Intelligent and Converged Networks
2021, 2(1): 1–15
ISSN 2708-6240
DOI: 10.23919/ICN.2021.0001

Toward long-range adaptive communication via information centric networking

Anthony Dowling, Lauren Huie, Laurent Njilla, Hong Zhao, and Yaoqing Liu*

Abstract: As Internet of Things (IoT) applications become more prevalent and grow in their use, a limited number of wireless communication methods may be unable to enable dependable, robust delivery of information. It is necessary to enable adaptive communication and interoperability over a variety of wireless communication media to meet the requirements of large-scale IoT applications. This paper utilizes Named Data Networking (NDN), an up-and-coming Information-Centric Network architecture, to interconnect differing communication links via the network layer, and implements dynamic forwarding strategies and routing mechanisms which aid in the efficient dissemination of information. This work targets the creation of an interface technique to allow NDN to be transported via LoRa. This is acheived via the coupling of LoRa and WiFi using the NDN Forwarding Daemon (NFD) to create a universal ad hoc network. This network has the capacity for high range and multi-hop Device-to-Device (D2D) communication together with compatibility with other network communication media. Testing of the system in a real environment has shown that the newly created ad hoc network is capable of communicating over a several kilometer radius, while making use of the features provided by NDN to capitalize upon various links available to enable the efficient dissemination of data. Furthermore, the newly created network leverages NDN features to enable content-based routing within the LoRa network and utilize content-based routing techniques.

Key words: Named Data Networking (NDN); ad hoc network; device-to-device communication

1 Introduction

With the growing ubiquity of the Internet of Things (IoT) and its applications[1], such as precision agriculture and smart city deployments, becoming more ubiquitous and require larger-scale deployments, a limit on the wireless transmission media available to the network will restrain the capability of the network to deliver information reliably and efficiently. LoRa[2] is a wireless technology developed to enable low data rate communications over long distances. This transmission medium is capable of providing communications over a long-range (up to 20 km) while maintaining a low power usage (up to and exceeding ten years of battery life). These capabilities of LoRa to perform long-range transmission while using little power demonstrate that LoRa-enabled devices may be deployed in a variety of environments, both inside and outside of buildings, while maintaining the capability to communicate with peers and the network gateway. LoRa technology is uniquely suited for use in deployments, such as disaster relief and rescue, due to its capabilities to operate off-the-grid and at high ranges. LoRa allows for

• Anthony Dowling is with the Department of Electrical and Computer Engineering, Clarkson University, Potsdam, NY 13676, USA. E-mail: dowlinah@clarkson.edu.
• Lauren Huie and Laurent Njilla are with the Air Force Research Laboratory Information Directorate in Rome, Rome, NY 13441, USA. E-mail: lauren.huie-seversky@us.af.mil; laurent.njilla@us.af.mil.
• Hong Zhao is with the Department of Electrical and Computer Engineering, Fairleigh Dickinson University, Teaneck, NJ 07666, USA. E-mail: zhao@fdu.edu.
• Yaoqing Liu is with the Gildart Haase School of Computer Sciences and Engineering, Fairleigh Dickinson University, Teaneck, NJ 07666, USA. E-mail: yliu@fdu.edu.
* To whom correspondence should be addressed.
Manuscript received: 2020-11-27; accepted: 2021-01-13

easy onboarding of new nodes to a network to provide wide coverage by a network. In its typical configuration, LoRa is deployed in a hierarchichal "star-of-stars" wireless network topology that includes backhaul servers, gateway nodes, and LoRa end devices. One of the most widely used wireless networking technologies is WiFi. This transmission medium provides high-speed Internet and network connectivity. Many devices including personal computers, video-game stations, smartphones and tablets, digital cameras, smart TVs, etc. use WiFi technology. WiFi is normally deployed in a hierachical structure. WiFi also supports ad hoc communication, but its short-range transmission capabilities limit its use for longer range communication. Also, typical WiFi ad hoc networks are only capable of sending and receiving messages via other WiFi links.

