iCAAP: information-Centric network Architecture for Application-specific Prioritization in Smart Grid

Anju K. James+, George Torres+, Sharad Shrestha+, Reza Tourani†, Satyajayant Misra+ *Member, IEEE*

***Abstract*—The smart grid is equipped with bi-directional information flow between its devices, aiming at automation, improved stability, resilience, and robust security. However, enabling effective and reliable communication in a smart grid is a challenging task. The majority of the proposed networking architectures fall short in addressing the key aspects of smart grid communication, including device heterogeneity, protocols and standards interoperability, and particularly application quality-of-service (QoS) requirements.**

In this paper, we propose iCAAP, an information-centric, QoS-aware network architecture that aims to satisfy the low latency, high bandwidth, and high reliability requirements of smart grid communications. In iCAAP, we categorize smart grid traffic (emanating from diverse applications) into three priority classes to enable preferential treatment of traffic flows. Our simulation results demonstrate the higher scalability of iCAAP in satisfying the stringent requirements of high priority traffic compared to the state-of-the-art.

***Index Terms*—NDN, smart grid, Quality-of-Service, communication.**

I. Introduction

The smart grid technology is realized as the next generation power grid system with two-way communication flow between the grid's appliances across the distribution, transmission, and generation subsystems. As the primary objective, the emerging smart grid technology aims at providing higher reliability, improved stability and security, and taking advantage of its socio-economic impacts (e.g., reduction in energy consumption). It is evident that the communication and networking infrastructure plays a crucial role in successful deployment of smart grid for achieving these goals. The smart grid communication is particularly important when considering the diversity of smart grid applications and their unique and stringent requirements in terms of latency, bandwidth, and reliability. Thus, a scalable and versatile communication infrastructure is imperative for the wide adoption of smart grid.

The majority of the proposed smart grid communication architectures adopt the widely used Internet Protocol (IP), which enables packet delivery using the source and destination addresses. Utilizing the host-centric communication model, however, falls short in meeting the quality-of-service (QoS) requirements of smart grid communication, such as reliability, many-to-many, and low latency [1], [2]. For instance, enabling the distributed energy market in smart grid requires efficient many-to-many communications, which can be only achieved by maintaining a large number of multicast trees in IP-based networks–a non-scalable practice when using IP-based communication architectures [3].

Over the past decade, a few initiatives that leverage the Information-Centric Networking (ICN) paradigm [4], [5], and its prominent realization Named-Data Networking (NDN) [6], have been proposed to satisfy the stringent requirements of smart grid communication [7]–[9]. These initiatives are in line with the growing popularity of data-centric Internet applications, in which obtaining the desired data is preferred over identifying the data source. It has been argued that NDN's unique features, such as expressive naming, built-in security, inherent multicast, and flexible forwarding plain make ICN a suitable candidate for smart grid communication [10]. These efforts resulted in a few data-centric architectural designs, which facilitate the communication between smart grid appliances, improve grid's stability, or enable publish-subscribe data dissemination. While these initiatives are the precursors of a holistic data/information-centric smart grid view, further innovation is needed for meeting the stringent QoS expectations of smart grid communication. In particular, the low latency, high bandwidth, and reliability of various smart grid traffic. In [3] and [10], the authors have presented an incremental NDN smart grid architecture design for meeting the needs of the smart grid applications.

To enable preferential treatment of the applications communications, in this paper, we further extend the architectural design and proposed iCAAP–a QoS-aware ICN architecture, to meet the latency, bandwidth, and reliability requirements of smart grid applications, which operates in a data/information-centric infrastructure. In designing iCAAP, we categorized the smart grid traffic into three classes based on attributes: *(i)* protection traffic, which requires low latency and high reliability, *(ii)* control traffic that needs high reliability, and *(iii)* best-effort traffic which does not impose stringent requirements (accounting for all other applications). Such a categorization in iCAAP allows us to assign a unique priority class to each traffic flow for it preferential treatment to meet its QoS needs. Our simulation results demonstrated that iCAAP can achieve higher reliability with lower communication overhead compared to existing communication architectures; particularly iCASM [10].

