Optimal Placement Algorithm (OPA) for IoT over ICN

Hatem Khedher*, Hossam Afifi*, and Hassnaa Moustafa§
*CNRS Samovar Lab, Institut Mines-Telecom, Telecom SudParis, Saclay, France.
Email: {hatem.ibn_khedher,hossam.afifi}@telecom-sudparis.eu
§Intel Labs, Intel Corp. Hillsboro, OR, USA
Email: hassnaa.moustafa@intel.com

Abstract—**Information Centric Networks (ICN) is very promising for Internet of Things (IoT) deployment, where the data-centric approach is useful in reducing the data retrieval latency as well as the network traffic for IoT services. Also, the in-network caching capabilities in ICN limits the massive data access to the data producers and so relaxes the need of continuous connectivity E2E connectivity between data producers and data consumers (this helps power efficiency as IoT devices can enter in sleep mode when they are not transmitting data). In this paper, we present an ICN-IoT architecture in which ICN nodes provide IoT gateways capabilities and ICN caching functions. Optimal Placement Algorithm (OPA) is proposed to choose the optimal placement location for ICN nodes. We evaluated OPA with respect to several performance metrics and the obtained results show improvement in services consumption latency and network load. Furthermore, we propose a caching strategy that shows to stabilize the network load despite any increase in the number of consumer interests.**

I. Introduction

Information-Centric Networking (ICN) is a candidate architecture that eases the deployment of massive IoT. IoT over ICN can provide a multi-point to multi-point communication model that helps achieving multi-source data retrieval with less overhead. Additionally, ICN naming provides flexible support to several services over the same IoT network, and ICN in-network caching at intermediate nodes allows IoT content retrieval with high responsiveness.

ICN Research Group (ICNRG) within the Internet Research Task Force (IRTF) highlighted IoT among the major ICN scenarios [1]. ICNRG overall requirements for IoT to leverage ICN are addressed in [2] and [3]. However, introducing ICN in IoT needs studies for optimal placement of ICN nodes while vouching the security of these intermediate ICN/IoT containers.

Despite the above mentioned IRTF efforts, there are not clear contributions on IoT over ICN infrastructure optimizing ICN nodes placement. Consequently, in this paper we present a maximum flow min cut algorithm (the Optimal Placement Algorithm) for ICN/IoT nodes, considering heterogeneous performance parameters not only covering network issues but also limited system resources.

As network optimization deals with very large graph topologies, classical exact optimization techniques do not fit our problem. Instead, we use the well known Gomory-Hu (G-H) method [4] to detect the network bottlenecks through the maximum-flow minimum-cut theorem [5] and to find different tree levels for potential upgrade in ICN/IoT nodes and gateways. G-H is an optimal network flow algorithm that compacts the network graph structure using cuts to retain only feasible candidate topology and consequently lead to a smaller scale ICN/IoT placement problem. Although each ICN/IoT node is secured since its production, we present some security considerations to emphasize the reduction in network security cost while enabling ICN for IoT with different security levels.

The rest of this paper is organized as follows: Section II highlights ICN and its operation in IoT and the related work. Section III presents our Optimal Placement Algorithm, (OPA) for ICN/IoT network architecture. Section IV discusses some security considerations and Section V evaluates the algorithm performance over large graph topology. In Section VI, we compare OPA in ICN/IoT with a traditional IoT network deployment. We conclude the paper and presents our future work in Section VII.

II. ICN in IoT and Related Work

A. Information Centric Networking

ICN names the content rather than the host in the networking level. Different ICN architectures are proposed such as: network of information (NetInf) [6], named data networking (NDN) [7], content centric networking (CCN) [8], data oriented network (DONA), and publish and subscribe information protocol (PSIRP) [9]. Most of these information-centric network architectures are implemented on top of TCP/UDP/IP/P2P layer. All of them, are originally inspired from the early work of van Jacobson [10] who introduced the baselines and the fundamental features of the ICN architecture (e.g., node model, naming, routing, transport, caching, etc.) and the strategy layer for the adaptive forwarding [11]. In ICN, names are hierarchical and similar to URLs. Name Resolution Service (NRS) and data routing procedures are in general integrated or coupled. The exchanged messages between consumers ”data requestors” and producers ”data original providers” are respectively in the form of *interest* packets and *data* packets. Indeed, end-users express only $what$ they want (content name), and they let the ICN network respond to the $where$ and how the content will be retrieved. Therefore, an appropriate ICN topology is an optimization per se. ICN network consists of consumers that request the content, producers that originally provide/publish these content, and ICN routers that cache/treat the content. The content router (CR) in ICN has three main data structures as follows: $i)$ Forwarding Information Base (FIB) table: it binds the content name to the next hop as in IP layer that binds the IP prefix to the destination, $ii)$ Pending Interest Table (PIT): it binds the content name of the unsatisfied requests to the requesting face, and $iii)$ Content Store (CS) table: it binds the content name to the data per se.