An ideal scenario is that heterogeneous wireless links/networks co-exist, interconnected in a universal (non-hierarchical) ad hoc network, where information can flow from one node to another through multiple hops seamlessly with long-range capabilities. However, it is extremely difficult, if not impossible, to flexibly switch from one network link to another without overhauling the existing Transmission Control Protocol (TCP)/Internet Protocol (IP) protocol stack and adding the necessary routing logic to support it. This fundamental limitation derives from the rigid network layering design and the location-oriented information retrieval model through IP addresses, where forwarding decisions are strictly enforced by the routing protocols. To circumvent the constraints, we leverage Named Data Networking (NDN) architecture, to design and build a ubiquitous information-centric wireless ad hoc network with a variety of heterogeneous wireless links and long-range capabilities. In this paper, we focus on implementing an interface between NDN and LoRa, and interconnecting LoRa and WiFi networks through NDN Forwarding Daemon (NFD) as the networking layer, and the same idea can be applied to other wireless technologies as well.

This system, or a similar system, with the proper application, could serve applications such as a disaster relief operation very well. With the lower power usage, our system will provide increased battery life to deployed devices, and not require excessive amounts of power for routing computations. Furthermore, with LoRa specifically, the higher range would allow for further reach and more reliable distribution in sparse network environments. This deployment would also benefit from the capability of NDN to return data upon the shortest possible path, allowing for minimized response time, as a person in distress would need to be able to request help with the most efficiency, and receive a digital response quickly to minimize their worry and personal risk, and then await the arrival of the physical aid that they requested. Furthermore, due to the lack of need for addressing in NDN, a new node in the network can join the network with very little overhead. The device would simply connect to the network, relay packets as needed, and upon issuing its own request, that request would be handled by the network to provide a response to the Interest packet. Any discovery and routing mechanisms would not necessarily be required, due to the capability of NDN to forward packets in a wireless environment with no knowledge of nearby nodes; the packets are simply rebroadcast, and other nodes deal with them as needed. Many applications, such as smart cities and sensor networks, will benefit from this new technology as well. In these IoT device filled environments, the density of devices creates a need for effective and resilient communication systems. Our system could provide a way to disseminate requests with minimal routing and processing overhead, and still provide the long-range, low-power communication needed.

Since IoT is growing rapidly, IoT and ad hoc network are popular research topics. The usability of LoRa for communication in a smart city is examined by Magrin et al.[3] Centenaro et al. put forth a survey examining the array of technologies that are available and are currently being investigated for potential use in IoT networks and smart city deployments[2]. The usability of LoRa within an IoT deployment in cities was examined by Augustin et al. using an assortment of configurations within a simulator[4]. Kokkonis et al. implemented a smart IoT fuzzy irrigation system using LoRa and Zigbee transmission media[5]. Reina et al. conducted a survey of works on multi-hop ad hoc networks in disaster response settings, and various researches had been done in that field[6]. Fang et al. used IoT to present an early warning system for flooding caused by snowmelt[7]. Narayanan and Ibe put forward Disaster Recovery Networks (DRN) and Search and Rescue Network (SRN) as two types of networks that are useful for disaster relief operations[8]. Ochoa and Santos also used a Delay Toerant Network (DTN) network for urban search and rescue operations[9]. Conti and Giordano showed how classic Mobile Ad hoc Networks (MANET) have evolved, and turned into DTN[10, 11]. Another work[12] investigated the performance of various opportunistic forwarding schemes in DTNs. Raffelsberger and Hellwagner combined the Better Approach to Mobile Ad-hoc Networking (BATMAN) and Optimized Link State Routing (OLSR) routing protocols to enable a network to switch between MANET and DTN modes[13]. Different from these address-based solutions, we designed an information-centric network that can be effectively used in IoT, smart city, and disaster relief scenarios. Yuan et al. described how forwarding can be handled within large scale NDN networks, and evaluated difficulties that larger networks introduced[14]. Mick et al. utilized both NDN and conventional wireless communication with the goal of enabling autonomous vehicles to securely share information[15]. A lightweight authentication system for NDN-based IoT deployments was developed in Ref. [15]. A prioritized forwarding scheme was implemented by Amadeo et al. to control traffic flows in vehicular networks, while maintaining the efficiency of the network[16]. Differing from the discussed studies, our work makes use of both LoRa and WiFi to create a heterogeneous NDN-based wireless network that has capabilities to communicate at high ranges.

