Initial insights into the performance of Data repositories at the Edge, the case of Store-Process-Send

Adrian-Cristian Nicolaescu
UCL
London, UK
a.nicolaescu@ee.ucl.ac.uk

Ioannis Psaras
UCL
London, UK
i.psaras@ucl.ac.uk

ABSTRACT

The Edge of the Internet is accommodating a large number of devices. Edge devices and their associated up-link bandwidth requirements are predicted to become a major problem soon. Hence, the popularity of Edge networks data management, analysis and processing is also increasing. This paper provides a performance analysis and a development framework for Mobile Data Repositories at the Edge [10]. A suitable simulation environment was first created for scalable simulations. The simulations were further used to analyze the behaviour of the Edge Data Repositories system under different situations and in a range of configurations. The potential benefit of implementing this system at the Edge of the Internet is outlined in this paper.

CCS CONCEPTS

• **Networks** → **Network simulations**; **Network performance analysis**; **In-network processing**; **Network management**; *Network measurement*; *Mobile networks*; *Wireless local area networks*.

KEYWORDS

Edge storage, Edge Computing, Data Repositories, IoT, V2I, Service Provisioning

ACM Reference Format:

Adrian-Cristian Nicolaescu and Ioannis Psaras. 2019. Initial insights into the performance of Data repositories at the Edge, the case of Store-Process-Send. In *Proceedings of ACM EdgeSys conference (EdgeSys 2019).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

The number of devices connected to the Edge of the Internet is predicted to exponentially rise in the next few years [3]. These devices are imposing increasingly high upload rates.

Currently, the problems associated with bottlenecks are created in the up-links, stemming from users. Users require constant connection to the cloud for storage, assistive technologies, streaming or gaming services. This issue will soon be exacerbated by the need for extra bandwidth and processing services closer to the Edge. This requirement is a result of increased traffic created by connected cars, homes, other IoT implementations (smart cities)[16] and/or augmented/virtual reality (AR/VR) devices[4]. We have developed an effective and feasible solution for the issues mentioned: implimenting a store-process-send system at the Edge of the Internet.

The subject of Edge computing is becoming one of the hottest topics, as it is continuously gaining support by creating opportunities and collaborations[12]. Networks are dispersing, hence resource redistribution is needed for this transition to occur smoothly.

This paper discusses and analyzes the performance, feasibility and deployment options of Edge Data Repository networks. We aim to alleviate the strain on up-links towards the cloud, provide connectivity, local processing and storage to users. These goals require a comprehensive study of the data manipulation and management performed on static repositories, implementing a store-process-send system. The resulting system improves overall user Quality of Service (QoS) and Internet Service Provider (ISP) networks for service provision, efficiency and profit.

The next section will discuss the rising interest for analyzing the concept of Edge Data Repositories by reviewing related literature (Section 2). Then we will show the modelling of the Edge Data Repository system (Section 3). Using the results, Metrics and Trade-offs obtained from the simulation, the system's performance will then be analysed (Section 4), leading to the development of plans for Future Work (Section 5). To conclude, the performance and potential of the system will be evaluated from the results and expressed in the last section (Section 6).

2 RELATED WORKS

Several works [12, 13, 17] suggest that user, IoT and vehicular networks will overload the network with information soon. Stress will be imposed on network processing by sending information into the up-link. Furthermore, [17] also suggests the solution with computation offloading. The concept is similar to ours, as it implements Mobile-Edge Computing (MEC) nodes connected to Road-Side Units (RSUs). This idea implies higher latencies, more data management and assumes small data quantities. This does not alleviate the up-link bottlenecks generated by the unprocessed data either.

A good example of data and processing analysis at the Edge (mostly considering IoT devices) is a paper concerned with local processing and dissemination of IoT data [6]. It evaluates the prediction of IoT data locality, generation rates and optimisation of data distribution, depending on the producer/receiver node numbers, processing capacity and delays and network strain. [6] complements our findings by offering a high-level view on how data itself can influence QoS and system performance.

EdgeSys 2019, March 2019, Dresden, Germany

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

As demonstrated in [11], the Opportunistic Network Environment (ONE) simulator[1] is famous for its support and ease of simulating opportunistic networks. In our case, the ONE simulator was considered an excellent launch pad for the simulator design. The extended simulator was designed to model the ONEs created at the Edge by the Data Repositories. The original version of the simulator and its extensions were valuable resources for the development of this extension. Details regarding the extension development process and the extra functionality will be provided in Section 3.