Caching in ICN usually implies the on-path caching and CSs use by default the Least Recently Used (LRU) replacement policy. Off-path is also supported by redirecting user interest to a Content Delivery Network (CDN) (as an example) and not to the source/publisher of the content. ICN needs new mobility management solutions different from host-centric approaches. The $Kite$ model [12] proposed by UCLA presents a novel solution for data producer mobility, leveraging the state of PIT table on each ICN router to reach Mobile Nodes (MNs). $Kite$ supports different mobile application scenarios such as push, pull, share and upload. Firstly, a correspondent node contacts an anchor node, and each time it requests published data, interests follow the PIT table to contact the anchor node, and then follow traced interests to reach the mobile node. The approach is relevant and requires interest packets targeting a mobile producer to always pass by anchor (which can present a single point of failure). Moreover, authors design how to support the data producer mobility (publishers) in ICN network, through letting the consumers fetch the produced data easily after producer mobility. A survey of different solutions for producer mobility is also found in [13].

B. *ICN in IoT: Related Work*

Only few work consider caching in IoT systems ([14], [15], [16]). The work in [14], evaluates the performance of content retrieval from different consumers with standard NDN in-network caching, however, the cache size of resource-constrained nodes (used in the experiments) is 1 Kbyte and information is ephemeral (short-lived, transient). The work in [15] analyzes the impact of IoT information freshness over NDN caching through using a consumer driven freshness approach besides the freshness parameter included in Data packets (establishing how many seconds the content can be valid in the CS). This improves the accuracy of the data received by consumers. The challenge here is how this caching approach sustains in the presence of big number of consumers with different freshness requirements. The work in [16] presents a first study on caching IoT content in Internet wired content routers (electrically powered static routers), proposing a distributed probabilistic caching algorithm where routers dynamically update their caching probability by considering their hop distances to the source and the consumers and the data freshness (i.e. the closer the caching location is to the source, the fresher the retrieved data packet is). This approach may not be always valid and depends on the type of data, sensor/IoT type emission rate. Unlike [16], wireless NDN-IoT multi-hop network composed of (mobile) resource constrained nodes is considered in [17]. A probabilistic caching strategy is proposed that considers the data freshness and the potential constrained capabilities of devices (mainly energy level and storage capacity).

III. OPA: Optimal Placement Algorithm for ICN/IoT Nodes

ICN does not specify how nodes can be deployed in a large scale network. One could simply say that it will be implemented everywhere in the underlying topology, but this is not efficient and can induce high cost (not all the network nodes could host an efficient caching or intermediate treatment service). Our algorithm, (Optimal Placement Algorithm, OPA), given a network infrastructure, will solve this deployment problem and will give ICN instantiation graph for new ICN positions. It takes as input, the global network topology consisting of :

- IoT group producer nodes ”sensors, cameras, etc.”
- IoT gateways ”aggregation hubs, routers” corresponding to all network elements including larger Internet.
- and consumers ”applications, servers in data centers”

The proposed algorithm finds the optimal deployment strategy for ICN/IoT nodes functionalities based on the following parameters:

- The required consumer end to end response time.
- The node system performance. It is the system overhead (memory and CPU resource) after deploying ICN functionality in the candidate nodes.
- migration cost (network optimization), which represents the total cost of moving ICN/IoT functionality in terms of network bandwidth.

Hereafter, we first state the system hypotheses and then present the placement models based on exact Integer Linear Programming (ILP) and heuristic graph theory optimization.

A. *The placement algorithm*

We consider two main hypothesis in our approach:

- Consumer groups (clients) that have no direct connectivity to the IoT devices (they connect to the IoT devices through a gateway).
- Refreshing periodicity between IoT-gateways and devices (to collect IoT data) and that is smaller than the OPA evaluation periodicity.

Table I defines the main system/network parameters and decision variables.