Contributions: Our key contributions include: a) the design of iCAAP, a more comprehensive, QoS-aware networking architecture for preferential handling of various smart grid traffic. The architecture includes multiple queues in a node to handle the traffic classes and a weighted fair queuing scheduler to provide QoS-aware traffic handling. b) Extending

+ are with the Department of Computer Science, New Mexico State University, Las Cruces, USA, e-mail: {anju,gtorresz,sharad,misra}@nmsu.edu. † is with the Computer Science Department of St. Louis University, St. Louis, USA, e-mail:{reza.tourani}@slu.edu.

ndnSIM [11], the *de facto* NDN simulator to implement the architecture for public release. c) We also present a simulation-driven scalability assessment of iCAAP strategy and its comparison with prior NDN-based implementations.

The rest of the paper is organized as follow. In Section II, we review the existing literature in ICN-based smart grid communication. In Section III, we discuss the design of our QoS-aware strategy. We present our experimental results in Section IV and draw our conclusion in Section V.

II. Background and Preliminaries

A. Related Work

It has been argued in the literature that IP-based network architectures can not effectively satisfy the communication requirements of smart grid applications, such as distributed energy markets and real-time communications [1], [2]. In this paper, we focus on proposed QoS techniques in the ICN domain. The state-of-the-art in QoS-aware communications in ICN use various techniques, such as namespace prioritization, Network Calculus (NC), colony optimization, and software defined networking (SDN). The authors in [12], [13] suggested the use of data naming for DiffServ. Priorities were assigned to packets using hierarchical names and prefixes, which are mapped using longest prefix match. Tourani *et al.* proposed an overarching smart grid networking architecture (iCenS) that operates on top of NDN architecture [3]. iCenS addresses the demanding smart grid communication requirements and challenges, including scalability, backward compatibility with TCP/IP, and network inter-operability. McCarthy *et al.* proposed a method to improve QoS in ICN based on data delivery deadline awareness [14]. The priority classification of request and data packets were made using QoS information encoded in the packets. Wang *et al.* proposed QoS-predicted energy efficient routing (QPER), which used network calculus (NC) to predict energy efficient, and QoS-enabled routing [15]. In [16], the features of CCN and SDN were combined to offer programmatic interface for tuning networking intelligence based on application requirements.

In the ICN domain, QoS requirements are most commonly addressed by customizing forwarding strategy. Abdelaal *et al.* proposed a combined forwarding and caching strategy (called liteNDN) to make decisions on data caching [17]. The authors claimed that liteNDN can significantly reduce Round Trip Time (RTT) and bandwidth usage compared to conventional NDN framework. Others, [13], [18]–[20], also addressed QoS issues using forwarding strategy and evaluated the performance based on quality parameters, such as RTT, bandwidth, and cost.

We note that none of the existing QoS proposals have considered rate-controlling of the traffic flows, which is essential to provide prioritized access to the urgent flows, while rate-limiting non-urgent ones. This is especially important when the network is congested. In this paper, we address this issue by proposing iCAAP. We use different queues for different priority classes and further control the rate of various traffic flows using the token bucket scheme, while providing fairness to flows using weighted fair queuing.

B. Named-Data Networking Overview

Named-Data Networking (NDN) [21] architecture employs a name-based data delivery model in which uniquely named data can be retrieved by the name. To guarantee provenance and data integrity, data producers should digitally sign each data piece upon their generation. Moreover, NDN features a forwarding strategy layer, which enables each entity to make flexible packet forwarding decisions (e.g., multicast and broadcast). NDN includes two types of packet–request and data. NDN's inherent flow control enforces that each request elicits a piece of data.

In NDN, each network entity is equipped with a content store, a pending interest table (PIT), and a forwarding-information base (FIB). The forwarding-information base has a similar functionality to the existing routing table in IP networks. The pending information table keeps track of in-flight requests and enables NDN's stateful forwarding and request aggregation. The content store acts as temporary cache to store the popular data packets, enabling the intermediate entities to satisfy the requests for cached data.

When a network entity receives a named request, it performs a lookup using the requested data name in the cache and returns the data if cache hit happens. Otherwise, it aggregates the request if there is a pending request for the same data in the pending interest table (the request will be dropped and the incoming interface will be stored for reverse path forwarding of the data). Otherwise, the network entity forwards the requests towards the data provider and store the states in the pending interest table as a means to return the data back to the requester.