3 DATA REPOSITORY MODEL AND THE ONE SIMULATOR

The system model of the Edge Data Repository store-process-send service was implemented and evaluated with an extension of the ONE Simulator[5] (developed specifically for this purpose)[2].

The initial, most essential part of this extension is the implementation of storage on specific (stationary) nodes as the repositories. Without this function, nodes cannot store messages outside their routing buffer.

Opportunistic routing is needed for the mobile nodes to route all messages available for transmission to the first encountered repository. Furthermore, for a more realistic view on the environment, we have also implemented parallel processing capability for file compression on the repositories. This was demonstrated in conjunction with cloud offloading, showing differences in to show the difference in data usage and up-link strain with and without different functionalities and repository nodes.

3.1 Storage

The module for message (data unit in our simulations) storage management is represented by a new class, *RepoStorage*, as one of the essential modules implemented on nodes which have storage capability. All repositories will have this ability, in order to store and send data. Depending on the scope of the implementation and of the data, this functionality should be sufficient for certain use cases, for example pedestrian multimedia caching and distribution services. In the case of our tool, this functionality was a simple addition to the system, which still greatly improves the development potential of Edge data distribution networks.

The storage management module is not only used for storing messages, but also for managing the data within storage. The module keeps both processing "queue" and non-processing storage within the limits of the total storage capacity. The java module also deletes messages when needed and keeps track of each message received, sent and deleted on each repository, for analysis.

3.2 Opportunistic Routing

Coverage and connectivity are essential in areas with limited or no connectivity due to overcrowding, lack of cell network provider coverage, slow connections due to bottlenecks upstream, or high demand for low-latency applications/processing. The aims should be completed within one hop for both latency and connectivity reasons. For this reason, opportunistic routing development was a necessity for the simulation model. The sole routing protocol used within this environment showcased and isolated the system, creating the best analysis of the worst-case environmental conditions.

The entire system is designed around opportunistic data offloading, keeping the strain on storage and processing on mobile nodes to a minimum and their experienced Quality of Service (QoS) as high as possible. These performance metrics are dictated by the ease of transfer, latency and efficiency improvements on nodes.

3.3 Message generation and classification

Messages are constantly generated by each device at the rates specified for each application associated with the groups of hosts. The application creates messages with a certain size, at a certain rate. Each message has an associated type - processing with an associated processing delay, or non-processing.

The message type is used to differentiate a message sent by a person from a message generated by an IoT system, like a car sensing system, for example. The information could be essential video or measurements to be sent to a local repository for processing.

The mobile nodes do not have storage or processing capability in the considered scenarios. Considering this, the potential for development of more complex functional algorithms of mobile Edge Data Repositories is great and will also be discussed in the *Future Work* Section.

The mobile generation rates used in this paper's evaluation of the system were based on CISCO VNI predictions [3] for 2022, several case studies ([9, 14]) and reasonable approximations. The approximations were made considering that the majority of total data traffic would be represented by vehicle communications. The average data production for each node is around 50GB per day and varies with different scenarios and type or group of data producer.

3.4 Storage, processing and offloading

The repositories have an internal data management system that stores, processes and sends messages. The management system handles messages according to their priority, which is determined by the age, type (processing or non-processing) and processing delays of the messages.

A flow chart of the message storage process is presented in Fig. 1. The process is the main part that handles the storage of messages and decides their priority during processing. A flow chart representing the constant update process is represented in Fig. 2, showing the logical process behind the compression processing, simple depletion and/or cloud offloading. Each repository has three main parts, described below.

Storage association and management - After a message is received at a repository, it is handled by two different functions: storage management system insertion (Fig. 1) and new message processing. The application triggers the processing of a message only if the message is of "processing" type. Once the message is stored, the storage insertion function updates the storage state.

Message processing and update handling - The application also updates and manages the storage every simulation cycle (0.1 simulation seconds), executing several steps on each update. The step executed before message depletion is shown in part 1 of Fig. 2, using parallel processing. Once depletion begins, step 1 is executed for each message processed and uploaded, but only on one thread.