Exact ILP Solution: The general formulation of the exact algorithm is as follows:

$$\min \sum_{s\in S}\sum_{f\in F} x_f^s \times p_f^s \quad (1)$$

$Subject\ to$

$$\forall s \in S : y_{v,f}^s \leq x_f^s \quad (2)$$

$$\forall v \in V \mid d_v^f \neq 0 : \sum_{s\in S} y_{v,f}^s = 1 \quad (3)$$

TABLE I: Mathematical Notation

Parameters	***Definition***
V	Data consumers (cloud data center or edge loud)
S	The set of server nodes (Data producers)
D^s	Maximum network capacity of the server $s \in S$
F	The set of ICN/IoT nodes or containers
f_{size}	ICN/IoT container's size in terms of memory ($f \in F$)
C^s	Maximum memory capacity of the server s
$L_{i,j}$	Link capacity between two nodes i and j (from i to j)
d_v^f	The set of consumer group's interests
p_f^s	The placement cost of f on s
Decision variables	***Definition***
x_f^s	Placement binary variable which indicates that the ICN/IoT ($f \in F$) should be placed on the (optimal) server $s \in S$
$y_{v,f}^s$	Mapping binary variable which indicates that consumer group ($v \in V$) needs an ICN/IoT container ($f \in F$) and ICN/IoT is placed on the server $s \in S$
$z_{i,j}^{v,f}$	Flow balance binary variable which indicates whether the link (i, j) is used for sending IoT data f to v

$$\forall s \in S : \sum_{v \in V} \sum_{f \in F} y_{v,f}^s \times d_v^f \leq D^s \tag{4}$$

$$\forall s \in S : \sum_{f \in F} x_f^s \times f_{size} \leq C^s \tag{5}$$

$$\sum_j z_{i,j}^{v,f} - \sum_j z_{j,i}^{v,f} = \begin{cases} 0 & \text{if } i \neq v, i \neq s \\ y_{v,f}^s & \text{if } i = s \\ -1 & \text{if } i = v \end{cases} \tag{6}$$

$$\forall i, j \in V \cup S : \sum_{v \in V} \sum_{f \in F} z_{i,j}^{v,f} \times d_v^f \leq L_{i,j} \tag{7}$$

The ICN instantiation graph results from the optimization process. It is applied on an input network graph (it can be considered as the larger Internet). After optimization, a set of nodes will host the ICN function. They are identified by a binary variable x_f^s (equals 1 if the node can be upgrade with ICN function and 0 otherwise).

When consumer v sends an interest message for a given ICN data, the request variable $y_{v,f}^s$ is equal to 1 when data is available in the node s and 0 otherwise.

Finally, if a link (i, j) is used in the instantiation graph, the binary variable $z_{i,j}^{v,f}$ will be equal to equal to 0 otherwise.

In eq. (1), we formulate the objective function that minimizes the total placement cost of ICN nodes in the IoT network.

In eq. (2), we ensure that the binary variable y is less than or equal x. In fact, y equals to 1, if and only if v needs f, and f is located on server s (we should not place an ICN function on node s if there is no interest for it).

Eq. (3) states that the optimal server s can serve the consumer nodes interested in the ICN data f. The sum prevents consumers from having to chose between different servers hosting the same ICN data.

Eq. (4) enforces network constraints. We cannot exceed the maximum downloading capacity.

Eq. (5), is relative to node system performance. It enforces the system caching feature of ICN nodes that should not exceed a maximum size.

Eq. (6) represents the network flow conservation between the intermediate ICN nodes and the consumers. In particular, if a node is upgraded to ICN, we ensure that flow balance equals to 1, meaning that it directly serves the incoming consumer interests. Otherwise, if the node is not upgraded, the flow balance is null. At the consumer side, there is no outgoing traffic (left sum is null). Hence the flow balance is negative.

In eq. (7), we ensure that the link capacity between network nodes should not exceed the available network bandwidth.

The above problem is NP-hard due to our combinatorial complex system and difficult to scale up. It can however be easily run on the $CPLEX$ environment. As we target very large ICN infrastructures, a graph based optimization algorithm has to be designed. We present hereafter, another scalable placement algorithm.

OPA Heuristic Algorithm: OPA is based on the well know Gomory-Hu scalable algorithm and it aims at placing ICN/IoT software with the same above mentioned strategy and parameters. Gomory-Hu is an off-line optimization step that compacts the the larger network topology to construct a tree with maximum-flow between all pairs of nodes. Algorithm 1 summarizes the pseudo code of OPA. Hereafter, we describe these main stages.