III. iCAAP Design and Implementation

As mentioned earlier, data naming and forwarding strategy two unique features of NDN. In designing iCAAP, we utilize these features to promote traffic classification and preferential traffic treatment to achieve the expected communication QoS. In this section, we elaborate on iCAAP's building blocks–namespace design that can be generically applied to various smart grid application and our multi-queuing mechanism–and then discuss our QoS-aware strategy.

A. Namespace Design

In NDN, data names follow a hierarchical, human-readable convention similar to the existing *url* addresses. To retrieve a named piece of data (called data chunk in NDN), a requester has to send a request (interest)–containing the data name–into the network. The intermediate entities perform longest prefix matching on the data for packet forwarding. In our vision for smart grid communication, the entities generating data (e.g., PMUs) send their generated data through payloaded requests. Payloaded interests are NDN interests that carry data payloads.

To illustrate our namespace design, consider PMU_1 (Fig. 1) that needs to send the generated sensory information in form of a payloaded interest (with the payload size of 200 bytes) to the corresponding PDC. Since PMU_1 is part of IEEE-39 bus system, it can generate a payloaded interest with the name "`/IEEE-39/Priority_Class/PDC/measurement`

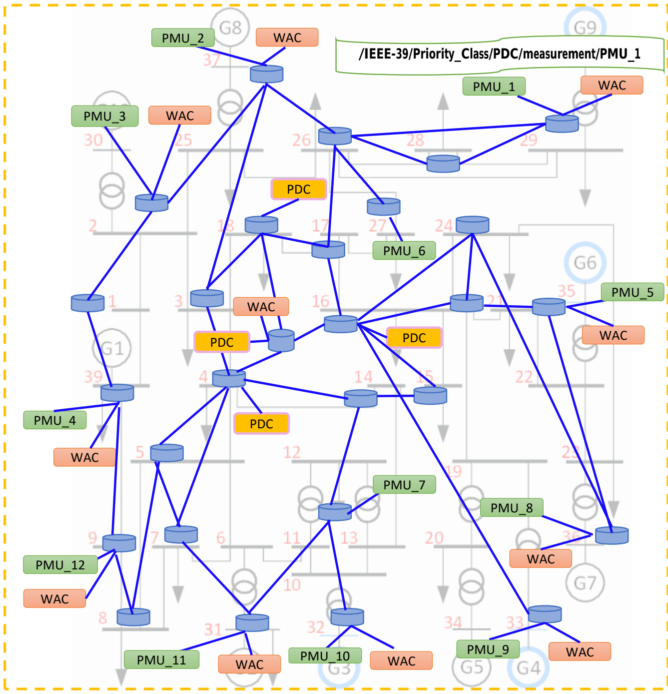

Fig. 1: IEEE-39 bus system along with the communication topology. We use this topology for our simulations.

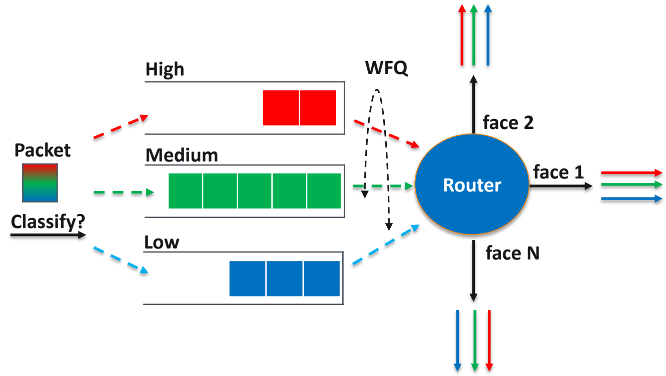

Fig. 2: Different priority classes and the WFQ Queueing Discipline running on each node.

`/PMU_1`" and send to the network. This expressive naming allows the network to forward the packet to any of PDCs that belong to IEEE-39 bus system while taking the priority class of the packet into account. Once the PDC receives the packet, it responds with an acknowledgment of receipt. Such an acknowledgment might not be necessary from the application standpoint. However, from the networking perspective, replying with the acknowledgement is necessary for the routers to release the PIT entry.

