Secure Real-Time Industrial IoT Communications in Smart Grids Using Named Data Networking

Henry Hui, James Grant, Kieran McLaughlin, David Laverty, Sakir Sezer
Queen's University Belfast
Belfast, United Kingdom

***Abstract*—This paper explores Named Data Networking (NDN) for secure Industrial IoT (IIoT) communications in smart grid applications. NDN is a next generation networking paradigm, which is data-centric and has the benefit of built-in security properties, such as data integrity. This work applies NDN to IEEE C37.118.2 PMU communications, as an example smart grid IIoT application, and proposes a new data-encapsulation approach for NDN for low latency data streaming. The proposed communication architecture allows sensor data streaming with a lower overhead compared to related work. Communications are demonstrated to be secured using a trust anchor which protects data integrity and provides data authentication, while supporting optional data encryption. The proposed solution represents IEEE C37.118.2 in a JSON format, which provides flexibility and facilitates application of the approach to different use cases.**

***Index Terms*-- Named-data networking (NDN), Phasor Measurement Unit (PMU), Industrial-IoT (IIoT), Security**

I. INTRODUCTION

Many SCADA protocols evolved from original serial links by adopting TCP/IP encapsulation. Such protocols are often heavily based on data objects, for example IEC 60870-5-101 uses Application Service Data Unit (ASDU) addresses, where data is classified into information objects, each provided with a specific logical address. Arguably this data model lends itself more naturally to a data-oriented communication approach, rather than TCP/IP, which is host-oriented. This approach also often requires various middleware and gateways to provide layers of security. This paper investigates using Named Data Networking (NDN), a new data-centric networking paradigm, for communication between Industrial IoT (IIoT) devices. Currently, NDN research is performed, for example, on a globally connected testbed hosted in different research institutions, and remains in the very early stages for industrial applications [1]. This paper considers the potential benefits if NDN is applied to communications in smart grid applications that use IIoT devices, particularly regarding provision of cyber security directly at the network layer (equivalent to the IP layer in TCP/IP). We do this by considering a critical component of smart grids, the Phasor Measurement Unit (PMU). Currently PMU communications typically use TCP/IP and standards such as IEC 61850-90-5 and IEEE C37.118.2 [2]. Most of these standards are generally implemented without security, so data streamed from devices often requires security implemented at the application layer or use of VPNs. However, there are costs, for example additional overheads above the TCP/IP layers [3]. This paper therefore presents a proof-of-concept implementation of real-time PMU data streaming based on NDN, with security built-in at the network layer. This paper is structured as follows: Section II provides an introduction to PMU and NDN; Section III highlights the key differences between IP and NDN, along with a comparison of the threats faced by both paradigms; Section IV demonstrates the NDN interest packet streaming; Section V summarises the paper and provides future research directions for this work.

II. BACKGROUND AND RELATED WORK

This section provides the necessary background for this work, including a brief introduction of PMUs and NDN, the existing proposal of data streaming using NDN, and how security is integrated in the design of NDN.

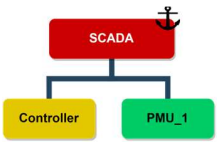

Fig. 1. Simple NDN trust hierarchy

A. Phasor Measurement Unit

Phasor Measurement Units (PMUs) are a key enabling technology of smart grids [4]–[6]. A PMU is an instrument which measures the voltage and current waveforms of an electrical power system, synchronised to a global time standard such as GPS. This allows unprecedented visibility of electricity networks for systems operators and supports, for example, the connection of renewable energy generation to the power grid. Detailed descriptions of PMUs can be found in various literature and are not repeated here, however, this section summarises some of the challenges for the wide adoption of PMUs [4]. The IEEE C37.118 synchrophasor standard originally requires a data rate of 30 frames/s. However, to fully harness the potential capabilities of the latest technology, an increased data rate, for example, at a minimum of 120 Hz will be utilised in this work. Although more information about NDN