OPA is based on a Gomory-Hu transformation of the initial graph $G = (V(G), E(G))$ where V are the set of vertices and E are the set of edges. Vertices represent the network servers and edges represent the relation between vertices. The initial network topology is supposed to be a scale-free network (the degree distribution follows power law). The output of this transformation is a cut-tree construction (Gomory-Hu tree, CTC) that represents the maximum-flow between all network server pairs. The cut-tree is used for bottleneck detection.

The model relies on the same information and parameters as in the exact ILP solution (number of servers, system capacity, network capacity, consumer interests). Then, an initial graph that holds the full parameters is created. Further, a Cut-Tree is constructed based on the topology capacity without including the consumer interests. The set of consumer interests is then

Algorithm 1 OPA: Optimal Placement Algorithm for ICN/IoT

```
Input: $V$, $S$, $D^s$, $F$, $f_{size}$, $C^s$, $L_{i,j}$, $d_v^f$, $p_f^s$, $G = (V(G), E(G))$,
$s_v$, $s_f$
Output: $x_f^s$, total ICN placement cost
$CTC \leftarrow$ Cut-Tree-Construction $(G, L_{i,j})$
Upgrading-ICN/IoT-software$(\,)$
ICN/IoT-Caching$(\,)$
```

passed on the tree. During this step, OPA algorithm explores the cut tree creating a path from the gateway to the server hosting the ICN/IoT. If the flow cannot reach the destination (i.e. the original server), caused by shortage in bandwidth on this path, we simply place the new ICN function before the rupture node of the cut tree. A test on system capacity is also performed to ensure that the target ICN node can host this new service. Although a path may not be obtained from the first trial, the problem is still polynomial compared to the NP-hard complexity of the exact solution. Finally, we would like to highlight that OPA was integrated in our virtual migration platform and outsourced in [18]. The Vios platform enables the placement and migration on NFV functions such as CDNs and ICN functions.

IV. Security Considerations

OPA places ICN/IoT nodes closer to consumers. Hence, the security of the IoT will be improved. Indeed, thanks to the ICN features, IoT data integrity is ensured through embedded encryption. We also minimize the network distance between the origin data producers (IoT devices) and consumers (that typically uses a LORA like protocol). If we assume that $Pr(attack) \propto N$ (N is the number of nodes from consumers to producers), in ICN, the $Pr(attack) < N$ A network security cost is proposed as the following:

$$Network\ Security\ Cost = \alpha \times d(Pr, C) \tag{8}$$

Where α characterizes the node stability, Pr is the ICN/IoT gateway (or an intermediate ICN/IoT container after using OPA), C is the consumer group, and finally d is the network distance between Pr and C.

Finally, the security level may be introduced in our initial objective function (1) as an additional constraint.

OPA improves three security issues:

- ICN Caching: OPA acts as a cache relay benefiting from ICN security. Although Intermediate caching nodes may be untrusted, still the ICN infrastructure guarantees the trust for IoT data.
- Processing: OPA enables data analytics, treatment and processing of the cached IoT data. Cached data is treated by intermediate ICN nodes. A security SLA has to be valid between those entities.
- Energy Efficiency: Since OPA is designed for ICN/IoT, it eliminates the need to establish a secure connection between the resource-constrained devices acting as data producers and all the data consumers.

V. OPA: Performance Evaluation

IoT network uses the Ultra Narrow Band (UNB) for the Machine to Machine (M2M) communication. This network poses different problems such as increasing the end to end delay. In general, such network interconnects more than 7 million of devices and uses a point-to-point communication. In this section, the network is assisted with our intelligent algorithm that introduces ICN nodes in different levels. We show that network update upon upgrading optimal nodes by ICN/IoT software reduces the delay and dynamically (through on-line optimization) proposes potential points of operation (placement/upgrade).

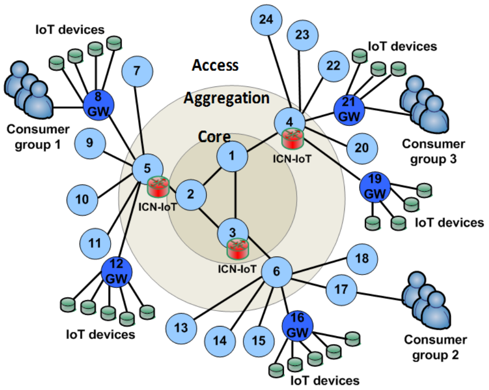


Figure 1: ICN-based IoT distribution network. Given the network topology, consumers requests, and objects served by content providers, OPA model chooses which server should be upgraded with ICN/IoT software.

