

# Performance Implications for IoT over Information Centric Networks

Akhila Rao<sup>\*†</sup>, Olov Schelén<sup>\*</sup>, Anders Lindgren<sup>†\*</sup>

<sup>\*</sup>Luleå University of Technology, Sweden

<sup>†</sup>SICS Swedish ICT AB, Sweden

{akhila.rao, andersl}@sics.se, olov.schelen@ltu.se

## ABSTRACT

Information centric networking (ICN) is a proposal for a future internetworking architecture that is more efficient and scalable. While several ICN architectures have been evaluated for networks carrying web and video traffic, the benefits and challenges it poses for Internet of Things (IoT) networks are relatively unexplored. In our work, we evaluate the performance implications for typical IoT network scenarios in the ICN paradigm. We study the behavior of in-network caching, introduce a way to make caching more efficient for periodic sensor data, and evaluate the impact of presence and location of lossy wireless links in IoT networks. In this paper, we present and discuss the results of our evaluations on IoT networks performed through emulations using a specific ICN architecture, namely, content centric networking (CCN). For example, we show that the newly proposed UTS-LRU cache replacement strategy for improved caching performance of time series content streams reduces the number of messages transmitted by up to 16%. Our findings indicate that the performance of IoT networks using ICN are influenced by the content model and the nature of its links, and motivates further studies to understand the performance implications in more varied IoT scenarios.

## 1. INTRODUCTION

The Internet of Things (IoT) is a wide umbrella that covers several different types of networks, with very varying devices, connectivity, data models and applications. Some examples are home automation networks, vehicular networks, industrial monitoring networks and smart city enabling networks. As the number of IoT devices grows and becomes more ubiquitous, there is a pressing need to provide more efficient and scalable network support for such applications. In the past decade, the research community has looked at multiple future internetworking architectures to improve the efficiency of networks and develop them to meet the demands of future applications. One such approach for a future network architecture is the information centric networking (ICN) paradigm. Several large research projects have proposed architecture definitions such as NetInf [10], NDN [28] and CCN [20] for information centric networking. The ICN approach attempts to modify the

host-centric communication paradigm of current networks, to an information centric one where named content objects are directly addressed and requested. This essentially decouples content from its location or the device it resides on. One of the goals of ICN is to evolve networks to be inherently efficient and scalable for content distribution. Named content objects, name based routing, in-network caching and securing content instead of securing end devices are some of the key features of ICN.

A large portion of the research on ICN has been focused on evaluating Internet-scale networks with video and web traffic request based content models such as those exhibited by YouTube and Netflix. IoT networks, however, are significantly different in terms of resources available, content models, applications and metrics. We do however believe that many of these characteristics of IoT networks indicate that there are significant potential benefits that can be achieved by utilizing information centric techniques for IoT applications. The evaluation of ICN for IoT networks is relatively new with a lot of open research questions.

In this paper, our contribution is an evaluation of ICN in the context of a prevalent IoT network model. We provide specific insight into in-network caching of time periodic sensor data and propose a cache replacement strategy called UTS-LRU that identifies periodic data and improves caching performance for such data. We evaluate the performance of caching in relation to packet retransmission time on lossy wireless links. We also evaluate the performance impact related to specific locations of lossy links in the network.

## 2. DESIGN CHOICES FOR IOT OVER ICN

Design choices for IoT to efficiently harness the benefits of ICN have been proposed by Lindgren et al. [17]. The authors discuss benefits and challenges in adapting IoT to ICN, and identify trade-offs related to their design choices. Features such as distributed caching, inherent handling of consumer mobility, context based content retrieval, and energy efficient object security and delivery are some of the obvious benefits for IoT on ICN. Challenges include, handling requests for alarm and triggered data, sending device specific commands to actuators, producer mobility, and requests for 'latest value' from a content stream.

A key design choice relevant to our work is to model each IoT data object as being immutable, and furthermore to include sequence numbers in the naming scheme to model dynamic data as streams of immutable objects. This prevents problems with cache inconsistency and the need for global synchronization of caches. An important design consideration for any IoT network is how producers advertise the properties of the content they publish. This could consist of the namespace used for content publishing, a way to map time to sequence numbers so that requesters can infer the name of the desired content, a model for content generation (trig-

gers for triggered content, time period for periodic content) and more. Our work implements the design proposals of sequence numbers and content immutability, and builds on the assumption of some others such as capability advertisements.

