Mobile Data Repositories at the Edge

Ioannis Psaras, Onur Ascigil,
Sergi Reñé, George Pavlou
UCL, UK
{*i.psaras,o.ascigil,s.rene,g.pavlou*}*@ucl.ac.uk*

Alex Afanasyev
FIU, USA
aa@cs.fiu.edu

Lixia Zhang
UCLA, USA
lixia@cs.ucla.edu

Abstract

In a future IoT-dominated environment the majority of data will be produced at the edge, which may be moved to the network core. We argue that this reverses today's *"core-to-edge"* data flow to an *"edge-to-core"* model and puts severe stress on edge access/cellular links. In this paper, we propose a data-centric communication approach which treats storage and wire the same as far as their ability to supply the requested data is concerned. Given that storage is cheaper to provide and scales better than wires, we argue for enhancing network connectivity with local storage services (e.g., in WiFi Access Points, or similar) at the edge of the network. Such local storage services can be used to buffer IoT and user-generated data at the edge, prior to data-cloud synchronization.

1 Introduction

It is being continuously claimed that in the very near future the majority of mobile devices (from wearables to every sensor on automotive) will be connected to the Internet. What is not often discussed is that these (mostly) mobile devices and sensors will be constantly producing enormous amounts of data. For instance, the amount of data produced by Internet of Things (IoT) in the global scale is forecasted to exceed 1.6 zettabytes by 2020 [1, 2]. In the case of surveillance cameras or car sensors, a constant stream of data is produced from each device. Also, user-generated content (e.g., real-time video streaming from user devices to social network applications) will stress the network further and might cause a big *data explosion* that the access network is not able to absorb.

We are therefore starting to see a reverse data-flow, according to which, data are produced at the edge and flow toward the core of the network to be stored or processed. This is in stark contrast to today's model, where we largely assume that data resides at the core of the network (in some data-centre or CDN server farm) and flows to users' devices connected at network edges.

In an environment where the demands for asynchronous data services (as opposed to synchronous telephony services) dominate mobile communications, plain connectivity between two end points—the service provided today by (Wireless) ISPs (wISPs)—is not the end, but only the means to the end. The end goal in data-intensive environments is access to information contained in data.

We, therefore, argue for complementing network connectivity with local *edge data repositories* that provide storage services (e.g., at WiFi Access Points) at the edge of the network (see Figure 1). According to our vision, storage allowance is provided by the wISP as part of the monthly contract, together with the voice and data plan. Similar to the connectivity service where one gets connected at any place, this storage allowance should also "move" with the user, i.e., enabling users to upload their produced data to local storage resources (e.g., in base stations, wifi access points) as they move along. The stored data may just wait to be eventually synchronized with the cloud, or get processed by edge services [3].

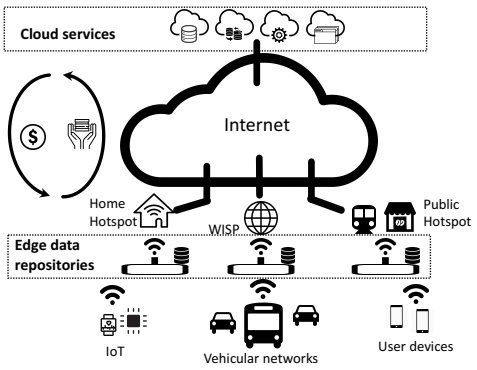

Figure 1: Edge data repositories

Our vision necessitates a data- or information-centric mode of network operation. First of all, a data-centric model defines the basic communication unit to be the data as opposed to a *channel* providing connectivity only (as in today's TCP/IP networks). This makes asynchronous access to data trivial, because wire connectivity and storage can be treated the same in their ability to supply data. Second, we envision the data to be packaged together with other necessary attributes, e.g., signature, so that data can be verified for authenticity and integrity without the overhead of establishing secure channels.

The above-mentioned properties—being able to retrieve data from anywhere and authenticating them independent of channels—can be natively supported by an Information-Centric Networking (ICN) paradigm. Therefore, we argue for network service providers to adopt an ICN-based network layer design such as the Named Data Networking (NDN) architecture [4] to provide data-centric communication service that incorporates edge data repositories.