For the sake of assessing OPA, we propose the following scale-free based topology that represents one of the major complex graphs as shown in Fig. 1. It depicts different IoT gateways that collect IoT data from tiny IoT devices. These gateways assisted with ICN software (ICN/IoT) act as data producers on behalf of IoT devices. OPA algorithm aims to place ICN/IoT nodes to serve data consumer interests.

A. Scale free networks: a Barabasi-Albert model-based network operator

We evaluate our scenario through the well known Barabasi-Albert model [19] [1] undirected and weighted graph. Vertex connectivities follow a scale-free power law distribution $P(k)$. It represents the probability that a vertex interacts to k other vertices is: $P(k) \sim k^{-\gamma}$. The initial graph has 100 vertices (IoT gateway nodes) with a degree distribution that follows the power γ of 2.5, and obeys to the scale-free implementation of $psumtree$. Its Cut-tree-based transformation (Gusfield transformation of the Gomory-Hu algorithm is used here [20]) has only 99 edges (49.5%).

To assess OPA, we introduce the following metrics:

$$Consumer\ delay = \sum_{v \in V} \sum_{f \in F} d(s_v, s_f) \times f_{size} \times \max_{(i,j) \in P_{s_f,s}} \frac{1}{L_{i,j}} \tag{9}$$

$$OPA\ placement\ cost = \sum_{s \in S} \sum_{f \in F} x_f^s \times p_f^s \tag{10}$$

$$OPA\ placement\ delay = \sum_{s \in S} \sum_{f \in F} x_f^s \times f_{size} \times \max_{(i,j) \in P_{s_f,s}} \frac{1}{L_{i,j}} \tag{11}$$

$$Caching\ Strategy = \sum_{s \in S \backslash \{s_f\}} \sum_{f \in F} x_f^s \tag{12}$$

[1] Barabasi-Albert graphs are not random topologies. Instead, they follow a power degree distribution (nonlinear model) so that can be used to assess network performance and as well as interpret the security benefits.

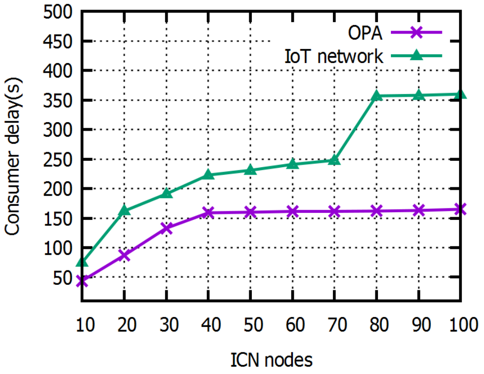


Figure 2: Data consumer delay

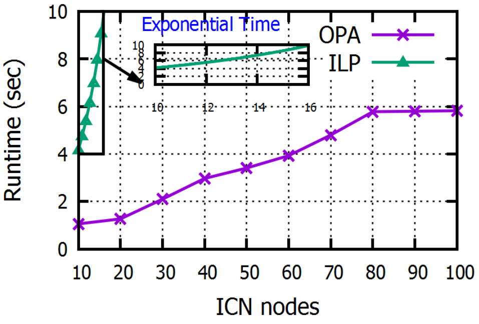


Figure 3: OPA run-time in scale-free network

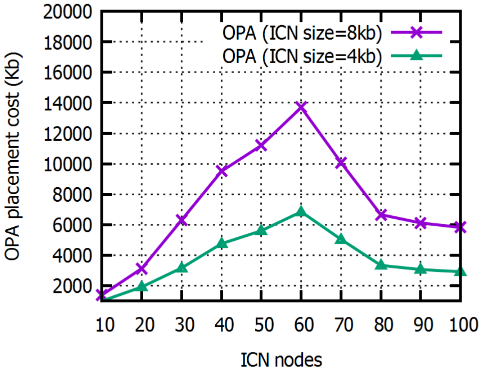
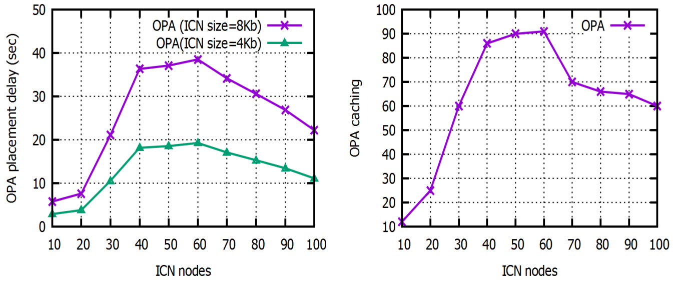