will be given in the next section, related work considers the possibility of sending PMU data using 30 to 120 packets/s [7]–[9], while measuring network performance metrics like latency and packet loss. It has been shown that, in both normal and lossy network environment, a better performance is observed when using NDN instead of IP. However, [8], [9] required a request per sample of PMU data, while [7] utilised a simulated network and testing was not performed on physical devices. Moreover, there is a general lack of information regarding security in the related work. Some PMU standards such as IEC61850-90-5 refer to security specifications, but these are rarely used in practice. One of the problems for security is that the PMU application developer must consider the security implementation. Where security is found, it is often via VPN [10]. A further weakness in current arrangements for PMU is the use of Phasor Data Concentrators (PDCs), which requires the unpacking, refactoring and repackaging of PMU data in the communication path [11]. This creates additional overheads and latency (delay), and from a security perspective introduces a possible point of cyber-attack in the IT network. Hence, a key motivation for this work is that an application developer should be able to simply use NDN libraries without needing to directly consider security implementations, with the network layer providing security features to the application.

B. Named Data Networking

NDN emerged in 2010 as one of five projects funded by the US National Science Foundation under its Future Internet Architecture Program. The ongoing NDN project investigates the proposed evolution from today's host-centric network architecture (IP) to a data-centric network architecture (NDN). Recent advancements of NDN are facilitated by a cooperative global testbed. Research on data routing, applications, network performance, and networking tools are typically developed on the testbed. In the presented work we utilise an offline testbed to facilitate security analysis of the proposed communications. NDN is also proposed for application in other areas of industrial applications, for example [12] uses NDN to provide security as an overlay network over the existing IP infrastructure. Communication in NDN is driven by data *consumers*, through the exchange of two types of packets, Interest and Data, between nodes in the network. Both packet types carry a *name* that identifies a piece of data. Although the physical connection of a NDN network is arbitrary, the network is organised logically in a hierarchical structure, for example in the naming of the nodes and the *trust* of the network. The word "*trust*" in this paper refers to the chain of trust, which means all the cryptographic keys used in the network can be derived from the *trust anchor*. The trust anchor is the only node in the network where its keys are established locally. Fig. 1 illustrates the network hierarchy of the network being tested. This will be further addressed in section II.D.

In terms of naming, the naming hierarchy can be for the local network or the global internet. For example, for a historian in a SCADA network, '/SCADA/historian/20230101' could be the name used for some data logged on 1 January 2023. The name of the data can be broken down into chunks, such as '…20230101/1', '…20230101/2', and so on, while "/SCADA/" is the NDN prefix of the host historian. For a consumer to indicate interest in specific data held somewhere

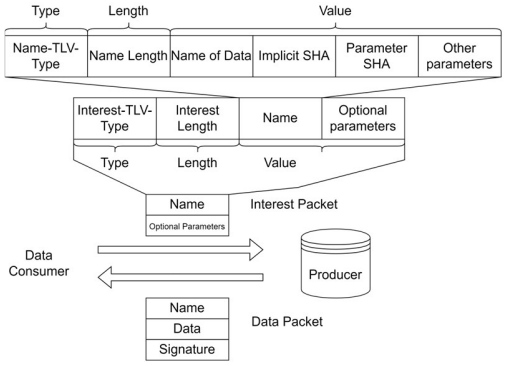

Fig. 2. NDN Interest and Data packet exchange and the default Type-Length-Value (TLV) of an interest packet.

in the network, the consumer sends an Interest packet to the network with the name of the desired data. This basic exchange is illustrated in Fig. 2, which also shows the Type-Length-Value (TLV) construction of the default Interest packet and the data name. Every NDN packet is encoded in the TLV format. Multiple TLV blocks can be encapsulated in a single packet. To enable a host or router to send, receive or route NDN traffic, the NDN Forwarding Daemon (NFD) is required. NFD uses a *face*, a more generalised term for a network interface, to send and receive data. This allows NDN traffic to be carried via Ethernet, or even as an overlay network on top of IP network infrastructure – in this work we use Ethernet. The literature covers standard approaches to route data [13]. In summary, upon receiving an Interest packet from a consumer, a router will query its NFD for any record in the cache of the same Interest packet (that was sent by other nodes but is yet to receive data). This is called a Pending Interest, which allows a reduction in the number of packets being forwarded. If there is no existing Pending Interest, the NFD looks in the data cache for the previously satisfied Interest. If a match is found, this data will be the sent to the consumer. Otherwise, the Interest is forwarded to the producer. The NFD will then wait for replies from the producer, before forwarding the data onward to the consumer.