In ICN, data is explicitly named and signed. Naming data directly enables the network to route based on names, rather than on end-host (IP) addresses. This is a important feature in an environment where data producers are highly mobile which renders addresses meaningless. Explicitly signed data enables *authenticate data* directly. In contrast to a host-centric environment, where users need to trust the containers that host the data, directly authenticated data simplifies service management and makes data re-usability between different users and applications straightforward.

Vision Summary: Edge storage in WiFi APs forms an ambient edge-cloud where data can be temporarily stored, processed and/or synchronized with the back-end cloud, *only when necessary and following the best strategy depending on the application requirements*. That said, edge network functions can better control when to upload the data, in turn being able to shape the upload stream (i.e., the volume of upload traffic) according to network conditions, as opposed to the network being merely a path to the cloud. An ICN network substrate serves well the purpose of such a vision through name-based routing and forwarding, securing data directly, and support for user mobility.

2 Background and Motivation

2.1 Illustrative Example

Consider the biker's helmet-camera or the car's camera that is constantly recording everything as the vehicle is moving around. These data is primarily of interest to insurance companies in case of a collision/accident, but of little use otherwise. The question then becomes, what would be the best way to handle all these data? According to the current Internet infrastructure, there are a few options of what one can do with such data produced at the edge of the network: i) assume that the helmet or car can apply image-processing functions onboard, and data are transmitted to the insurance company only when a collision is detected after processing the data, or ii) transmit all data through the cellular network to the backend-cloud for storage and processing. The first option would require significant processing power on the helmet camera or the car itself, which would in turn increase significantly the price of these devices. The second option presents several challenges:

- **Cell network would be brought to its knees.** Despite increasing capacities of cell towers and last-mile links, it is highly unlikely that the mobile backhaul network will reach the capacity of broadband connections any time soon. In other words, cellular networks may not have the capacity to transfer all the edge created data.
- **The current ISP-relationship business model would be turned on its head.** Today's edge/Eyeball ISPs business is traffic download. When orders of magnitude more upload traffic are produced at the edge, ISPs will have to upgrade their network accordingly. This may pose a tremendous challenge, as it could be a show-stopper for IoT as a whole: the increased costs for edge/eyeball ISPs would push them to increase their charges/subscription costs to end users and IoT application providers, making network usage more expensive.
- **Mobility is an unsolved problem in IP networks** [5, 6]. User mobility (both client- and producer-/server-mobility) has traditionally presented a challenge for the IP network. When users move and therefore disconnect from their point of connectivity, the session is temporarily broken until the user connects to the next access point. The session-based, synchronous mode of communication supported by IP is unfit for purpose in case of asynchronous data services needed by edge-produced data.

2.2 Limitations of IP Architecture

According to today's TCP/IP communication model, upon production, data are transferred from a mobile device to a backend cloud over the cellular network. As argued above, this is not sustainable given the enormous capacity requirements of IoT data being generated. With edge-data repositories, data are offloaded there first and fetched to cloud servers as needed. However, to support edge repositories using the current TCP/IP stack, each data object would have to be mapped to the IP address of the corresponding edge repository. The IP address of the edge-data repository can be communicated to the backend part of the application it belongs. Any subsequent request for this object should be redirected from the backend cloud to the edge data repository.

Such an implementation may look straightforward in case of relatively static data generation for a single application, i.e., a whole object is generated and offloaded with no end user mobility involved and synchronized to a single cloud service provider; the case becomes more complex when the end user/IoT device is moving and

connecting to different edge repositories as it goes. ICN provides inherent mobility support, while the IP does not, i.e, there is no need to renew the local address and establish a new session adding delay that can limit data transfers when mobility is high, or to keep alive old sessions using suboptimal solutions (e.g., [7, 8, 6]) adding complexity and overhead.