Figure 4: ICN/IoT placement (in network caching) cost



(a) ICN/IoT placement delay. (b) ICN/IoT caching strategy.

Figure 5: OPA in scale-free IoT network.

Eq. (9) defines the consumer delay metric that represents the response time while using OPA instead of the legacy IoT networking. sf and sv represent the data producer (aggregator or ICN gateway) and the data consumer point of attachments respectively. Equations (10) and (11) define the placement cost in terms of memory cost and placement delay (total delay to perform the placement along the shortest path from the IoT gateway to the optimal server node). Eq. (12) represents the average number of instantiated ICN nodes.

Fig. 2 shows the impact of ICN nodes on the data consumer delay. Results show that OPA reduces the total delay.

Fig. 3 depicts the execution time of OPA for our scale-free network. It demonstrates the feasibility and the efficiency of OPA as results are of the order of the second $(6\,\mathrm{sec})$. ILP exact solution is tested using CPLEX environment as a proof of correctness of our model. We provide the curvature of this solution which is exponential and explodes when ICN number equals 20.

Fig. 4 shows the OPA placement cost against ICN node number in scale-free based IoT network. Different ICN sizes (in terms of memory) are used. Fig. 5a and Fig. 5b show the OPA placement delay and the OPA caching respectively (OPA caching equals to the number of ICN migrating nodes which represents the caching policy strategy). The curves have an increasing slope up to $|F| = 60$ and then they decrease. This point represents the average ICN node number stabilizing the IoT network according to consumer group interests.

B. OPA Efficiency: Comparison with IoT networks

IoT networks such as Sigfox [21] are dedicated to low rate wireless data gathering. Several field trials and operational customers have started to use this facility. LoRA corresponds to the wireless part between devices and gateways. As Lora networks cover very large geographical areas, an infrastructure has to be built for data collection and routing. OPA can enhance this core network and provide the flexible ICN function explained before.

In Table II, we highlight how an IoT network such as Sigfox could be improved by applying OPA.

VI. Conclusion and Future Work

We proposed in this paper using ICN for IoT deployment and we introduced a new notion, which is how to assign ICN functionality to ICN/IoT nodes in a dynamic way based on network load and required services latency. In this context, we presented an Optimal Placement Algorithm (OPA) to enhance the caching deployment by network providers. OPA selects optimal network locations to serve as intermediate IoT publisher and pursue in-network caching. We illustrated IoT benefits from the in-network caching feature in ICN especially when applying our proposed algorithm. And we compared OPA in IoT over ICN with IoT over SigFox network (as an example of a popular IoT network deployment in France). Encouraging results assure that OPA is scalable and efficient

TABLE II: Efficiency comparison for OPA and IoT SigFox network

Metrics	***ICN/IoT assisted OPA***	***Sigfox***
Caching strategy	Migrating ICN/IoT nodes acting as edge/fog computing nodes.	Using cloud data centers.
Delay	Minimize the end-to-end consumer delay	Significant.
Optimization cost	Additional cost of placement of ICN/IoT nodes	Minimum deployment cost.
Bitrate	High throughput network due to the in-network caching feature.	Low throughput network due to the UNB modulation.
Security	Object-based security that allows caching in untrusted intermediate nodes (proxies, caches, etc.).	Session-based security, frequency hopping.
Actuation latency	Bounded in with cache avoidance (OPA helps in routing).	Unbounded
Medium Access Control (MAC) layer	ICN networking stack that implies optimal bandwidth occupancy.	Without collision-avoidance that limits the bandwidth of IoT gateways [21].

in terms of placing ICN/IoT nodes in a dynamic way. Our next step is to compare OPA in IoT over ICN against IoT solutions over WiFi and cellular networks without ICN. We also work on a distributed algorithm for OPA. Finally, Vios platform will be enriched with ICN dockers.