## 2.1 Content Centric Networking

Over the past years, several network architectures embodying the information centric networking paradigm have been defined, such as the previously mentioned NetInf, NDN and CCN. In our evaluation of IoT on an ICN architecture, we have chosen to use the *content centric networking* (CCN) architecture. This decision was motivated by its popularity in the research community. Specifically, we use the latest version of CCN - CCNx 1.0. Although our evaluations have been performed on the CCN architecture, many results can be generalized to other ICN architectures.

There are many introductions to CCN [18], so we only discuss specific aspects of CCN that are relevant to our study of IoT networks. In CCN, when a client application is interested in content, it expresses this in the form of an *interest message* with the *name* of the content required. Name based forwarding is used to forward the message through the network to any location that may be able to respond. The response is a *data message* or *content object* (CO) with a matching name that takes the reverse path of the interest message. CCN performs stateful forwarding which allows flow balance between interest and data messages and also enables ubiquitous caching.

When a *content object* (CO) is on its reverse path towards the requester, it is cached at intermediate nodes along the path. This translates to the flow of content dictating where the content gets cached. It is hence cached at the edge of the network in regions where it is more often requested. The default caching strategy in CCN is *cache all*, where each node caches any CO that passes through it, such that a CO is cached all along its path to the requester. The default cache replacement strategy is *least recently used* (LRU). This in-network caching improves network scalability by reducing redundant traffic using locally cached content.

Losses in CCN can be handled with router level retransmissions and/or requester level retransmissions, on interest timeout. CCN dictates that the client applications choose the retransmission time appropriate for them.

## 2.2 Model for IoT over CCN

IoT encompasses varied topologies, network architectures, content models and applications. Our study is narrowed down to a specific IoT model which captures some major features of IoT networks. In this section, we provide the details of our model and the assumptions made. We also bring out the differences between models used for Internet scale networks and our model for IoT networks. We model IoT networks as possessing the following characteristics.

- Nodes are constrained in memory and computational resources.
- Efficiency and scalability are major concerns due to resource constrained nodes and large number of devices.
- Content is generated and published as time series data, where most consumers are interested in the latest value of a measurement. Data is hence ephemeral and of interest to consumers mostly within a certain time window until the next measurement is available.
- Edge links in the networks are typically wireless and hence lossy. But unlike in other networks, the producers are usually sensors and hence producers may also be connected by lossy wireless links.

In Internet scale networks, popularity of content is modelled by the Zipf's law [3]. At this scale, content popularity does not change over short time durations. This type of content forms the majority of the Internet traffic, in which case, it is beneficial to cache popular content for long durations. Content in IoT networks is significantly different with most content being small packets of ephemeral data such as of sensor measurements, actuator commands, alarms, control and management messages etc. An example use case is crowd sensing applications where many users produce either redundant or related information, possibly sensitive data that users do not want to store in a central cloud repository. Distributed consumers can probe selectively for information and upon some findings ask for more information from related sensors. Sensors may be intermittently connected due to network conditions or power conditions.

Most sensors generate periodic data in a time series manner where each new CO generated is a more recent value of a reading than the previous one. Content from sensors are modelled as streams of immutable objects being published with increasing sequence numbers in their names. The immutability condition here is important as it prevents cache inconsistencies. A mapping from time to sequence numbers and a way to map the context of data to unique names are assumed to be defined by the producers and advertised through capability advertisements [17]. To improve caching performance for such data, we propose a new caching policy in Section 4.2.

When a new CO from the content stream of a producer is published (made available) it is requested by consumers interested in it. These requests are highly correlated in the time window after their publishing and before a newer one is made available. Requests for older data are either non-existent or infrequent in time series content streams. This model is very different from those used for traffic in Internet scale networks where requests for a certain content object could be spread over long time durations.