On applying NDN in the IIoT use case, there are a couple factors that require consideration, namely how push-based data is handled and how data is streamed. In the context of this work, push-based data can be interpreted as the data that the data producer sends out to consumers that have previously registered an interest. The NDN project has suggested the use of either data-encapsulated interest packets or "long-lived interest" packets. This work proposes the former method, and the details can be found in section IV. A data-encapsulated interest packet allows data to be included in the *optional parameters* field of the interest packet, or to append the data to the end of the *name* of the packet. A "long-lived interest" packet allows the request of data until a specific time, or when the interest is expired. On top of the two methods mentioned by the NDN project, two further methods have been described, but are considered out of scope for this paper [14]. On the issue of streaming, prominent work on NDN data streaming belongs to two categories: real-time media streaming and multi-host data synchronisation. In each category it appears that existing methods always require an Interest packet per data request (thus many packets per sample). Additionally, methods related to real-time media

streaming assume data producers have all data already available. This is not the case for IIoT sensors like PMUs, where data samples are continuously created. Therefore, we focus on work related only to data synchronisation, which aims to update multiple nodes in the network with the latest data available. Moreover, a majority of state-of-the-art research on NDN and its code base are related to the core networking functionalities, like routing protocols and performance enhancement. However, to harness the full potential of NDN, research must explore diverse and realistic use cases. From the perspective of smart grids and IIOT, legacy IP based solutions have performance shortcomings and suffer TCP/IP related security issues. This work addresses the latter problem by resolving gaps in streaming secure sampled data using NDN.

C. NDN Data Synchronisation

Related work on data synchronisation includes ChronoSync [15], Dataset Synchronization protocol (DSSN) [16], State-Vector-Sync (SVS) [17], and Prefetch Loss-Insensitive sync (PLI-Sync) [18]. ChronoSync was an early attempt, studied in the context of Internet Relay Chat (IRC), where it is a good alternative for multicast synchronisation since NDN can broadcast interest to all participants. DSSN was proposed to address ChronoSync overhead issues, caused by nodes pushing updates to all other nodes when new data is available. DSSN distributes data as a dataset, or vector, and consumers only express interest in data from specific nodes. SVS removes DSSN features that were tailored for lossy wireless environments, and instead assumes the network is robust. SVS participants organise in sync-groups and are informed whenever new data is published by using sequence numbers, which are published as a state-vector in a NDN interest broadcast. Nodes can request the newest data (with the highest sequence number) by sending interest to a corresponding participant. However, a node must wait for the latest state-vector to be published before data can be requested. PLI-Sync improves this by allowing nodes to assume new data is available and issue interest before receiving the state-vector. This reduces the need for state-vector broadcast packets. However, all the mentioned methods lack security evaluations and related details, despite it being one of the major benefits of NDN. Therefore, the next section discusses the security measures that NDN can support.

D. How NDN Provides Security

NDN security is built into the data packets themselves. Each piece of data is signed by the producer, securely binding them. The intention is to secure the content, not the container or communication channel, which is what solutions such as VPNs aim to achieve. Integrity protection guarantees the authenticity of the data bound to the name by including the producer signature of the data plus its name. There are a few supported signature types [19], of which SHA256 with RSA is utilised in this work. The signatures of data packets (and signed interest packets) are created using the key distributed by the trust anchor, where only an authorised node can receive a key. A trust model resembling a hierarchy of the network is generally adopted in NDN (see Figs. 1 and 2): The *Controller* can obtain a certificate of the *PMU_1* using the name: "/SCADA/PMU_1/KEY/ghi789"; similarly, the certificate for *PMU_1* can be retrieved from "/SCADA/Controller/KEY/def456", where "ghi789" and "def456" are the name (identifier) of the key. The trust anchor in this network is "SCADA", where the root certificate is published in "/SCADA/KEY/abc123" and other keys and certificate in the network will be based on this published certificate. To authenticate the node sending the data packet (or the node requesting data with a signed interest packet), an authorised receiver node can request the public key from the trust anchor to verify the identity of the sender. Unauthorised users will not be able to obtain the key for signing or verifying. If the signature is not verified, the receiver will drop the data packet. In practice, keys and certificates can be exchanged and signed using *ndnsec*, a tool that accompanies the NFD. This can also be done programmatically, using the ndn-cxx library (this is the standard NDN C++ library). While signatures are mandatory, data encryption is optional, and applications can distribute data encryption keys as encrypted NDN data. NDN Name-based Access Control (NDN-NAC) [20], provides a unified mechanism to sign and encrypt NDN data by applying the trust schema mentioned previously. To manage the signing keys and encryption keys separately, all the names in the original namespace are reallocated in the "read" and "samples" namespace respectively. For example, for a data producer named "/SCADA/PMU", the data is published and signed with the name "SCADA/samples/PMU" while the name "SCADA/read/PMU" will be used to handle the distribution of keys. The security mechanisms implemented for NDN are often well tested in the IP environment. For example, [21] demonstrated applying a Kerberos-based key-exchange protocol to secure an industrial protocol named FF-HSE used in fieldbus networks. However, there are some addition considerations when applying security for PMU communications that we will explore in this paper.