2.3 Benefits of Edge Data Repositories

Edge Data Repositories allow the producers to simply offload their data to the network and let the network manage the storage and access to data. All this is done without requiring the data producers to establish a channel with an endpoint (e.g., cloud server) and handle the transfer of data as in the current connectivity-based model.

The ability to store data at the edges can lead to cost saving opportunities in terms of bandwidth usage. Future APs with computation capabilities can process the data locally. Therefore: *i)* data can be pre-processed locally at the network edge to significantly reduce the amount of data sent upstream, and *ii)* the transfer of data to the cloud can be scheduled over longer period of time to reduce the upstream traffic rate, and thus transit costs. In cases where data are only relevant to local consumers, those users can be redirected to the local repository within the domain without crossing expensive inter-domain links.

2.4 Related Work

Related to our work is the concept of a Reverse-CDN by Schooler et al. [9]. This work proposes an architectural design vision to use both Fog Computing and Information-Centric Networking (ICN) combined in order to process IoT data locally at the edges. Also, Satyanarayanan et al. [10] proposed edge computing to process IoT data locally to improve real-time video analytics. Earlier data-centric solutions also exist, such as Haggle [11], but used mainly for enabling delay-tolerant and device-to-device communications and not for making data available globally using data repositories.

The concept of distributed edge repository storage is similar in rationale to decentralized content-addressed storage systems, such as IPFS [12] or Cloudpath [13]. However, these approaches lack data-centricity and suffer from the drawbacks of host-to-host communications. Specifically, off-loading or retrieval of data requires establishing a channel with an endpoint, which can be difficult especially when the hosts are mobile. ICN-based approach that we advocate in this paper, on the other hand, enables any mobile user to off-load or retrieve data without creating a channel and makes it possible to secure the data itself without a mandatory requirement to secure a channel.

3 Mobile Edge Data Repositories: Technical Challenges and Directions

3.1 System Overview

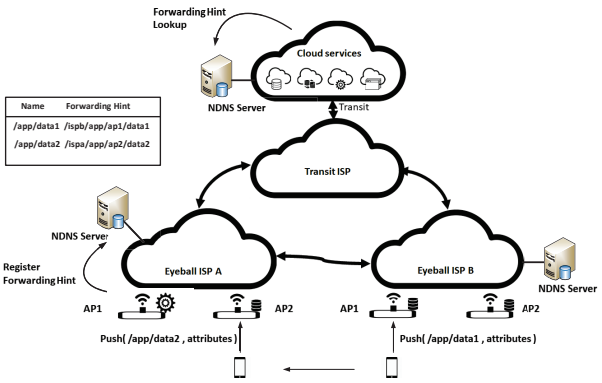

Figure 2: Data-centric communications using edge data repositories

In our edge data repository environment illustrated in Figure 2, data produced by the mobile device are immediately pushed to or pulled by edge APs as discussed in Section 3.3). APs act as the stable in-network rendezvous points for the consumers and producers, decoupling the act of sending packets by the producers from the act of receiving packets by the consumers. Furthermore, given that the data are named at the granularity of packets (i.e., *chunks*) and are not bound to a connection between two endpoints, the network simply performs *name resolution* to forward request packets towards the AP which stored the intended data packets. Having data stored in the distributed edge repositories requires the network to implement data resolution mechanisms in order to provide access to data. In order to do so, the APs must inform the name resolution mechanism with updates on the whereabouts of the stored data chunks. Applications with real-time access requirements to data and mobile producers make the job of the resolution mechanism more challenging as we describe next.

3.2 Data Resolution & Producer Mobility

Once data are stored at an edge repository, the network must enable access to the data. The *Named Data Networking (NDN) architecture* [4] uses data names directly in packet routing and forwarding. Routing on data names require a name resolution process. One example is an *in-network name resolution*, where a routing protocol updates the forwarding state of the nodes so that they can

collectively map names to producers as a result of the routing protocol convergence.