References

[1] K. Pentikousis, B. Ohlman, D. Corujo, G. Boggia, G. Tyson, E. B. Davies, A. Molinaro, and S. Eum, "Information-Centric Networking: Baseline Scenarios," RFC 7476, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7476.txt

[2] D. Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. C. Schmidt, and M. Whlisch, "Information-Centric Networking (ICN) Research Challenges," RFC 7927, Jul. 2016. [Online]. Available: https://rfc-editor.org/rfc/rfc7927.txt

[3] Y. Zhang, D. Raychadhuri, L. A. Grieco, S. Sabrina, H. Liu, S. Misra, and R. Ravindran, "ICN based Architecture for IoT," Internet Engineering Task Force, Internet-Draft draft-zhang-icnrg-iot-architecture-00, Jul. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-zhang-icnrg-iot-architecture-00

[4] R. E. Gomory and T. C. Hu, "Multi-Terminal Network Flows," *Journal of the Society for Industrial and Applied Mathematics*, vol. 9, no. 4, pp. 551–570, 1961. [Online]. Available: http://dx.doi.org/10.2307/2098881

[5] D. R. Ford and D. R. Fulkerson, *Flows in Networks*. Princeton, NJ, USA: Princeton University Press, 2010.

[6] NetInf. (2016, Jun.) Network of information (netinf). [Online]. Available: https://www.netinf.org

[7] NDN. (2016, Jun.) Named data networking. [Online]. Available: https://named-data.net/

[8] CCN. (2016, Jun.) The ccnx project, parc a xerox company. [Online]. Available: http://blogs.parc.com/ccnx/

[9] PSIRP. (2016, Jun.) Publish-subscribe internet routing paradigm. [Online]. Available: http://www.psirp.org/

[10] V. Jacobson. (2006, Jun.) A new way to look at networking. [Online]. Available: You Tube. Google Talk.

[11] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," *Commun. ACM*, vol. 55, no. 1, pp. 117–124, Jan. 2012. [Online]. Available: http://doi.acm.org/10.1145/2063176.2063204

[12] Y. Zhang, H. Zhang, and L. Zhang, "Kite: A mobility support scheme for ndn," in *Proceedings of the 1st ACM Conference on Information-Centric Networking*, ser. ACM-ICN '14. New York, NY, USA: ACM, 2014, pp. 179–180. [Online]. Available: http://doi.acm.org/10.1145/2660129.2660159

[13] Y. Zhang, A. Afanasyev, J. Burke, and L. Zhang, "A survey of mobility support in named data networking," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2016, pp. 83–88.

[14] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch, "Information centric networking in the iot: Experiments with ndn in the wild," in *Proceedings of the 1st ACM Conference on Information-Centric Networking*, ser. ACM-ICN '14. New York, NY, USA: ACM, 2014, pp. 77–86. [Online]. Available: http://doi.acm.org/10.1145/2660129.2660144

[15] J. Quevedo, D. Corujo, and R. Aguiar, "Consumer driven information freshness approach for content centric networking," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2014, pp. 482–487.

[16] S. Vural, P. Navaratnam, N. Wang, C. Wang, L. Dong, and R. Tafazolli, "In-network caching of internet-of-things data," in *2014 IEEE International Conference on Communications (ICC)*, June 2014, pp. 3185–3190.

[17] M. A. Hail, M. Amadeo, A. Molinaro, and S. Fischer, "Caching in named data networking for the wireless internet of things," in *2015 International Conference on Recent Advances in Internet of Things (RIoT)*, April 2015, pp. 1–6.

[18] Tamayo-IbnKhedher. (2016) vios (virtual infrastructure optimization simulator). [Online]. Available: https://github.com/TelecomSudparis-RST/vIOS

[19] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999. [Online]. Available: http://www.sciencemag.org/cgi/content/abstract/286/5439/509

[20] D. Gusfield, "Very simple methods for all pairs network flow analysis," *SIAM J. Comput.*, vol. 19, no. 1, pp. 143–155, Feb. 1990. [Online]. Available: http://dx.doi.org/10.1137/0219009

[21] Sigfox. (2017) Sigfox developer portal. [Online]. Available: http://makers.sigfox.com/

Disclaimers: "Intel technologies features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer. No computer system can be absolutely secure. Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance. Cost reduction scenarios described are intended as examples of how a given Intel- based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction. Intel, the Intel logo and others are trademarks of Intel Corporation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others. 2017 Intel Corporation.