III. Comparison of IP and NDN for PMU Communications

There is a wide array of literature regarding inherent security issues of IP-based communication, and multiple papers investigating such threats in relation to IP-connected PMUs. [22] presented a way to categorise attacks against PMU communications as: 1) Interruption, 2) Interception, 3) Fabrication, and 4) Modification. In the context of the solution presented in this paper, NDN offers inherent protection against points 2, 3 and 4, at the network layer.

For networks using TCP/IP there are some fundamental and inherent vulnerabilities in how these protocols operate that are ubiquitous and challenging to completely mitigate. For example, IP address spoofing at the IP layer, or abusing TCP flags to create SYN floods, or to scan a network to identify hosts and services. For NDN, the equivalent of IP address spoofing is not possible because packets are signed, and hosts cannot be scanned because network communication is based on data names rather than host addresses. Considering forms of attack that intercept PMU communications, both IP and NDN are susceptible to network sniffing once a device in the network is compromised. Therefore, in the IP domain, an encrypted channel is often deployed to protect against sniffing, however this is often realised using an application layer solution, e.g. a

VPN. Meanwhile, NDN (NDN-NAC) provides encryption for every data packet natively. A more active attack that attackers might perform is a Man-in-the-middle (MITM) attack in an IP network, where the attacker tricks the endpoints to route the packets to them. The attacker will then either observe the information or modify it, before routing the packets back to the intended endpoints. However, unless a device with existing credentials is compromised, this attack is not possible in NDN since MITM requires the adversaries to send packets to the endpoints, where the data signature is verified by the trust anchor [23]. More information will be given in section IV.B.1). Fabrication attacks typically build upon any of the above attacks, such as using a MITM attack to capture and modify data in transit. This is where data encryption is essential. As highlighted above, this is typically achieved in many industrial systems via VPN solutions, or in NDN with NAC. It is worth noting that NAC provides an added benefit where access control is performed along the exchange of encryption keys. Furthermore, VPNs are often implemented at a gateway point in an IP network, behind which packets are not encrypted or authenticated, while in comparison, NDN provides these security mechanisms fully end-to-end between devices. Other than directly intercepting PMU communication as mentioned above, PMU data modification can also be done indirectly by targeting PDCs. Phasor Data Concentrators, or PDCs, are devices that aggregate data from a number of PMUs. Note that scanning such devices or hosts is much more difficult in NDN networks because NDN is not based on connecting hosts end-to-end, but is based on a node making requests to the network for data. Network owners can apply optional interest signatures to support authentication.

IV. IMPLEMENTATION

Having considered the issues for push-based data, streaming, and security configurations, this section will explore a test-bed implementation using NDN to support live data streams from PMUs to a controller station, where the PMU is outputting data according to the C37.118.2 standard.

A. *Networking and Instrumentation*

Two test networks have been set up. The first one is a simple hardware setup that involves a generic low-power PC and two embedded controllers, which realise an NDN router and the PMUs. Additionally, a set of docker containers running Ubuntu servers 20.04 are configured to realise a virtualised NDN network. This facilitates flexible and scalable testing. In both setups, NFD is run on all nodes to enable the handling of NDN traffic. Currently, the NDN face and routes are set up manually using the *nfdc*, a command line tool to manage a running NFD. The communication was verified using test NFD traffic which can be generated by tools such as *ndnping* (a NDN version of the ICMP ping command). The test setups communicate using NDN on Ethernet, using the MAC address of the network interface.