In the proposed system, once an edge repository receives named data chunks, it assumes the role of a producer of the named data chunks. In the case of a moving producer, the producer spreads its data chunks to different AP locations while moving. Producer mobility together with real-time access requirements to named data chunks presents a challenge for the network, because fast name resolution is required to enable immediate access to data, e.g., for real-time applications such as a producer streaming video and consumers watching, the requests for named video chunks need to be resolved and forwarded to the correct (up-to-date) repository location immediately without delay.

Given it is infeasible to rely on network routing to handle node mobility, we propose an indirection-based name resolution instead. A possible candidate for indirection-based name resolution is NDNS [14]: APs inform an authoritative NDNS server of a forwarding hint (i.e., a directional hint for NDN forwarders on where the requested data can be found) [15] for data chunks they have received locally with a registration operation as shown in Fig. 2. In this case, data consumers will look up NDNS first to learn the forwarding hint for their desired data, then send request packets indicating the name of the desired data chunks together with the forwarding hint name. APs can act as authoritative NDNS servers for mobile producer namespaces. As a further optimization, the APs can replicate the data at both the current location and the previous (i.e. current default) location of the mobile, until the name resolution converges and consumers start sending requests to the new AP.

3.3 Push vs. Pull Communications

One of the main features of NDN (and ICN in general) is the *network-layer pull-based communication model*. Pull-based communications offer several advantages over push-based communications, such as built-in multicast delivery, receiver-oriented congestion control [16], and native support for client mobility [17].

In our case, pull-based communication is achieved by instantiating lightweight versions of applications inside the edge repositories, e.g., lightweight version of a dropbox-like application to store personal videos. The only task of the instances would be to pull the producer data and store them into the edge repository. This can be initiated by a producer sending a "request for pull" message to an application instance, which in turn triggers a pull request to be sent back to the producer by the instance. The deployment of virtual application instances within an edge network can be realized through an edge computing infrastructure [18, 3].

3.4 ISP Relationships

In the current customer-provider business relationship model, customer ISPs typically commit to a certain rate of traffic (in Mbps) and depending on the committed rate, they are charged per Mbps for the 95th percentile rate, i.e., excluding the bursts. With increasing edge data production, the volume of data that needs to be sent to higher-tier ISPs, and the rate of requests for the stored data will affect the transit costs of eyeball ISPs. This is likely to cause a "tussle" [19] between last-mile networks and data producing applications in a similar way to the on-going tussle between overlay (i.e., peer-to-peer) routing applications and ISPs due to violation of the ISP routing policies by the overlay traffic.

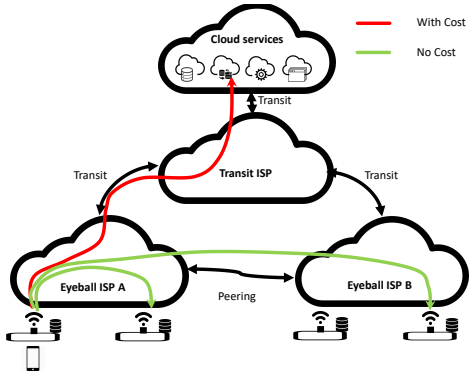

Figure 3: Data consumption scenarios

In general, data might be requested: i) locally from within the same domain, in which case there is no transit cost, ii) from peering domains, order to achieve successful deployment of edge data repositories in the access network, an important investment is required by wISPs, that need to upgrade existing or deploy new hardware. That said, wISPs are incentivised to use edge repositories as they can reduce their transit costs (similarly to [20]), in order to reduce traffic peaks and avoid being overcharged by transit ISPs.

4 Data Management

In the data-centric communication model, the name and content of each data object is cryptographically bound together, enabling the verification of authenticity and integrity of the object. Furthermore, we consider certain data management related attributes to inform the the network of the *intent* [21] of the applications in terms of how their data is to be treated. We envision that such attributes can potentially be encoded through naming—effectively bringing application semantics into network-layer forwarding. For instance, if the data is to be processed locally, the hierarchical name of the data can be prefixed with `/exec`, or if the data is to be eventually

stored or processed at a remote cloud service within a certain deadline, then a deadline may be encoded in the name. These attributes can be expressed with a list of tags or keywords [22, 23] as part of a separate component of the data names. In this section, we first discuss possible attributes of data in Section 4.1 and then describe possible data management strategies that can make use of the management attributes in Section 4.2.