[1] The ONE: http://akeranen.github.io/the-one/
[2] The ONE Repos: https://github.com/Chrisys93/the-one-repos

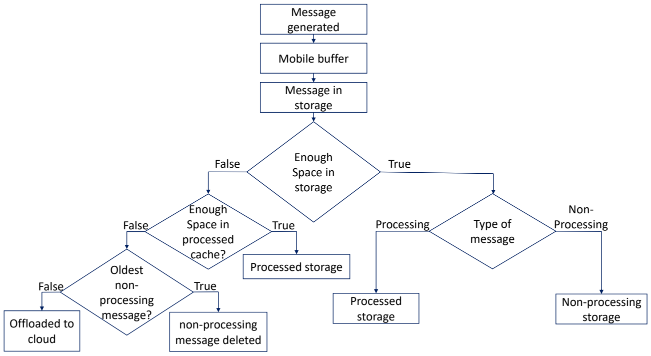

Figure 1: Flow chart of the Repository Internal Management

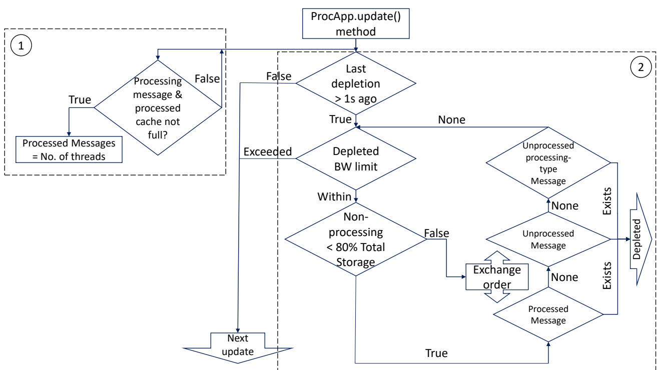

Figure 2: Flow chart of the `ProcApp.update()` method

The number of parallel processing threads is determined in the configuration file and every update checks processing availability. The number of threads per repository is influenced by the ISP's network traffic, users' needs and both their budgets (accounting for QoS and expenses). Depending on the percentage of processing or non-processing storage usage, processed message depletion is prioritised over unprocessed message depletion, by default.

Depletion of messages - The processing application also manages message depletion - see part 2 of Fig. 2. The upload speed is determined by the parameters of the simulation set in the configuration file. For every processed message uploaded, another newly added message is processed.

By configuring the upload bandwidth and the expected mobile device generation speeds, an expected Data Repository input/output bandwidth ratio trade-off is created. The optimal configuration of the repositories can then be designed, depending on the specifications of the ISP, its QoS requirements and the expected user payment plans and services expectations.

3.5 Reporting and measurements

Reporting is the project's results gathering tool; it collects data from the repository and mobile data transfers, buffering and storage, then reports results with respect to the appropriate entities. The values are produced for use in the system's further analysis and improvement. The reporting tool provides all results necessary to obtain an overall view of the system performance.

The reports show: **per repository input speed, processing message storage, non-processing message storage, processing message up-link bandwidth, non-processing message up-link bandwidth, messages deleted from mobile buffers and total repository storage occupancy**.

4 METRICS AND TRADE-OFFS

A main purpose of this system is to reduce up-link bandwidths and latencies for in-network processing. Unless otherwise stated, the simulations used to assess the metrics and trade-offs were set up as follows.

There are 80 repositories and three other different types of mobile devices: pedestrians, cars and buses. The type of interest is cars, of which there are 500 nodes. There are also 40 pedestrians and 15 buses in each simulation. The nodes create messages at these ratios (non-processing:processing): 2:0 messages per pedestrian per 5s, 0:2 messages per car per 2s with a processing delay of 0.1s and 1:2 messages per bus per 2s with a processing delay of 0.2s. All messages are 1MB in size. Each repository has a total storage capacity of 10GB, a compression ratio of 0.5, a processed messages cache storage of `total storage/(2*compression ratio)` capacity and 8 parallel processing threads. The compression ratio determines the size of processed messages compared to unprocessed messages.

All physical layer interfaces used were WLAN, thus having reduced range, yet higher bandwidth as compared to LTE.

4.1 Main Trade-offs

The main factors determining the trade-offs are: **number of users (mainly cars), processing power per repository, repository system scope, up-link bandwidths and storage capacity.**

Numbers of generating nodes supported per repository vs. investment into WLAN-LTE adaptation/ multiple-link system/ coverage;

Purpose of repository system (Massive storage vs. processing workforce vs. processed message up-link bandwidth);

Parallel processing power per repository;

Offloading vs. local service provision for (subscriber) connectivity.