In iCAAP's design, we use the NDN's naming feature to categorize applications traffic into three priority classes for preferential treatment of traffic. In particular, the first class (Type I) includes the traffic flows directed to the PDCs ("`/IEEE-39/TypeI/PDC`"). We consider this class of traffic as high priority protection and control signals. The second class (Type II) includes the traffic flows to the WACs ("`/IEEE-39/TypeII/WAC`"), which are considered as the medium priority for control and information signals. The last class (Type III) includes other traffic flows, such as background and congestion information ("`/IEEE-39/TypeIII/BGD`"), which will be treated as low priority traffic. In iCAAP, these priority classes are part of the packets names. Thus, enabling the intermediate routers to treat packets based on priorities.

B. Multi-Queuing Mechanism

In order to utilize the full potential of our priority-based packet forwarding in iCAAP, we envision each network entity to leverage multiple input queues per interface–one for each priority class. The rationale behind using a dedicated queue per priority class is to prevent the traffic with high rate and low priority from filling up the shared queue, causing the high priority traffic to remain in the queue or even being dropped from a saturated queue.

As shown in Fig. 2, we implemented a queuing mechanism such that each node will be assigned three input queues one output queue per interface. All the packets (i.e., request and data) are classified into one of the three priorities classes based on the differentiated service code point (DSCP) value that we encoded into the naming convention. The packets, upon arrival to an interface, will be classified and enqueued based on their priority classes.

In iCAAP, packets are processed using the weighted fair queuing (WFQ) discipline. In WFQ, incoming traffic flows will be assigned with weights proportional to their priority classes. When a packet is processed by an input queue, our QoS-aware strategy selects an outgoing interface (multiple outgoing interfaces for high and medium priority traffic classes) and forwards the packet towards the destination through the selected interface(s). To regulate traffic and ensure QoS, we use the token bucket algorithm for each queue with different token generation rate corresponding to the priority class of the queue. In the next subsection, we will elaborate on our strategy and the token bucket framework.

C. QoS-aware Forwarding Strategy

NDN's strategy layer, one of its unique features, allows network entities to make more informed and fine-grained packet forwarding decisions. NDN supports a wide range of forwarding strategies, including best-route, multicast, and multiRAT [9]. In this paper, we design the QoS-aware forwarding strategy that uses a combination of multicast and unicast packet forwarding models. The primary objective of using such a hybrid forwarding model is to satisfy the diverse requirements (e.g., latency, bandwidth, and reliability) of traffic classes. iCAAP also employs the token bucket algorithm, which along our priority classes, prevent low priority traffic flows from exhausting all the network resources.

In our design, we deploy the token bucket algorithm at each router in the network. As shown in Fig. 3, each queue will be associated with a token bucket that has a fixed token capacity (b) and a token generation rate (r) per unit time. We assign different token generation rates for different queues

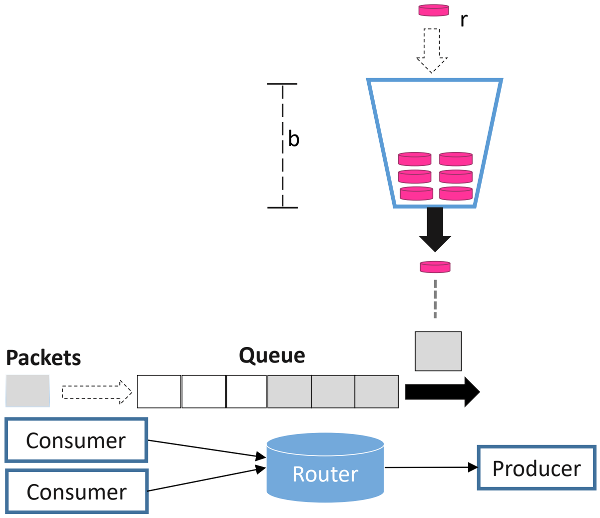

Fig. 3: The illustration of our architecture.

based on their priority classes–the highest token generation rate is assigned to high priority queues and the lowest token generation rate is assigned to low priority queues. Thus, the total number of tokens that will be added to each bucket during every S unit of time will be $r \times S$. In the case that the bucket is full–the rate of token generation is higher than token consumption–additional generated tokens will be dropped from the bucket.