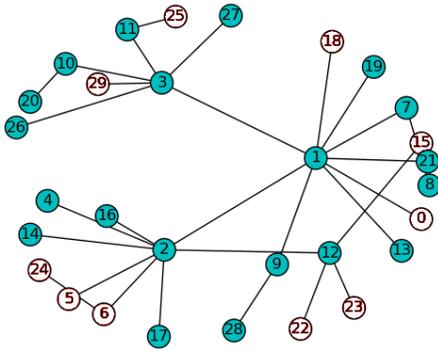
A CO published by a sensor node is assumed to always be available at the source indefinitely. Through the previously mentioned capability advertisements, consumers can be made aware of the timing and the rate at which periodic content is generated in streams. This allows consumers to estimate the publishing of content and request it appropriately such that interest is not expressed before a CO is generated at the source. Some research on streaming in ICN, however, has considered buffering interests that can be served locally within a defined timeout duration [27]. This issue is general to ICN and hence we do not attempt to address it.

## 3. EVALUATION SETUP AND SCENARIOS

We focus our evaluations on a type of topology that reflects several characteristics of typical real world scenarios. We chose a topology based on the Barabási-Albert (BA) graph for scale-free networks [7]. The current Internet is known to be scale-free [25], and the BA graph is suitable for networks with similar topological properties. Figure 1 shows the randomly generated BA network graph used for our evaluations. To ensure that the results were not specific to the particular instance of the graph, we performed evaluations with two instances of a BA graph and found the results to be very similar. We hence present the results from only one instance in this paper. Details of the chosen topology are provided in Table 1.

### 3.1 Link Model

IoT networks typically have wireless links near both the producers (e.g. sensors) and the consumers (e.g. mobile devices) which mean the edge links of the network are lossy. They are often deployed in dynamic environments with fading links, which could experience long durations of fade or outage. We attempt to capture this link behavior by using a simple two-state Markov chain model



**Figure 1: Barabási-Albert 1 edge preferential attachment network graph. The network has 30 nodes, with 10 sensor nodes (white circles) which are content publishers and 20 consumers (blue circles) interested in the content.**

**Table 1: Network topology parameters**

Number of nodes	30
Number of sensor nodes	10
Number of consumer nodes	20
Average path length	3.38 hops
Max. path length	6 hops

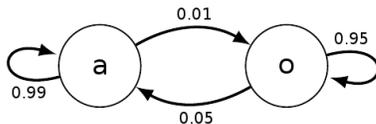
as shown in Figure 2, where each link changes between an active(a) and an outage(o) state based on a transition probability matrix  $P$ . This model is widely known as the Gilbert-Elliot model [13].

$$P = \begin{bmatrix} 0.99 & 0.01 \\ 0.05 & 0.95 \end{bmatrix}$$

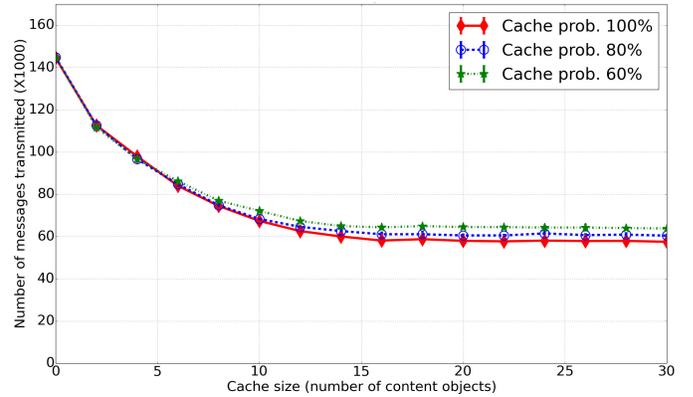
The average active and outage durations were 10 and 2 seconds respectively, chosen from the expected range for a link between stationary or slow moving devices [15]. When a link is in the active state, it has a low packet drop probability ( $P_{act} = 0.01$ ), and a high drop probability when it is in the outage state ( $P_{out} = 0.5$ ). This simple model captures the time correlation property of packet losses and provides a more accurate model than time independent random losses. In our evaluations, we study the impact of the location of the lossy links in the network by either letting all edge links exhibit the above loss characteristics, only the links connecting producers, or only the links connecting consumers.