Depending on the specific use case, the parameters above can be traded by the system designer to achieve optimum performance.

4.2 Processing

The processing-type messages should be processed within close proximity to the source, to use less bandwidth up-stream or for lower latency. After processing, the information may be sent to the cloud, back to the user or to another interested entity. In the case of vehicles, messages may be uploaded directly to insurance companies, associated and transport companies or vehicle manufacturers. In the case of an incident, a processed diagnostics message could also be sent to emergency services in due time.

All repositories in the simulations have processing power and provide a processing service. If the repository is overloaded with messages to be processed, some of the messages may be offloaded directly to the cloud. This produces a smaller impact than without the implementation of repositories (see Figs. 3 and 5).

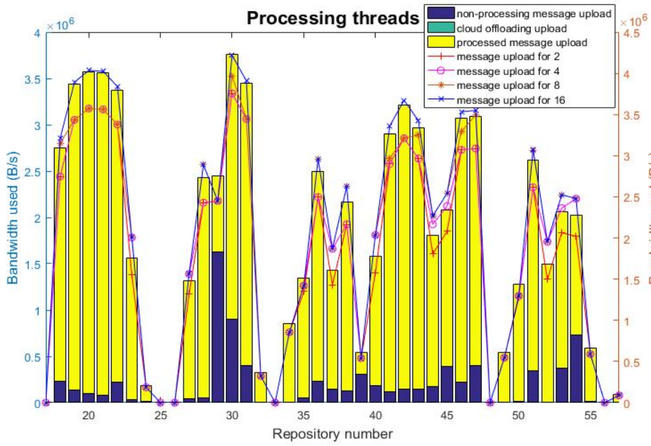

Figure 3: Average upload BW (for all repositories) in scenarios with different numbers of processing threads per repository (line graphs correspond to right hand side Y-axis).

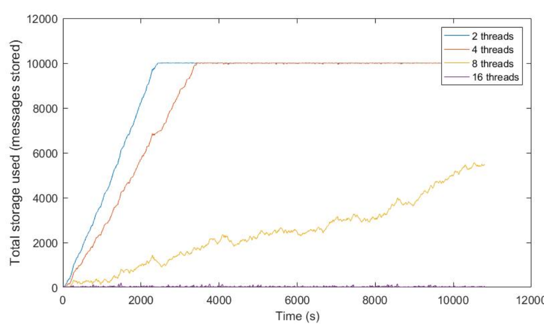

Figure 4: Storage occupancy of one of the most used (High-traffic) repositories; with top up-link bandwidth of 10 MB/s and different numbers of processing threads per repository.

The system designer should ensure that the trade-off between processing power, storage capacity and upstream offloading offered on each repository is given special attention. Per repository processing power influences efficiency, storage and information output, demonstrated in simulations with different numbers of parallel processing threads per repository (Figs. 3 and 4). Note that the input rates do not differ between the cases, as the input is totally dependent on generation rates and mobility patterns.

Processing messages overloading the repositories may become a problem. A potential solution would be using spare processing power of other repositories for processing the overloading messages. This topic will further be discussed in the *Future Work* Section.

Please note: *In the above examples, the maximum upload speed of 10 MB/s is used, as it was determined to be higher than the needed bandwidth for the most used repository.*

4.3 Up-link bandwidth allocation

The offloading of data towards its final destination is theoretical in these simulations. It is only considered to be limited by a certain bandwidth from the repository, up-stream, or back towards the Edge, set in the configuration file of the simulation. Data and function execution offloading is already practiced, but only to the cloud, implying high latencies and up-link bottlenecks, creating the necessity for retransmissions.

By prioritising different types of messages in various phases of processing, the system maintains the amount of information (messages) sent as high as possible, imposing less strain on the cloud (as shown in Fig. 3). In the case of maximum depletion rate change between simulation scenarios, the upload bandwidth remains the same for both cases. When the processing power of some repositories does not match the input bandwidth, repositories cannot overcome the strain imposed by the simulation conditions.

Please note: *In the above examples, there are 8 processing threads per repository, as it was determined to be the current average processing capability of a normal computer.*
The simulations presented are extreme cases, in which all information is forwarded to the repositories. This amount of usage and stress is irregular throughout a day.