To forward a packet, our QoS-aware forwarding strategy first dequeues a packet from one of the three queues of an interface using the WFQ algorithm (as mentioned in Subsection III-B). The strategy, then, checks the corresponding bucket for a token. Finally, it forwards the packet towards the outgoing interface(s) if at least one token is available in the corresponding bucket. Packets will not be dequeued from a queue when there are either no available tokens, or the link is currently full. A packet will dequeued and forwarded as soon as a new token is generated.

IV. Performance Evaluation

A. Simulation Setup

We used ndnSIM 2.7–a module of ns-3 network simulator–to assess the scalability and effectiveness of iCAAP. We simulated a communication network derived from IEEE-39 bus topology, including 27 routers, 10 WACs, 12 PMUs, and 2 PDCs. We allocated 10 source nodes for Type I traffic (protection PMUs communicating to the PDCs), 8 source nodes for Type II traffic (communications to WAC nodes), and 2 source nodes for Type III (background) traffic generating a total of 5 flows. Packet generation rates for Type I, II, and III were set to 90 packets/sec, 150 packets/sec, and 300 packets/sec respectively. Token bucket capacity for all traffic types (Type I, II, and III) were set to 1000 tokens. The token generation rates were set to 2500, 2000 and 800 tokens per second for Type I, II and III traffic, respectively.

We set the size of the transmission queues to hold a max of 50 packets in our simulations. In iCAAP we use the base ns-3 queue along with our priority queues, thus we set the packet limit for the base ns-3 queue to 20 while our priority queues were set to 10 each, for a max total of 50 packets. The payload size for Type I & II requests were set as 200 bytes and Type III requests were set to 1024 bytes. Each simulation was run for a duration of 300 seconds without packet re-transmission. In our analysis, we compared iCAAP with the baseline NDN (using best-route forwarding strategy) and iCASM [10] in terms of communication latency, packet loss, and overhead.

B. Results and Analysis

In terms of communication latency, as it is shown in Fig. 4, iCAAP outperformed iCASM and the baseline NDN across all traffic classes. For Type I and II traffic classes, Baseline NDN performed similar to iCAAP–with both performing better than iCASM. However, for Type III traffic, iCAAP showed more improvement over both baseline NDN and iCASM. We attribute the increased latency of iCASM to the overuse of packet multicasting for all traffic, which also results in unnecessary congestion in the network and communication overhead.

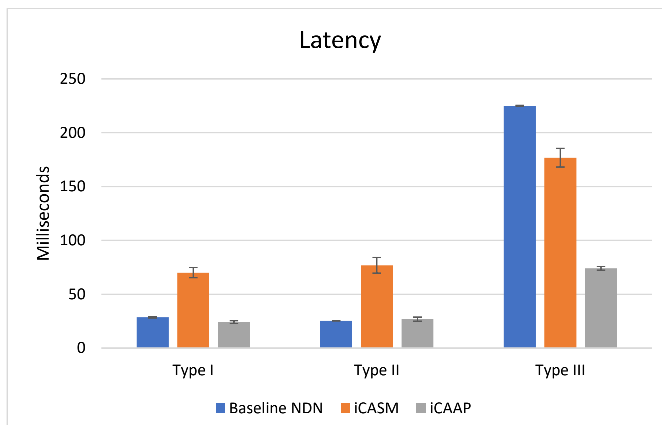

Fig. 4: Averaged latency measurements across all types.

Fig. 5 presents the loss rate results of all strategies across all traffic classes. In this regard, we observed that iCAAP achieved the lowest packet loss rate for Type I and II traffic followed by iCASM and Baseline NDN. For Type III traffic, however, Baseline NDN showed improvement over iCAAP and iCASM; also, iCAAP outperformed iCASM. The rationale behind the lower loss rates of iCAAP and iCASM for Type I and II traffic is that iCAAP uses controlled multicast (using token bucket) and iCASM uses multicast for high priority traffic while Baseline NDN uses best-route (unicast). However, all strategies use unicast for Type III traffic, which results in higher packet loss for iCAAP and iCASM as they give higher priority traffic classes more availability.