4.1 Data Attributes

There is a set of data attributes that are of interest, when a network manages storage of and access to application data. The data attributes listed below relates to security properties of data objects, semantics of the application(s) that consume them and producer preferences.

- *A persistent name:* which does not change with mobility. This name may also be used by the network to locate, replicate, cache and access the data.
- *Verification information:* necessary to confirm the authenticity and integrity of data object. This may include a signature computed over the data object, instructions to verify the signature (i.e., name or location of a certificate to verify the signing key), and so on.

The rest of the attributes are optional:

- *Destination:* for data objects that require transfer to a particular endpoint, e.g., for storage, or computation, would be required to provide a locator or a name associated with the destination endpoint.
- *Shelf-life:* indicates an expiration time, after which the data may be safely discarded.
- *Access scope and urgency:* indicates the expected origin of requests for a data object, which may be strictly local, strictly global, or a mix of both. Also, the access to data can be immediate or delayed. In the case of delayed access, a deadline can be provided (see below).
- *Deadline:* For data objects that require certain time-sensitive actions such as access, computation or relaying to a destination, a deadline may be specified.

These and possibly other attributes can be desirable for the edge-networks to manage data. Next, we describe data management strategies for edge-networks.

4.2 Data Management Strategies

A data management strategy dictates how an eyeball network coordinates the management of edge data stored at a repository. Below are a list of possible strategies:

- *Proactive:* In this strategy, the local repositories transfer the incoming data proactively to the intended destination (e.g., cloud storage) immediately at the rate permitted by the outbound capacity of its link to the domain's upstream provider. This scenario uses the storage at a local repository only in the case when upstream link capacity is below the rate at which data is produced.

- *Reactive:* In this strategy, the repository registers the name of the data to the name resolution system in order to enable access and notifies the destination of the data. In case of NDNS resolution, the registration includes a forwarding hint to reach the repository. This makes it possible for consumers or cloud servers to pull the data from the repository when necessary. This way, the edge repositories handle data transfers in a "lazy" manner, i.e., transfers data only when necessary.

- *Data-specific:* In this strategy, the repositories make use of data attributes such as scope, shelf-life and deadline to determine actions specific to each data object. For example, the edge repositories can follow the reactive strategy and store a specific data locally in case the data has limited shelf-life or has only local access scope. Alternatively, the repositories can monitor access to data objects and proactively transfer them to their intended destinations in case of heavy global access.

5 Conclusions and Future Work

The imminent data explosion at network edge calls for new architectural designs. Local storage and processing at the edge of the network provide an elegant solution, according to which data are temporarily stored close to the source of data. Depending on application requirements, data is locally processed (if needed) and transferred to more permanent storage when network conditions allow.

In this paper we have presented an information-centric approach to edge-produced data, built on top of the Named Data Networking architecture. We have proposed several potential data management strategies to handle data stored at the edge, as well as producer mobility. Our design allows for extensions to incorporate edge processing—an integral part of our vision, which we plan to address in our future research.

Acknowledgment

This work is partially supported by the EC H2020 ICN2020 project (GA no. 723014) and EPSRC INSP fellowship (EP/M003787/1), and by US National Science Foundation under award CNS-1719403.

References

[1] D Evans. The Internet of Things: How the next evolution of the Internet is changing everything. In *Cisco Internet Business Solutions Group (IBSG)*, volume 1, pages 1–11, 01 2011.

[2] D. Shey A. Markkanen. Edge analytics in IoT. 04 2015.

[3] Michal Król and Ioannis Psaras. NFaaS: Named Function as a Service. In *Proceedings of the 4th ACM Conference on Information-Centric Networking (ICN)*, 2017.

[4] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3):66–73, 2014.

[5] C. Perkins. Ip mobility support for ipv4, August 2002. RFC3344.

[6] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil. Proxy mobile ipv6, August 2008. RFC5213.

[7] C. Perkins. Ip mobility support for ipv4, revised, November 2010. RFC5944.