4.4 Mobile node support capabilities

The main influences on the number of mobile nodes supported per repository are: physical layer protocols, radio coverage, capacity (bandwidth and number of connections) and processing power.

The use of LTE interfaces would be advantageous, by placing the repositories in base stations and having greater coverage. Thus, the number of mobile nodes supported by each repository could be increased. Better options would be implementing a WLAN-LTE [1] physical layer or using a mix of distributed LTE- and WLAN-enabled repositories (increased costs) and/or multi-packet receiving. Otherwise, mobile nodes drop generated messages that overload their buffer space. See Fig. 6.

Figs. 5 and 6 show results of increasing the number of mobile nodes without increasing the number of repositories. Fig. 7 shows that even though the upload bandwidths increase in the 1000-car simulation, the processing power is insufficient in high-traffic repositories. This imposes upload throttling, due to insufficient resources to store and process enough messages to satisfy the input bandwidths.

4.5 Generation vs. Storage vs. Offloading

People's expectations/perception of network speed is not as demanding/fast as device processing deadlines and cache-like storage usage. Devices use, store, process and send large amounts of data quickly. Therefore, considering services management, there may also be a need for a local services provision repository, which uploads less or no data to the cloud. In this case, the processing functionality of the repository may be used less frequently, or would not be overloaded at least. The users would mostly utilise these environments for personal storage or processing. A good counterexample for this statement is the Google Glass adaptation from [4], which is also a situation that could be improved by the other types of Edge Data Repositories.

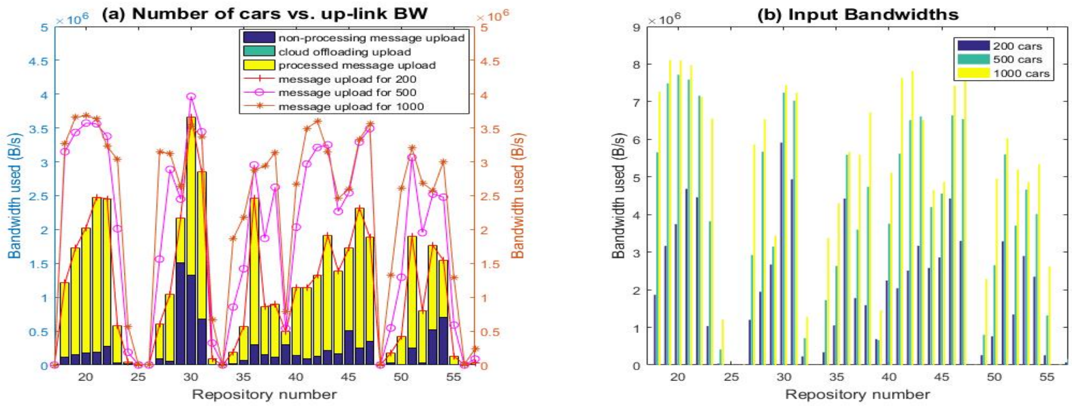

Figure 5: (a) The average upload BW used (for all repositories) in scenarios with different numbers of cars simulated, 8 processing threads and 10MB/s maximum upload BW per repository. (b) Baseline upload bandwidth per repository at input.

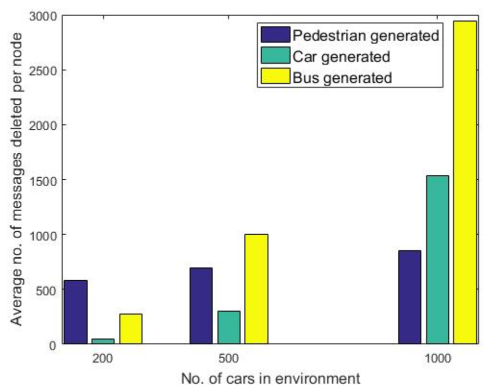

Figure 6: Average numbers of deleted messages from mobile nodes in scenarios with different numbers of cars.

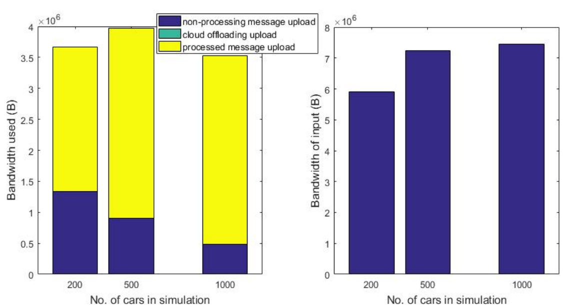

Figure 7: Average upload BW compared to average input BW on one of the most used (High-traffic) repositories; with top up-link bandwidth of 10MB/s and 8 processing threads.