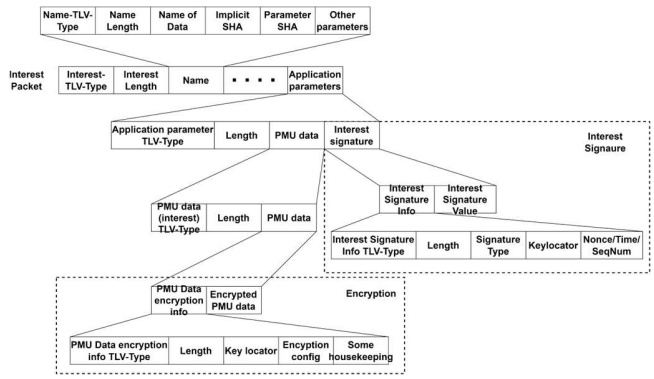

Fig. 3. Packet structure for the proposed PMU communication

B. *Proposed Data Streaming Architecture*

A communication architecture is now proposed, which involve three phases, authentication, control and data streaming. To reduce the overhead compared to the options noted in section II.C, the architecture handles data streaming using data-encapsulated interest packet. This is especially important in the IIoT use case where data is transmitted in relatively small packets, and where latency is of concern. Compared to some of the aforementioned methods, the information provided by the state-vectors is not necessary for the purpose of real time PMU data collection. Furthermore, in the proposed approach, control packets exchanges and data-encapsulated interest packets will be signed and can optionally be encrypted. As shown in Fig. 2, there are three nodes in the network being tested, the controller, PMU_1 and SCADA. Fig. 3 illustrates the proposed structure of a typical packet containing streaming data. The general details of different NDN fields are provided by [13]. Specific to our proposed architecture, the PMU data resides in the application parameter field. If the data is encrypted, the original PMU data will be replaced by the encrypted data. The interest signature is located at the end of the application parameters field. If the receiver wishes to verify the signatures (and to decrypt the packet when applicable), the key locator provides the *name* of the trust anchor, where the keys can be obtained by authorised nodes.

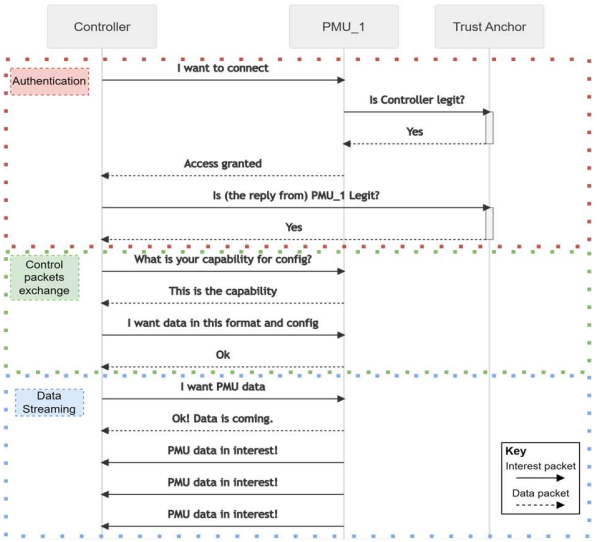

Fig. 4. Three phases for the proposed PMU communication

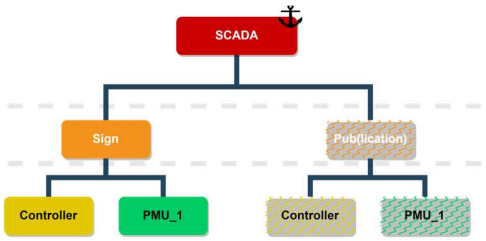

Fig. 5. Expanding trust hierarchy

The proposed three phases for the PMU communication (authentication, control packet exchange and data streaming) are shown in detail in Fig 4. For the rest of this section, the related security details and a description of the authentication phase are given, followed by the operational details of the architecture, along with a description of the remaining two communication phases in control and data streaming.