### 3.2 Metrics

The scalability and energy efficiency of a network is directly related to the number of messages transmitted or bandwidth used in the network to achieve a certain communication task. A key benefit of caching is that it can reduce the number of message transmissions in the network. We hence use the number of messages transmitted during an emulation run as our key metric to quantify performance. This includes interest message and data message



**Figure 2: A 2 state Active, Outage Markov chain describing the link model for lossy links.**



**Figure 3: Plot of number of transmissions versus cache size for caching probabilities 100%, 80% and 60%. The 100% caching probability scenario is the same as the default cache all scenario. The cache replacement strategy used is LRU.**

transmissions.

While using the number of messages transmitted as a metric in scenarios with losses, interest messages could be dropped after attempting to retransmit a certain number of times (default in CCNx 1.0 is 2)). This metric would be unfair when comparing scenarios with different delivery rates. To make it fair even in lossy scenarios, we increase the number of retransmissions to a high value of 10 such that the end-to-end delivery rate even in lossy scenarios is always above 99.5%.

### 3.3 Set-up

Our evaluations were performed as an emulation on a network of CCN-lite nodes [1]. CCN-lite is a light-weight implementation of the CCNx 1.0 protocol in C. Even though it is a bare minimum implementation, it includes the key features necessary for our evaluations. Multiple instances of CCN-lite nodes are initialized on a single host machine and the topology (Figure 1) is imposed on them using the link model described in Section 3.1 for the edge links<sup>1</sup>. The emulation was performed over 200 second runs and repeated 10 times. Our plots show averaged results with 95% confidence intervals.

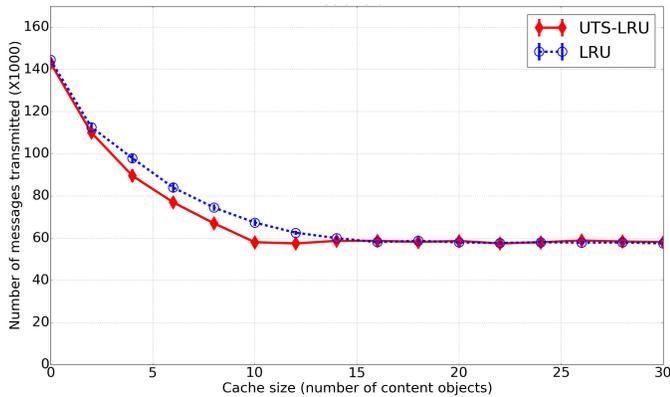
## 4. RESULTS

To study properties of caching in CCN IoT networks, we begin in Section 4.1 by presenting the results of evaluations with lossless links. We then, in Section 4.3, present the results of our evaluations with lossy links.

### 4.1 Lossless Networks

Figure 3 plots the total number of messages transmitted in the network, as a function of cache size (number of objects). It shows that the number of messages transmitted in the network goes from its maximum when no caching is used (cache size zero) to a drastically lower value even with a caching capacity of only a few objects. This plot is indicative of the impact of our content model on caching in CCN. Each CO generated as part of a content stream is mostly requested within the time period before the publishing of a new one. This translates to the amount of caching needed at any node in the network as a function of the number of *fresh* COs

<sup>1</sup>Emulation code and scripts have been made available at <https://github.com/akhila-rao/IoT-ICN-emulation.git>



**Figure 4: Plot of number of transmissions versus cache size in a lossless network scenario for UTS-LRU and LRU cache replacement strategies. The caching strategy of *cache all* was used for both replacement strategies.**

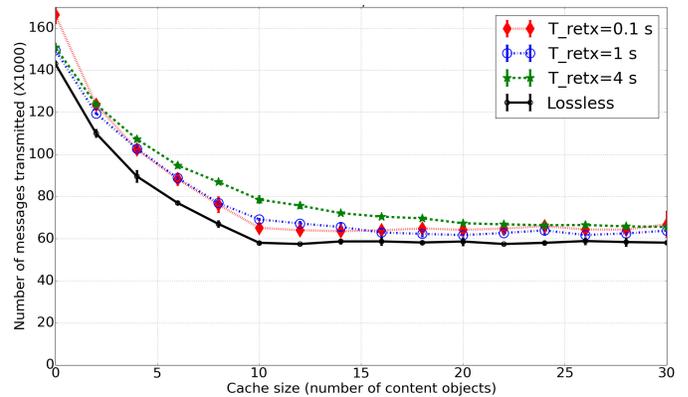
present in the network at any given time. In our current scenario, there are 10 content sources or sensors each publishing a content stream of data. We can see from the plot that the number of messages transmitted level off at less than half the original number of message transmissions when the cache size at each node is in the same order of magnitude as the number of streams. Having a cache size much larger than the number of content streams is not beneficial when the requests for the data are highly correlated in time and confined to one time period of publishing.