4.6 Repository classification

Different trade-offs, better performance and various system configurations can be advantageous depending on the position the repositories are placed in and their configuration. An important factor to consider is that a system formed with repositories could support low up-link bandwidths or work in a stand-alone setup, for local storage, processing and/or service provision. The following classification makes a distinction between repository types and placement, considering their main functionality and performance requirements in terms of storage and processing.

***High-traffic repositories* - trade-off considered - processing power vs. offload capacity**

Higher priority is given to processing. Fig. 4 shows how different processing powers per repository can influence the strain imposed on it. Fig. 3 gives another clear representation of this type of repository (in the same context), showing the change in upload speed, depending on processing threads per repository.

For this type of repositories it is important to consider the need for processing power, according to the ISP's budget (up-link bandwidth allocation and investment) and expected device data generation and usage.

***Connectivity repositories* - trade-off considered - allocated up-link BW vs. location-generation-type of generation**

These repositories are mostly connected, and are generally able to handle slightly higher or lower traffic in an adaptive way (Fig. 8). They are not designed to deal with constant strain from either.

This type of repositories is mainly concerned with maintaining a good connection with Tier-2 ISP managed networks. Thus, the balance between maximum allocated bandwidth and the storage/processing scope of these repositories is more important for investment decisions.

***Low-traffic repositories* - trade-off considered - services offered vs. "subscriber" capability/user QoS**

By default, higher priority is given to storage, intended mostly for mobile users' need for nearby information storage. This service would be provided for a fee, considering that personal devices do not need low-latency connections to retrieve data mostly from cold online storage.

We have used an even distribution of repositories to give a general idea of how distribution influences the influx of data. The percentage of influx that each type of data represents is also shown.

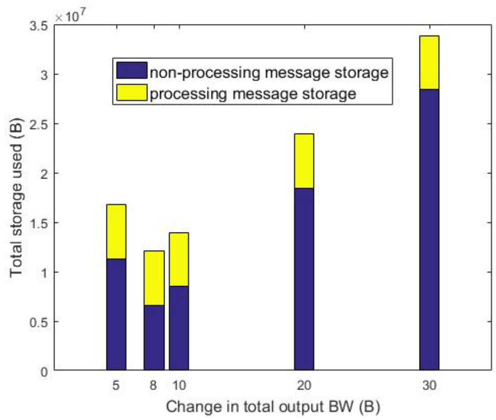

Figure 8: Storage occupancy of one of the moderately used (Connectivity) repositories; with 8 processing threads.

Depending on a specific area and its predominating traffic, the optimum distribution of each type of repository could potentially be obtained, for an effective and efficient system.

5 FUTURE WORK

There is great potential in the extension to a multi-hop routing system. This approach would require the integration of route prediction/pattern recognition into the forwarding algorithm, to account for network traffic engineering and congestion control. It was also mentioned in an earlier section that spare processing power of less used repositories could be reallocated to messages which overload high-traffic repositories. This could be implemented when a less used repository is near the overloaded one and utilised up to a certain threshold. This measure could also become an opportunity for sending the message closer to its destination.

Repository distribution and configuration depending on traffic intensity and type are important subjects in this paper. Considering developments in technology, repositories may potentially have dynamic configurations for storage, processing and offloading schemes. This would depend on the requirements of the users and maintaining ISP.

Public transport as the mobile support for repositories could also be considered for further study. By piggybacking buses, the store-process-send system performance could improve dramatically.

The work presented in [6] also has potential for implementation with the Data Repositories, with the repositories being used as "brokers". It is a higher-level assessment of data dissemination, thus fitting in with our outlook on data distribution and repository systems placement. A further extension of this integrated system could then be implemented with the help of an Information-Centric Networks (ICN)-based implementation. This would use NFaaS [8]/Computation Offloading with ICN [7] and the principles of PIoT [15] and/or A Keyword-based ICN-IoT Platform [2].

A final idea for development is a self-sustained, local store-process-send environment. It would have less or no interaction with the cloud, for security and round trip time (RTT) purposes.