To enable the use of LoRa's long-range transmission capabilities, there are three primary technical challenges that we must overcome: (1) Interfacing NDN with LoRa. At present, there is no direct method for NFD to communicate via LoRa. (2) Synchronization of sending and receiving operations. LoRa uses a half-duplex transmission mechanism, and thus is unable to perform true full-duplex communication. Therefore, we must design a mechanism that is able to synchronize the sending and receiving of messages. (3) Enabling LoRa to conduct ad hoc communication. Currently, the LoRa transmission protocol does not explicitly allow for ad hoc device-to-device transmission, so we need to implement this feature. Addressing these challenges, the following contributions are made in this paper.

(1) Framework. We design and implement a framework to enable information centric ad hoc networking and multi-hop long-range communication through heterogeneous wireless links. This newly developed framework can provide benefits to an assortment of high-impact deployments including smart cities, disaster relief, and more.

(2) Interface. Specifically, the implementation focuses on creating a bi-directional interface that enables NDN communication via LoRa so that applications gain the capability to transport data from one LoRa device to many others in an ad hoc, multi-hop manner. This interface technique is not only suitable for LoRa, but is able to be applied to other similar wireless transmission technologies.

(3) Emulators. Throughout implementation, the necessity of a generic testing mechanism became apparent. This is to enable the debugging of different wireless protocols with NDN. Thus, we design and develop some universal emulators that enable testing below the network layer of NDN. These emulators are capable of imitating a variety of wireless traffic types both from, and to, NDN applications.

(4) Interoperability. NDN provides the capability to design dynamic forwarding strategies within its network layer. Thus, the design of several forwarding strategies to interconnect heterogeneous wireless links is enabled. These strategies help to establish a universal ad hoc wireless network that connects a variety of links, regardless of their protocol types.

(5) Experiments and results. Using Raspberry Pi and LoRa devices, real experiments are performed in the field with the goal of testing the effectiveness of our system. The results of these experiments show the capabilities of the system to perform long-range communication. The system is also capable of interconnecting LoRa and WiFi frictionlessly without any additional overhead for multi-hop communication. The advantages of using an information-centric architecture for information dissemination without support from infrastructure are also demonstrated through the results.

The paper is organized as follows: Section 2 describes the NDN architecture along with the advantages over the TCP/IP architecture that it provides. A bird's eye view of our system is provded in Section 3. Details of the implementation are given in Section 4. The evaluation of the system in a real environment is presented in Section 5. Our future plans are discussed in Section 6, which concludes this paper.

2 NDN architecture

NDN is a networking architecture that was implemented based on the information-centric networking paradigm[17, 18]. In the NDN networking paradigm, two types of packets are used: the Interest packet and the data packet. These packets are handled by three types of network nodes: the producer, router, and consumer. The producer serves the same role as a server in the TCP/IP model. Upon receiving a request for data, the producer produces data and returns it to the requester. The consumer similarly serves the role as a client: It requests data. The router serves to route requests toward an appropriate data producer[19]. The primary software system that handles NDN packets is named the NFD. NFD uses three main data structures while processing packets: the Pending Interest Table (PIT), Forwarding Information Base (FIB), and the Content Store (CS)[14]. Figure 1 shows how NFD processes packets. First, upon receiving an Interest packet, the CS is checked. This data structure stores previously received data packets, and they are used to satisfy Interests that are received later. If the desired data exist in the CS of a node along an Interest forwarding route, then the node having that

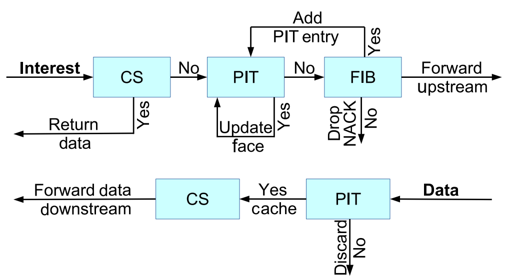


Fig. 1 NDN Interest and data flow.