1) Security Configurations

The authentication phase handles the key establishment process for authorised hosts. In these experiments, it is assumed that the authorisation of different nodes had been performed separately prior to the sensor data communications. This information and other details of NDN trust relationship can be found in [23]. In the context of this paper, a list of authorised nodes and the corresponding public keys is stored in the trust anchor (*SCADA*), and only these nodes can request keys for signing (or encrypting) the data. The trust anchor can authenticate users from the authorised list previously mentioned, before sending out keys for decryption and signature verification. In addition, to facilitate encryption and the corresponding key management, a separate namespace is allocated to the logical trust hierarchy, similar to part of the NDN-NAC specification [20]. As illustrated in Fig. 5, the trust hierarchy from Fig 1. is expanded. A logical namespace layer is introduced including the name "Sign" and "Pub", which hosts the keys for signature and encryption respectively. The trust hierarchy of "Sign" and "Pub" under the trust anchor are a mirror of each other. It should be emphasised that this logical architecture is only use for key distribution purpose and it does not add to the number of hosts in the network. To obtain the public key to verify the *Controller*'s signature, an interest packet can be sent to the name "/SCADA/Sign/Controller/KEY /def456". This information can be found in the Key Locator field of the packets sent by the *Controller*. To initiate the authentication phase in Fig. 4, the controller (or any node in the network) can send an interest packet to the PMU requesting the establishment of trust. The PMU (or any node in the network) can contact the trust anchor for the public key to verify that the signature in the interest packet is valid. The PMU can then reply to the controller with an authentication token, which the controller will check against the trust anchor, before replying with an acknowledgment packet. This implementation focuses on signature handling since it is built into NDN itself. Optional data encryption is part of ongoing research and will be addressed in future work as part of the complete security architecture for IIoT. However, to apply encryption, the *Controller* node will obtain its credential with "/SCADA/Pub/ Controller/E-Key/asdf12" under the Pub namespace.

2) PMU Streaming Operation

The control phase in the proposed architecture allows both communicating nodes to agree on the configuration, e.g., the length of a streaming session, credential expiration and anything use-case-specific. For the PMU use case, configuration information contains information like calibration factors, metadata and datatypes relevant to the IEEE C37.118 synchrophasor data format. This information is necessary for the data receiver to decode the data message. The Data streaming phase utilises the previously agreed upon configurations and established credentials for signing (and encrypting). Upon receiving the data request, *PMU_1* will reply to the *Controller* with the streaming data token, which can include any information relevant to the application. This work utilises JSON to carry the C37.118.2 data. JSON, or JavaScript Object Notation is a common open standard file and data exchange format that is widely adapted in IT applications, and its strong flexibility allows the utilisation of the proposed architecture to be easily adapted to other smart grid or IIoT protocols and applications. Fig. 6 shows C37.118.2 data carried via traditional TCP without security, while Fig. 7 shows JSON used to carry the data across the NDN network, including the secure data signature after the application parameters.