[8] J. Kempf and R. Koodli. Distributing a symmetric fast mobile ipv6 (fmipv6) handover key using secure neighbor discovery (send), June 2008. RFC5269.

[9] E. M. Schooler, D. Zage, J. Sedayao, H. Moustafa, A. Brown, and M. Ambrosin. An architectural vision for a data-centric IoT: Rethinking Things, Trust and Clouds. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 1717–1728, June 2017.

[10] M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, and B. Amos. Edge analytics in the internet of things. *IEEE Pervasive Computing*, 14(2):24–31, Apr 2015.

[11] Jing Su, James Scott, Pan Hui, Jon Crowcroft, Eyal De Lara, Christophe Diot, Ashvin Goel, Meng How Lim, and Eben Upton. Haggle: Seamless networking for mobile applications. In *Proceedings of the 9th International Conference on Ubiquitous Computing*, UbiComp '07, pages 391–408, Berlin, Heidelberg, 2007. Springer-Verlag.

[12] Juan Benet. IPFS-content addressed, versioned, P2P file system. *arXiv preprint arXiv:1407.3561*, 2014.

[13] Seyed Hossein Mortazavi, Mohammad Salehe, Carolina Simoes Gomes, Caleb Phillips, and Eyal de Lara. Cloudpath: A Multi-tier Cloud computing framework. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, SEC '17, pages 20:1–20:13, New York, NY, USA, 2017. ACM.

[14] A. Afanasyev, X. Jiang, Y. Yu, J. Tan, Y. Xia, A. Mankin, and L. Zhang. NDNS: A DNS-like name service for NDN. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9, July 2017.

[15] Alexander Afanasyev, Cheng Yi, Lan Wang, Beichuan Zhang, and Lixia Zhang. SNAMP: Secure namespace mapping to scale NDN forwarding. In *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, pages 281–286. IEEE, 2015.

[16] Klaus Schneider, Cheng Yi, Beichuan Zhang, and Lixia Zhang. A practical congestion control scheme for Named Data Networking. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, ACM-ICN '16, pages 21–30, New York, NY, USA, 2016. ACM.

[17] Yu Zhang, Alexander Afanasyev, and Lixia Zhang. A survey of mobility support in Named Data Networking. In *Proceedings of the third Workshop on Name-Oriented Mobility: Architecture, Algorithms and Applications (NOM2016)*, April 2016.

[18] Pedro Garcia Lopez, Alberto Montresor, Dick Epema, Anwitaman Datta, Teruo Higashino, Adriana Iamnitchi, Marinho Barcellos, Pascal Felber, and Etienne Riviere. Edge-centric computing: Vision and challenges. *ACM SIGCOMM Computer Communication Review*, 45(5):37–42, 2015.

[19] David D Clark, John Wroclawski, Karen R Sollins, and Robert Braden. Tussle in cyberspace: defining tomorrow's Internet. In *ACM SIGCOMM Computer Communication Review*, volume 32, pages 347–356. ACM, 2002.

[20] Nikolaos Laoutaris, Georgios Smaragdakis, Pablo Rodriguez, and Ravi Sundaram. Delay tolerant bulk data transfers on the internet. In *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '09, pages 229–238, New York, NY, USA, 2009. ACM.

[21] Yehia Elkhatib, Geoff Coulson, and Gareth Tyson. Charting an intent driven network. In *Network and Service Management (CNSM), 2017 13th International Conference on*, pages 1–5. IEEE, 2017.

[22] M Ascigil, Sergi Rene, George Xylomenos, Ioannis Psaras, and George Pavlou. A keyword-based ICN-IoT platform. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*, pages 22–28. ACM, 2017.

[23] I. Psaras, S. Rene, K. V. Katsaros, V. Sourlas, G. Pavlou, N. Bezirgiannidis, S. Diamantopoulos, I. Komnios, and V. Tsaoussidis. Keyword-based mobile application sharing. In *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture*, MobiArch '16, pages 1–6, New York, NY, USA, 2016. ACM.