data will not forward the Interest further. Instead, the node will reply to Interest with the data packet cached in its CS. If no matching data packets are found within the CS, the forwarder then checks for matching PIT entries for that Interest. The PIT keeps track of unsatisfied Interest packets that have already been forwarded and the Faces that they are received from. If the Interest matches an entry in the PIT, then the Face that Interest arrived on is recorded in the existing PIT entry, and that Interest is discarded. If no PIT entry exists, then the FIB is checked for a valid entry. The FIB serves as the routing table for NFD. It contains a list of Interest name prefixes and the Faces that they should be forwarded to. If a node finds a valid entry in the FIB for an Interest, a new PIT entry is created and the Interest is forwarded onward[20]. Figure 1 also illustrates the processing of data packets as they are returned from producer nodes. When a data producer receives an Interest packet, then the Interest will be replied to with the data requested, should the producer have it. Then the data are forwarded back to the consumer, following the PIT entries created by the Interest packet. These PIT entries are cleared, and each node stores the data in its CS for later use.

NDN retrieves data by its unique name rather than its location in the network. This new paradigm allows data to be cached in the network layer for a time by any router rather than in an application. This caching capability allows for the retrieval of data without a direct route to the data's producer[21]. The traditional concepts of source and destination are no longer supported well by this new paradigm. Obtaining data by its name instead of location eliminates the need for setting up data delivery pathways or session establishment that were essential to the operation of a TCP/IP network[22], where each node is assigned an IP address and applications communicate by sending data to each other's address. Moreover, NDN can take advantage of the broadcast nature of various wireless links to enable effective information dissemination as we did in this paper, whereas IP-based approaches attempt to use wireless routing protocols[6] to determine a single best path over an individual wireless interface as the next-hop physical medium and thus result in ineffective information propagation, particularly when the destination IP address is unknown, e.g., in disaster rescue scenarios.

3 Framework overview

An illustration of the overall architecture of the NDN-based ad hoc wireless network and Interest flow is shown in Fig. 2. The protocol stack consists of three general layers: NDN application layer, NFD (network) layer, and the wireless layer, including both LoRa and WiFi. Similar to the standard model, the application provides an interface for end users and allows them access to various services, such as map and chat. At this layer, conversation with other NDN applications may be initiated through the creation of NDN Interest packets. As the Interests flow out of the application layer, routing and forwarding services are provided by the network layer. In NDN, this layer includes a forwarding strategy module. This module allows the network to make intelligent, informed decisions regarding the forwarding of Interest packets. These decisions may include whether or not to forward a packet to multiple interfaces simultaneously, delivering the packet to the application layer, or potentially dropping the packet. Therefore, this strategy layer enables packets to be transmitted without friction, and with minimal overhead via heterogeneous network paths, by enabling the dynamic forwarding of packets. This layer is used by our work to demonstrate how a forwarding strategy can aid in enabling communication through a network path that includes both WiFi and LoRa links. Within the wireless layer, point-to-point data delivery between nodes is provided. This can include different types of wireless links, such as LoRa, WiFi, and others. The service connecting the network and wireless link layers is refered to as the transport service. NFD currently supports WiFi and other tunnel-based multi-access transports, such as User Datagram Protocol

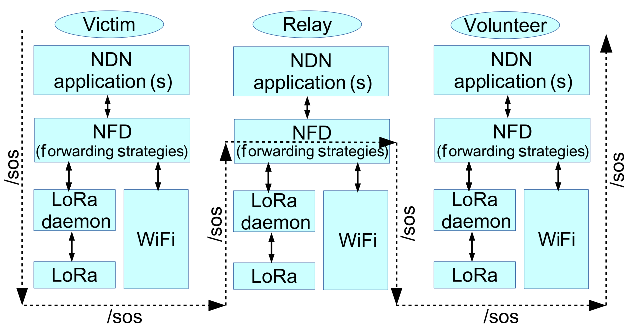


Fig. 2 NDN-based ad hoc network overview.