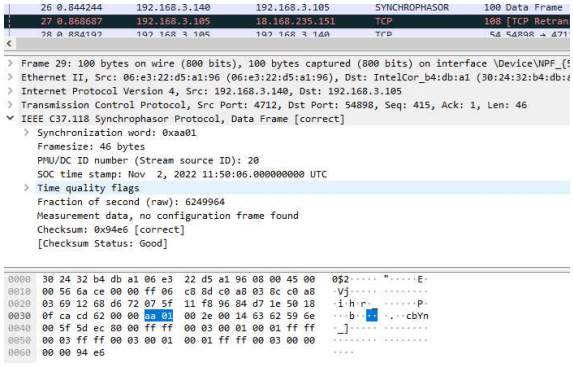

Fig. 6. IEEE C37.118 data in TCP packets

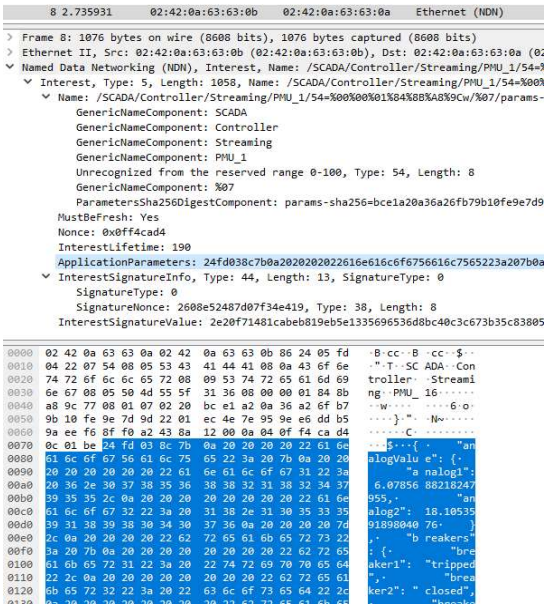

Fig. 7. JSON object being sent by NDN packets

C. Discussions

By default, NDN requires an interest packet for each piece of data. There is no built-in push-based communication, which is preferable for real-time IIoT sensor data. This means for streaming sensor data, 50% of the packets being sent in the network would be interest packets, which makes the overheads due to interest packets relatively high and adds latency to every data transmission. This overhead problem has been overcome with the proposed communications architecture described in Fig. 4. Specifically, under the default NDN settings, the length of an interest packet mostly depends on the length of the *Name* of the data being requested. For instance, recalling the example earlier, '/SCADA/historian/20230101', contributes to 25 bytes of data, without providing information of the current time, e.g. GPS, which is essential to PMU applications. Even if assuming a 50-byte interest packet, an overhead of about 6 kB/s can be reduced with the proposed interest-based data streaming technique for a 120Hz signal. Furthermore, the proposed approach involving compulsory signatures for every interest packet, which ensures all PMU data received is authenticated as being genuine and untampered – even without the use of data encryption. Additionally, to facilitate the optional implementation of data encryption, building on related work, a logical security namespace layer is introduced to handle signature and encryption keys respectively. Compared to related work [7]–[9] on using NDN for PMU communication, the proposed method provides security through the aforementioned mechanisms. Moreover, the network implementation has also been tested and validated using both a docker-based environment and physical devices.

V. Conclusion and Future Work

This paper proposes a new NDN communication architecture for streaming PMU data, as a replacement for IP in the network layer. A general comparison between IP and NDN is also presented. NDN provides built-in security at the network layer, greatly helping smart grid and IIoT application developers by providing security as a standard feature, rather than being an additional burden to be implemented by applications. Compared with previous research investigating NDN approaches for delivering PMU data, this architecture improves the security being provided, as well as reducing undesirable packets overheads by improving existing data synchronisation approaches. This mainly involves the utilisation of data-encapsulated interest packets to push PMU phasor data, after authentication and control packet exchange. Different to the related work, the proposed architecture mandates authentications and interest packet signatures. It has been demonstrated, using physical devices and docker containers, PMU data are being streamed across physical and emulated network. The PMU data originally in IEEE C37.118.2 format is carried in a JSON structure for flexibility, allowing easy redeployment across similar SCADA or IIoT protocols. Future work includes improvements to the deployment of a secured PMU and IIoT node. In the current set-up process a node requires manual setup of security and to utilise the NDN-NAC command line tool to setup the trust anchor, which is potentially open to error. Moreover, the introduction of security proxy and/or dual stack router will increase the flexibility and interoperability between existing IP infrastructure and NDN based networks.

References

[1] 'NDN Testbed', *Named Data Networking (NDN)*. https://named-data.net/ndn-testbed/ (accessed Jan. 29, 2023).

[2] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, 'Analysis of IEEE C37.118 and IEC 61850-90-5 synchrophasor communication frameworks', in *IEEE Power and Energy Society General Meeting,* 2016,

[3] K. Ghanem, S. Ugwuanyi, J. Hansawangkit, R. McPherson, R. Khan, and J. Irvine, 'Security vs Bandwidth: Performance Analysis Between IPsec and OpenVPN in Smart Grid', in *2022 International Symposium on Networks, Computers and Communications (ISNCC)*, Jul. 2022, pp. 1–5.

[4] C. Muscas and P. Attilio Pegoraro, 'Opportunities and Challenges for PMU Deployment in Distribution Systems', *IEEE Smart Grid*, 2014.