Figure 3 has three curves for different caching probabilities at the nodes. When caching probability is set to a value other than 100% it samples a random number to make the caching decision. In scenarios where cache resource is small compared to the amount of content that flows through the network, having a probability based caching policy increases the cache diversity by spreading out the contents in caches along the way from the source to the consumers. In our scenario, however, we see that reducing the caching probability does not provide gains in number of transmissions even at very small cache sizes. We have a small number of content streams in our example scenario, and hence, the benefit of probability based caching is not witnessed. The point to note here is that the number of content streams and not the number of COs decide the caching behavior or requirement.

## 4.2 UTS-LRU Cache Replacement Strategy

On identifying that the cache sizes at nodes need not be much larger than the number of content streams in the network, we wanted to see if there was a way to use the knowledge that content is being published as a time series, with consumers interested in only the latest value, and improve the cache management strategy based on it. Content in different streams could be published and consumed at different rates, so an object that is published less frequently could be evicted from the cache because requests to it are spread over a longer time duration. If, instead, a more recent object from a content stream replaces an older object from the same stream, if present in the cache, then it could increase the hit rate for that cache. Based on this idea, we implemented a cache replacement strategy that, on identifying a CO as part of a time series content stream, first looks to replace the oldest available object of the same stream. If not present, it reverts to using LRU. We call this replacement strategy UTS-LRU (Update Time Series - Least Recently Used).

Figure 4 shows the comparison of our evaluation performed with



**Figure 5: Plot of number of transmissions versus cache size for retransmit times ( $T_{retx}$ ) 0.1, 1.0 and 4.0 seconds, in a scenario when all edge links are lossy. The caching strategy used is *cache all*, and the replacement strategy used is UTS-LRU.**

UTS-LRU and the default - LRU. We see that UTS-LRU always performs better than the default LRU strategy. At its best, UTS-LRU sends 16% fewer messages than LRU. We can also see that the number of messages transmitted for UTS-LRU flattens out completely after the cache size equals the number of streams, while LRU does the same at a higher cache size. The order in which new COs of different streams arrive at a cache is not consistent since they have different publishing rates and other random time factors. LRU could thus replace an object still being requested in the network, while UTS-LRU would reduce this occurrence by replacing an older object in the same stream. This stretches LRU's cache size requirement beyond what UTS-LRU would require.

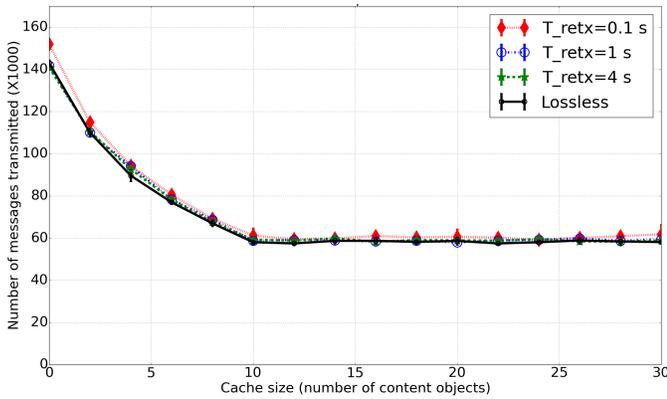
## 4.3 Lossy Networks

Edge links in IoT networks are more likely to be wireless and lossy in nature. As an example, edge links could consist of links to sensor devices, and at the other end consist of wireless access links by which consumers connect to them. This example describes a scenario where links near the producers of content in an IoT network could be wireless, which is atypical for traditional networks.