6 CONCLUSIONS

The assessment we have completed in this paper should give a good idea of the possible implementations of Edge Data Repositories. We have also discussed their potential for deployment and data management in depth. It was also demonstrated that Edge Data Repositories' distribution and systems design potential in the telecommunications industry is promising.

The evaluations produced in this paper are important for the following reasons: They showcase trade-offs, evaluating which trade-offs are to be considered more important in different scenarios. The evaluations anticipate the potential improvements and integration of Data Repositories with other systems. Finally, they provide a framework and tool, for the design of systems which integrate Edge Data Repositories.

There is a wide range of developments that can be based on this system, depending on their final purpose of deployment. This collection of devices not only forms a configurable and adaptable system, but further development could enhance its valuability as a solution for the Edge of tomorrow's Internet.

REFERENCES

[1] 2017. LTE-WLAN Aggregation (LWA) : Benefits and Deployment Considerations.
[2] Onur Ascigil et al. 2017. A keyword-based ICN-IoT platform. *Proceedings of the 4th ACM Conference on Information-Centric Networking - ICN '17* September (2017), 22–28. https://doi.org/10.1145/3125719.3125733
[3] Cisco. 2018. Cisco Visual Networking Index : Forecast and Trends ,. (2018), 2017–2022.
[4] Kiryong Ha et al. 2014. Towards wearable cognitive assistance. *Proceedings of the 12th annual international conference on Mobile systems, applications, and services - MobiSys '14* (2014), 68–81. https://doi.org/10.1145/2594368.2594383 arXiv:arXiv:1011.1669v3
[5] Ari Keränen et al. 2009. The ONE simulator for DTN protocol evaluation. *Proceedings of the Second International ICST Conference on Simulation Tools and Techniques* (2009). https://doi.org/10.4108/ICST.SIMUTOOLS2009.5674
[6] Shweta Khare et al. [n. d.]. Scalable Edge Computing for Low Latency Data Dissemination in Topic-based Publish/Subscribe. ([n. d.]). https://doi.org/10.1109/SEC.2018.00023
[7] Michal Krol et al. 2018. Computation offloading with ICN. *Proceedings of the 5th ACM Conference on Information-Centric Networking - ICN '18* (2018).
[8] Michał Król and Ioannis Psaras. 2017. NFaaS. *Proceedings of the 4th ACM Conference on Information-Centric Networking - ICN '17* 11 (2017), 134–144. https://doi.org/10.1145/3125719.3125727
[9] Eton Manor. 2012. CASE STUDY Wi-fi access at London 2012 The London 2012 Olympic Park wi-fi network delivered 100 per cent availability throughout the Games. (2012).
[10] Ioannis Psaras et al. 2018. Mobile Data Repositories at the Edge. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*. USENIX Association, Boston, MA. https://www.usenix.org/conference/hotedge18/presentation/psaras
[11] Sujoy Saha et al. 2014. e-ONE:Enhanced ONE for Simulating Challenged Network Scenarios. *Journal of Networks* 9, 12 (2014), 3290–3304. https://doi.org/10.4304/jnw.9.12.3290-3304
[12] M. Satyanarayanan. 2017. The Emergence of Edge Computing. *Computer* 50, 1 (Jan. 2017), 30–39. https://doi.org/10.1109/MC.2017.9
[13] Eve M. Schooler et al. 2017. An Architectural Vision for a Data-Centric IoT: Rethinking Things, Trust and Clouds. *Proceedings - International Conference on Distributed Computing Systems* (2017), 1717–1728. https://doi.org/10.1109/ICDCS.2017.243
[14] TfL. 2000. Case study: Transport for Londom. (2000), 1–2.
[15] Yuhang Ye et al. 2016. PIoT: Programmable IoT using Information Centric Networking. *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium* Noms (2016), 825–829. https://doi.org/10.1109/NOMS.2016.7502908
[16] Andrea Zanella et al. 2014. Internet of Things for Smart Cities. *IEEE Internet of Things Journal* 1, 1 (2014), 22–32. https://doi.org/10.1109/JIOT.2014.2306328
[17] Ke Zhang et al. 2017. Mobile-Edge Computing for Vehicular Networks. *IEEE Vehicular Technology Magazine* 12, June (2017), 2–10. https://doi.org/10.1109/MVT.2017.2668838