We compared the total traffic transmitted in the network in baseline NDN, iCASM, and iCAAP (refer Fig. 6). By design, the baseline NDN, with best route, sends the least amount of packets, which is why for Type I & II flows iCASM and iCAAP have significantly more packets. In fact, iCASM and iCAAP send more new packets than the baseline as there are less packets that are dropped and need to be resent. The real

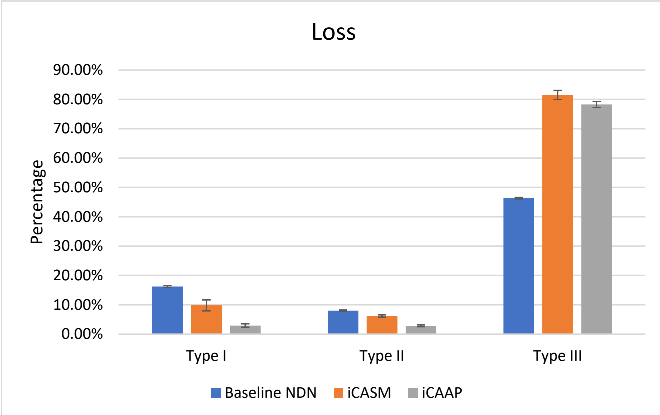

Fig. 5: Packet loss rate measurements shows the effectiveness of iCAAP in treating high priority traffics (Type I and II).

overhead of Type I and II flows w.r.t the baseline are: $\sim 104\%$ and 101% for iCASM and iCAAP and $\sim 85\%$ and 81% for iCASM and iCAAP respectively. More interestingly, iCASM and iCAAP send much less Type III traffic than the baseline, with the same application quality. This is because with the preferential treatment the network is not as congested, thus allowing more best effort traffic as well.

Consider grid stability management using data from PMUs and PDCs and corresponding control actions (transmitted to the WACs), iCAAP with its better packet delivery rates and low latency would result in faster convergence.

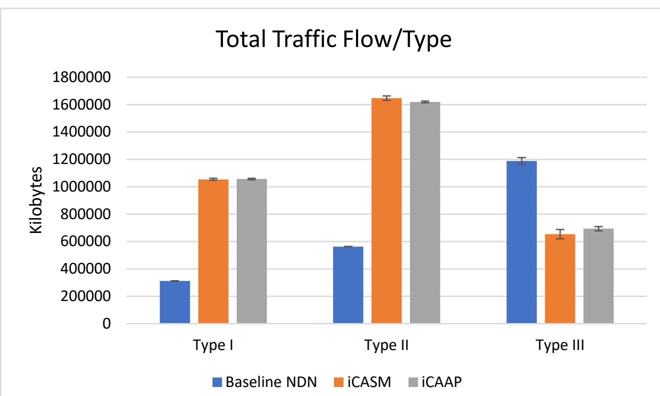

Fig. 6: The total traffic flow/type across three architectures.

V. Conclusion

In this paper, we presented a QoS-aware information-centric networking architecture (iCAAP), which is designed to prioritize smart grid communication to satisfy its stringent applications' requirements. In designing iCAAP, we envisioned three classes of traffic (i.e., high, medium, and low priority) with different characteristics. Using simulations, we demonstrated that despite the network being congested overall, iCAAP can satisfy the QoS expectation of high priority traffic.

In future, we will augment iCAAP by considering other network characteristics such as round trip time. We also plan to use machine learning models that can learn network behaviors and dynamically adjust token generation rates for different classes, aiming at reducing the network overhead.

References

[1] H. Liang, B. J. Choi, A. Abdrabou, W. Zhuang, and X. S. Shen, "Decentralized economic dispatch in microgrids via heterogeneous wireless networks," *IEEE J. Sel. Areas in Commun.*, vol. 30, no. 6, pp. 1061–1074, 2012.

[2] H. Liang, B. J. Choi, W. Zhuang, and X. Shen, "Stability enhancement of decentralized inverter control through wireless communications in microgrids," *IEEE Trans. on Smart Grid*, vol. 4, no. 1, pp. 321–331, 2013.