Our evaluation of lossy networks was done by setting only the edge links in our topology as lossy. The loss model used on these links is as described in Section 3.1. We are interested in understanding the tradeoffs involved in choosing the retransmit time and the impact of location of lossy links in the network.

We begin by looking at the impact of retransmit time for lossy links in the network. There are two phenomena that influence performance. Since the channel has an average outage time period of 2 seconds, if the retransmit time is short compared to the channel outage time, then the retransmitted message is more likely to encounter a loss than if it was transmitted after a longer time. On the other hand, if the retransmit time is large, the nearest caches are more likely to have evicted the CO that is being requested for retransmission, potentially requiring the interest to travel more hops. So there is a tradeoff between cost to retransmit while the channel is still in outage and the cost to travel to a farther cache to obtain the required CO. Having a large retransmit time also means that, for objects being published more frequently than the retransmit time, a new object in the stream could be available while the consumer is still requesting an older one.

Figure 5 shows the results of an evaluation where  $T_{retx}$  has been varied with values 0.1, 1.0 and 4.0 seconds. In the no caching sce-



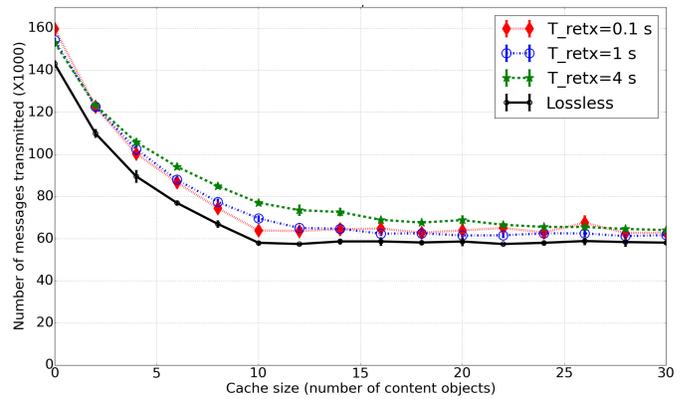
**Figure 6:** Plot of number of transmissions versus cache size for retransmit times ( $T_{retx}$ ) 0.1, 1.0 and 4.0 seconds, in a scenario when only edge links to content producers are lossy. The caching strategy used is *cache all*, and the replacement strategy used is UTS-LRU.

nario, as expected, we see  $T_{retx} = 4$  performs the best and  $T_{retx} = 0.1$  performs the worst due to the time correlated losses on the channel. In the scenario where cache size is large enough for COs not to be evicted before retransmission request, we expected  $T_{retx} = 4$  to perform the best. For the non-zero cache sizes in between, we expected to see a tradeoff. The results however for these scenarios are not as we expected.  $T_{retx} = 4$  instead has the worst performance for all non-zero cache sizes. There is no visible tradeoff between  $T_{retx} = 0.1$  and  $T_{retx} = 1$  as we increase cache size. This indicates that the cost of travelling more hops to reach the required content seems to be larger than the cost of increased number of retransmissions due to link outage coherence. The results, however, are affected by the average number of hops in the network and the outage loss rate. But for the chosen topology and loss model, a smaller retransmit time has better performance.

Figures 6 and 7 show results of a similar evaluation, but with only the edge links connecting producers being lossy and alternatively the edge links connecting consumers being lossy. The results we see in Figure 7 are similar to the results we saw for the *all edge links lossy* scenario in Figure 5. In Figure 6 we see that the performance for all the  $T_{retx}$  values are similar. Observing Figures 6 and 7 in comparison with Figure 5 we see that the behavior of the curves in the lossy networks scenario is influenced only by the losses near the consumers. The effect of losses near the producers is overcome by in-network caching. The effect of losses near the consumers, however, are unavoidable and exacerbated by small caches and long retransmit times. In summary from all the lossy network evaluation plots, we learn that network performance is affected by the location of the lossy links in the network.