(UDP) and TCP socket tunnels, but there is no transport service provided to directly communcate via a LoRa link. Therefore, one of the challenging and essential parts of this work is providing an out-of-band solution that allows NFD to communicate via LoRa (or other currently unsupported wireless links) without modifying the existing complex transport services of NFD. For this purpose, a LoRa daemon application has been designed and subsequently developed.

While developing our interface system, we found that testing it without deploying it was difficult, and deploying a potentially flawed system to the devices for testing was inefficient. Thus, we developed emulators allowing for the bottom-most layer of the system to be replaced with a UDP socket that can communicate with a local application. The local application would then act either as an NDN producer application, an NDN consumer application, or a relay node.

To show an example of the network, as shown in Fig. 2, a victim needs assistance during an emergency but she does not know who can volunteer to help her. She issues an Interest messages */sos* to the NFD, where the forwarding strategy is configured to use a LoRa link. Thus, the message is delivered via the LoRa link to a relay node, who is not able to help. This node's forwarding strategy, however, is configured to relay the message through a WiFi link to the network. Finally, a volunteer receives the message via her WiFi channel and decides to provide assistance to the victim and sends a data packet back using the reverse path. As we can see from the forwarding process, the forwarding strategies can be flexibly configured to enable adaptive communication over multiple heterogeneous wireless links. With the power of LoRa's long-range communication, the victim is able to achieve long-range adaptive communication in this NDN-based ad hoc network.

4 Implementation

In this section, first, we designed and implemented an interface between NFD and LoRa through two internal UDP sockets, and developed four universal emulators to test and debug wireless links without using wireless devices. Additionally, two forwarding strategies were designed and implemented to assist in the interoperability of LoRa and WiFi via NDN's network layer and to facilitate testing.

4.1 LoRa details

While designing our system, we discovered that the addressing provided by LoRa was fairly restrictive compared to what was required by our system. Due to the "address-less" nature of NDN, any addressing scheme will add unnecessary overhead to our system. Thus, we discovered that if all node addresses are set to "0" and all packets are sent with a destination address of "0", then this address works as a broadcast address, so all devices in range of the sender will receive the sent packets. This enables NDN's capabilities to function completely with no overhead from address-based routing protocols.

Thus, our system avoids using the LoRaWAN protocol, and focuses on using NDN's features for communication. Our implementation does, however, operate in a manner that is analogous to a class C LoRaWAN device. Thus, our system listens for incoming packets all of the time unless it is sending a packet.

LoRaWAN packets carry their own header, and NDN packets contain the header information for NDN packets. However, while testing in small areas, we found that we needed to create our own packet header for effective testing. Our packet header contains one byte, which stores the size of the NDN packet, and another byte storing an identifier for the node that sends the packet. This header is illustrated in Fig. 3. This header is added to the packets and interpreted by the encoder and decoder modules of our program. These addresses must be configured before starting the system on a given node, and there is no protocol for automatically configuring them while the network is running. These two fields are used for a simple addressing system to aid in the effective delivery of packets, and to allow for a true multi-hop communication, even in a small area. This mechanism is necessary for testing the system's multi-hop capabilities

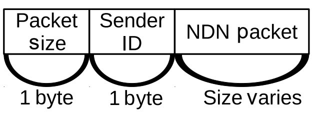


Fig. 3 Packet header added to NDN packets.

in a small area, as it can be used to limit which nodes may receive packets from a given node. In a 3-node network, this allows for the producer and consumer to be configured with the same addresses, and they listen for packets from the relay node. Then the relay node listens for packets from the producer and consumer. Thus, in a short-range network, multi-hop communication may still be tested. This mechanism is necessary because of a key feature of NDN communication: A packet may take the shortest route possible to a consumer, whether or not that is the exact reverse path of the Interest. While this feature is very useful and reduces latency in real deployments, it can interfere when testing multi-hop wireless communication in a small area.

The size field in our packet header notifies the LoRa daemon of the exact size of the NDN packet contained within the LoRa packet. During development, there were instances where a number of extra bytes of data would be read from the LoRa device into the packet buffer and would cause the NDN packet to be marked as corrupted. Thus, this size field became necessary.