[3] R. Tourani, S. Misra, T. Mick, S. Brahma, M. Biswal, and D. Ameme, "icens: An information-centric smart grid network architecture," in *Int. Conf. on Smart Grid Commun.* IEEE, 2016, pp. 417–422.

[4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, 2012.

[5] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE commun. surveys & tutorials*, vol. 16, no. 2, pp. 1024–1049, 2013.

[6] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Computer Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.

[7] J. Zhang, Q. Li, and E. Schooler, "ihems: An information-centric approach to secure home energy management," in *IEEE Int. Conf. on SmartGrid Commun.*, 2012, pp. 217–222.

[8] R. Katz, D. Culler, S. Sanders, S. Alspaugh, Y. Chen, S. Dawson-Haggerty, P. Dutta, M. He, X. Jiang, L. Keys *et al.*, "An information-centric energy infrastructure: The berkeley view," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 1, pp. 7–22, 2011.

[9] R. Tourani, S. Misra, and T. Mick, "IC-MCN: An architecture for an information-centric mobile converged network," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 43–49, 2016.

[10] G. Ravikumar, D. Ameme, S. Misra, S. Brahma, and R. Tourani, "iCASM: An Information-Centric Network Architecture for Wide Area Measurement Systems," *IEEE Trans. on Smart Grid*, 2020.

[11] A. Afanasyev, I. Moiseenko, L. Zhang *et al.*, "ndnsim: Ndn simulator for ns-3," *University of California, Los Angeles, Tech. Rep*, vol. 4, 2012.

[12] C. Gündoğan, J. Pfender, M. Frey, T. C. Schmidt, F. Shzu-Juraschek, and M. Wählisch, "Gain more for less: The surprising benefits of qos management in constrained ndn networks," in *Proceedings of the 6th ACM Conference on Information-Centric Networking*, ser. ICN '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 141–152. [Online]. Available: https://doi.org/10.1145/3357150.3357404

[13] G. Ravikumar, D. Ameme, S. Misra, S. Brahma, and R. Tourani, "iCASM: An Information-Centric Network Architecture for Wide Area Measurement Systems," *IEEE Trans. on Smart Grid*, vol. 11, no. 4, pp. 3418–3427, 2020.

[14] J. McCarthy, P. Kuppuudaiyar, R. Loomba, and S. Clarke, "Towards an icn approach to qos in a dynamic edge environment," in *2019 IEEE Globecom Workshops*, 2019, pp. 1–6.

[15] K. Wang, J. Li, J. Wu, and G. Li, "Qos-predicted energy efficient routing for information-centric smart grid: A network calculus approach," *IEEE Access*, vol. 6, pp. 52 867–52 876, 2018.

[16] N. El Houda BenYoussef, Y. Barouni, S. Khalfallah, J. B. H. Slama, and K. Ben Driss, "Mixing sdn and ccn for content-centric qos aware smart grid architecture," in *IEEE/ACM 25th Int. Symp. on Quality of Service*, 2017, pp. 1–5.

[17] M. Abdelaal, M. Karadeniz, F. Dürr, and K. Rothermel, "litendn: Qos-aware packet forwarding and caching for named data networks," in *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, 2020, pp. 1–9.

[18] A. Kerrouche, M. R. Senouci, and A. Mellouk, "Qos-fs: A new forwarding strategy with qos for routing in named data networking," in *IEEE Int. Conf. on Commun.*, 2016, pp. 1–7.

[19] I. Araujo, A. Silva, A. Klautau, and N. Linder, "Qos in forwarding strategies for icn: New algorithm and experimental evaluation," *J. of Commun. and Information Systems*, vol. 34, pp. 275–286, 2019.

[20] A. Kerrouche, M. R. Senouci, A. Mellouk, and T. Abreu, "Ac-qos-fs: Ant colony based qos-aware forwarding strategy for routing in named data networking," in *2017 IEEE Int. Conf. on Commun.*, 2017, pp. 1–6.

[21] L. Zhang *et al.*, "Named-data networking," *ACM SIGCOMM Computer Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.