodes, the average latency from edge node to each hop of the infrastructure is 0.79, 0.86, 0.87, 2.39 seconds respectively.

Related Work Similar works have been done on data collection infrastructure but differ from our goals. VitalCore [1] has developed an analytics and support dashboard that can be integrated with Proteus; it eliminates the tall pipeline of reading HL7 format health data. Multi-modal sensor infrastructure [2] has shown various sets of data being sent to the cloud but did not handle edge computation, updating, monitoring units on the field, or scalability.

4 CONCLUSION

In this poster, we introduced Proteus, a data-agnostic scalable infrastructure for a research team to collect data and run edge analysis at scale. Our infrastructure supports time-series data and events. It leverages mature cloud practices for automatic scaling, deployment, and update using remote edge control for sensors and edge server. Resiliency mechanisms (retry attempts, local edge database storage, watchdogs) are incorporated to ensure minimal data loss. Our preliminary results demonstrate negligible data loss for 200 Mbps data rate from 32 edge nodes, 20× and 100× reduction in setup time and LOC for adding new data types.

REFERENCES

[1] Hyonyoung Choi, Amanda Lor, Mike Megonegal, Xiayan Ji, Amanda Watson, James Weimer, and Insup Lee. Vitalcore: Analytics and support dashboard for medical device integration. In *IEEE/ACM CHASE 2021*, pages 82–86. IEEE, 2021.

[2] Przemyslaw Woznowski, Xenofon Fafoutis, Terence Song, Sion Hannuna, Massimo Camplani, Lili Tao, Adeline Paiement, Evangelos Mellios, Mo Haghighi, Ni Zhu, et al. A multi-modal sensor infrastructure for healthcare in a residential environment. In *IEEE ICCW 2015*. IEEE, 2015.