## 5. RELATED WORK

Using ICN as a basis for a future internetworking architecture has been studied for the past decade with several works having evaluated different aspects. Most ICN evaluations are focused on the larger Internet and assume a web and video traffic content model, where the popularity of content objects follows the Zipf's law [22]. Rossini et al. [24], Chai et al. [9] and Rossi et al. [22] have evaluated the performance and benefits of caching in ICN. Li et al. [16], Bernardini et al. [8] and Nakayama et al. [19] have looked at ICN caching with a popularity based approach. Wang et al. [26] study the optimal distribution of cache resources in a network for content



**Figure 7:** Plot of number of transmissions versus cache size for retransmit times ( $T_{retx}$ ) 0.1, 1.0 and 4.0 seconds, in a scenario when only edge links to consumers are lossy. The caching strategy used is *cache all*, and the replacement strategy used is UTS-LRU.

popularity based on the Zipf's model. Rossi et al. [23] have evaluated heterogeneous cache resource distribution in a network based on its topological properties. Fricker et al. [12] have looked at different types of traffic on the Internet such as web traffic, file sharing, user generated content and video on demand. They evaluate cache requirements for these different traffic sources and propose a way to handle a traffic mix.

There has been more recent interest in evaluating ICN for IoT networks as well. Quevedo et al. [21] have performed a basic evaluation of ICN for IoT networks and conclude that ICN can be beneficial in solving several IoT challenges. Amadeo et al. [4] perform an architectural superposition of what ICN offers and how IoT can use it. They also mention several opportunities for open challenges in implementing IoT applications over ICN.

Hail et al. [14] present multiple caching strategies for ICN IoT networks and propose a new caching strategy based on the freshness of the data, energy, and memory resources available at a node. There have been some attempts to evaluate ICN for the specific data patterns of IoT. Amadeo et al. [5] look at how push traffic can be supported in ICN, which is inherently pull based, and propose some solutions. Francois et al. [11] optimize a push mechanism for forwarding content to IoT consumers that require updates at different frequencies.

While some, such as the authors of [6] and [2] have considered wireless links in the network, most have not included it in their evaluations. The work by Abu et al. in [2] shows some interesting results about the effects of lossy links and interest retransmission on pending interest table sizes. They assume a link model with independent losses on each link.

The ephemeral data aspect of IoT, the correlated losses and their impact on performance in ICN are yet to be evaluated and provide us the motivation for our study of ICN for IoT networks.

## 6. CONCLUSIONS

In this paper, we discussed some benefits that ICN can bring to IoT networks. We studied the impact of ephemeral IoT data on the performance of caching. We proposed the UTS-LRU cache replacement strategy for improved caching performance of time series content streams and showed that at its best, it reduces the number of messages transmitted by 16%. We emulated a lossy network, compared its caching performance to lossless networks and looked

at the tradeoffs in choosing retransmit time for dropped packets. Finally, we studied the performance impact of the location of lossy links in the network and concluded that losses significantly affect performance only when they are located near the consumers.

Our work is a step in the direction of addressing the challenges of adopting ICN for IoT networks. We address the scenario of having time series periodic sensor data in the network and show how that influences caching behavior. In future work, we would like to extend the evaluation to a larger scale with different topologies and access patterns. We would also like to address the additional challenges of triggered data, actuator and alarm data, each of which involve a different content model.

## Acknowledgments

This work was partially funded by the Future Networking Solutions action line of EIT Digital, by the KKS funded READY project, by the Vinnova project GreenIoT, and by the Vinnova funded Cloud-berry Datacenters.