[5] Y. Hu and D. Novosel, 'Challenges in Implementing a Large-Scale PMU System', in *International Conference on Power System Technology*, 2006

[6] D. M. Laverty, J. Hastings, D. J. Morrow, R. Khan, K. Mclaughlin, and S. Sezer, 'A modular phasor measurement unit design featuring open data exchange methods', in *2017 IEEE Power & Energy Society General Meeting*, IEEE, 2017, pp. 1–5.

[7] Z. Hu, Y. Li, J. Wu, J. Guo, and H. Gu, 'Research of PMU data transmission mechanism in smart grid based on NDN', in *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, Nov. 2017, pp. 1–6. doi: 10.1109/EI2.2017.8245472.

[8] G. Ravikumar, D. Ameme, S. Misra, S. Brahma, and R. Tourani, 'iCASM: An Information-Centric Network Architecture for Wide Area Measurement Systems', *IEEE Trans. Smart Grid*, vol. 11, no. 4, Jul. 2020,

[9] D. Ameme, S. Misra, and A. Mtibaa, 'A Case for Information Centric Networking For Smart Grid Communications', in *Proceedings of the SIGCOMM Posters and Demos*, in SIGCOMM Posters and Demos '17. New York, NY, USA: Association for Computing Machinery, Aug. 2017

[10] M. Z. Gunduz and R. Das, 'Cyber-security on smart grid: Threats and potential solutions', *Comput. Netw.*, vol. 169, p. 107094, 2020.

[11] C. Tu, X. He, X. Liu, and P. Li, 'Cyber-attacks in PMU-based power network and countermeasures', *IEEE Access*, vol. 6, 2018.

[12] A. P. Plageras, K. E. Psannis, B. Gupta, C. Stergiou, B.-G. Kim, and Y. Ishibashi, 'Solutions for inter-connectivity and security in a smart hospital building', in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, IEEE, 2017, pp. 174–179.

[13] L. Zhang *et al.*, 'Named data networking', *Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014, doi: 10.1145/2656877.2656887.

[14] R. C. Sofia and P. M. Mendes, 'An overview on push-based communication models for information-centric networking', *Future Internet*, vol. 11, no. 3, p. 74, 2019.

[15] Z. Zhu and A. Afanasyev, 'Let's ChronoSync: Decentralized dataset state synchronization in Named Data Networking', in *2013 21st IEEE International Conference on Network Protocols (ICNP)*, 2013, pp. 1–10.

[16] X. Xu, H. Zhang, T. Li, and L. Zhang, 'Achieving Resilient Data Availability in Wireless Sensor Networks', in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2018

[17] P. Moll, V. Patil, N. Sabharwal, and L. Zhang, 'A Brief Introduction to State Vector Sync', *NDN Tech. Rep. NDN-0073*, 2021.

[18] Y. Hu, C. Serban, L. Wang, A. Afanasyev, and L. Zhang, 'PLI-Sync: Prefetch Loss-Insensitive Sync for NDN Group Streaming', in *ICC 2021 - IEEE International Conference on Communications*, Jun. 2021, pp. 1–6.

[19] Named Data Networking Project, 'NDN Packet Format Specification V0.1', 2013. https://named-data.net/doc/NDN-packet-spec/current/index.html (accessed Aug. 29, 2022).

[20] Y. Yu, A. Afanasyev, and L. Zhang, 'Name-based access control', *Named Data Netw. Proj. Tech. Rep. NDN-0034*, 2015.

[21] K. Hosoya and H. Miyata, 'Applying security architecture to industrial network protocol', in *2010 8th IEEE International Conference on Industrial Informatics*, IEEE, 2010, pp. 443–448.

[22] C. Beasley, X. Zhong, J. Deng, R. Brooks, and G. K. Venayagamoorthy, 'A survey of electric power synchrophasor network cyber security', in *IEEE PES Innovative Smart Grid Technologies, Europe*, 2014, pp. 1–5.

[23] Y. Yu, A. Afanasyev, D. Clark, K. C. Claffy, V. Jacobson, and L. Zhang, 'Schematizing trust in named data networking', in *proceedings of the 2nd ACM Conference on Information-Centric Networking*, 2015, pp. 177–186.