## 7. REFERENCES

- [1] CCN-lite. <http://www.ccn-lite.net/>.
- [2] A. J. Abu, B. Bensaou, and J. M. Wang. Interest packets retransmission in lossy ccn networks and its impact on network performance. In *Proceedings of the 1st international conference on Information-centric networking*, pages 167–176. ACM, 2014.
- [3] L. A. Adamic and B. A. Huberman. Zipf’s law and the internet. *Glottometrics*, 3(1):143–150, 2002.
- [4] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro. Named data networking for iot: an architectural perspective. In *Networks and Communications (EuCNC), 2014 European Conference on*, pages 1–5. IEEE, 2014.
- [5] M. Amadeo, C. Campolo, and A. Molinaro. Internet of things via named data networking: The support of push traffic. In *Network of the Future (NOF), 2014 International Conference and Workshop on the*, pages 1–5. IEEE, 2014.
- [6] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch. Information centric networking in the iot: Experiments with ndn in the wild. *arXiv preprint arXiv:1406.6608*, 2014.
- [7] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.
- [8] C. Bernardini, T. Silverston, and O. Festor. Mpc: Popularity-based caching strategy for content centric networks. In *Communications (ICC), 2013 IEEE International Conference on*, pages 3619–3623. IEEE, 2013.
- [9] W. K. Chai, D. He, I. Psaras, and G. Pavlou. Cache "less for more" in information-centric networks. In *NETWORKING 2012*, pages 27–40. Springer, 2012.
- [10] C. Dannowitz. Netinf: An information-centric design for the future internet. In *Proc. 3rd GI/ITG KuVS Workshop on The Future Internet*, 2009.
- [11] J. François, T. Cholez, and T. Engel. Ccn traffic optimization for iot. In *Network of the Future (NOF), 2013 Fourth International Conference on the*, pages 1–5. IEEE, 2013.
- [12] C. Fricker, P. Robert, J. Roberts, and N. Sbihi. Impact of traffic mix on caching performance in a content-centric network. In *Computer Communications Workshops (INFOCOM WKSHPs), 2012 IEEE Conference on*, pages 310–315. IEEE, 2012.
- [13] E. N. Gilbert. Capacity of a burst-noise channel. *Bell system technical journal*, 39(5):1253–1265, 1960.
- [14] M. A. Hail, M. Amadeo, A. Molinaro, and S. Fischer. Caching in named data networking for the wireless internet of things. In *Recent Advances in Internet of Things (RIoT), 2015 International Conference on*, pages 1–6. IEEE, 2015.
- [15] L. Korowajczuk. *LTE, WiMAX and WLAN Network Design, Optimization and Performance Analysis*. Wiley, 2011.
- [16] J. Li, H. Wu, B. Liu, J. Lu, Y. Wang, X. Wang, Y. Zhang, and L. Dong. Popularity-driven coordinated caching in named data networking. In *Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems*, pages 15–26. ACM, 2012.
- [17] A. Lindgren, F. B. Abdesslem, B. Ahlgren, O. Schelén, and A. M. Malik. Design choices for the iot in information-centric networks. In *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 882–888. IEEE, 2016.
- [18] M. Mosko, I. Solis, E. Uzun, and C. Wood. CCNx 1.0 protocol architecture. Technical report, PARC, 2015.
- [19] H. Nakayama, S. Ata, and I. Oka. Caching algorithm for content-oriented networks using prediction of popularity of contents. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 1171–1176. IEEE, 2015.
- [20] PARC. Content Centric Networking CCN. <https://www.parc.com/work/focus-area/content-centric-networking/>.
- [21] J. Quevedo, D. Corujo, and R. Aguiar. A case for icn usage in iot environments. In *Global Communications Conference (GLOBECOM), 2014 IEEE*, pages 2770–2775. IEEE, 2014.
- [22] D. Rossi and G. Rossini. Caching performance of content centric networks under multi-path routing (and more). *Relatório técnico, Telecom ParisTech*, 2011.
- [23] D. Rossi, G. Rossini, et al. On sizing ccn content stores by exploiting topological information. In *INFOCOM Workshops*, pages 280–285, 2012.
- [24] G. Rossini and D. Rossi. A dive into the caching performance of content centric networking. In *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2012 IEEE 17th International Workshop on*, pages 105–109. IEEE, 2012.
- [25] X. F. Wang and G. Chen. Complex networks: small-world, scale-free and beyond. *Circuits and Systems Magazine, IEEE*, 3(1):6–20, 2003.
- [26] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie. Optimal cache allocation for content-centric networking. In *Network Protocols (ICNP), 2013 21st IEEE International Conference on*, pages 1–10. IEEE, 2013.
- [27] H. Xu, Z. Chen, R. Chen, and J. Cao. Live streaming with content centric networking. In *Networking and Distributed Computing (ICNDC), 2012 Third International Conference on*, pages 1–5. IEEE, 2012.
- [28] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, et al. Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3):66–73, 2014.