4.2 Interfacing LoRa with NFD

Throughout the course of developing the system, several challenges arose that needed to be overcome before the system could be made properly functional. These challenges affected the system's design at a few points. To start, NFD does not provide a mechanism for communicating via LoRa. Therefore, our system was designed to utilize NFD's already provided transport services and mechanisms to provide connectivity. NFD supports UDP sockets and our LoRa devices come with an application that can support UDP sockets as well. Therefore, we use UDP sockets to link both ends to a common Translator module. Second, due to the capability of NFD's Face system to avoid packet looping, an Interest packet that cannot be satisfied by a node is unable to be forwarded to the Face that it was received from by NFD. This feature made it so that in a multi-hop environment, Interest packets would not be further relayed once they were received to NFD since the Interests were from this same UDP Face. The design of a Relay Listener was required to overcome this issue. This module maintains a second UDP socket for communication with NFD. Through this mechanism, Interest packets that require relay may be sent back to the LoRa for further transmission. Thirdly, we had to modify the ArduPiLoRa library to have it work properly for the implementation of our system since we discovered a bug within the library that negatively impacted our system, and led to non-deterministic repeated packets that were very difficult to track down and fix.

Figure 4 illustrates the four main parts of the LoRa daemon that process packets as they flow through the system. These parts include: the LoRa Operator, the Translator, the Transfer Worker, and the Relay Listener. While all of these parts contribute to the flow of packets within the system, they each do different portions of the work and are ran in seperate program threads. The modules are introduced in details as follows.

4.2.1 LoRa operator

The half-duplex sending and receiving operations via the physical LoRa device are handled by the LoRa Operator. This module listens for packets from LoRa for a pre-configured time. Upon receiving a packet from the LoRa device, the packet is added to a queue that the Translator will process when it is available. If the time runs out or a packet is received, then the LoRa Operator performs a check for new packets from the Translator. If it finds any packet in the queue, it will send the first packet and send it onward through LoRa. By implementing the communication with LoRa within its own module, the interface to LoRa is abstracted to the higher layers of the system, and those other modules of the program do not need to directly interface with LoRa. Also, due to the receiving time required by the ArduPiLoRa library when listening for packets, this module has to run in its own thread, allowing for minimal need for synchronization with the other modules. This design allows for this module to have the most possible time available to interface with the physical layer, rather than having to do other operations. While we do not use the LoRaWAN protocol directly, our system operates analogously to a class C LoRaWAN device, as it is almost always receiving unless it is sending a packet.

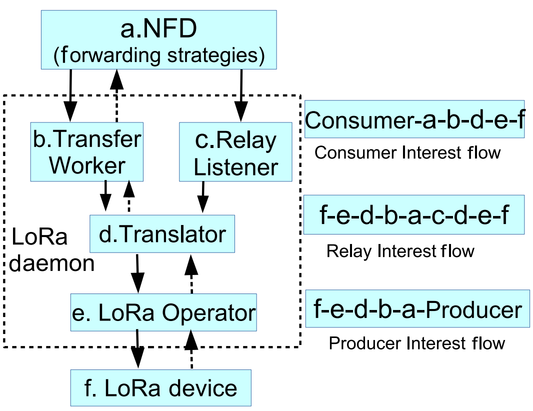


Fig. 4 Internal transport mechanisms of LoRa daemon.

4.2.2 Translator

The encoding and decoding of LoRa packets are handled by the Translator. This module serves as a “core” for the system’s internal transport, where packets are moved between the LoRa layer and NFD. A total of three queues are maintained to allow the other transport objects to place their packets in. After a configured tick time passes, the Translator will process all available packets from the other transport modules. Packets from NFD are encoded and sent to the LoRa Operator, and packets from the LoRa Operator are decoded and sent onward to NFD. Because the LoRa Operator can only conduct unidirectional sending or receiving messages, this module is not required to be constantly checking for packets, or moving packets; it merely needs to run with a frequency that allows it to move packets up to NFD and back to the lower LoRa Operator within the receiving window defined in LoRa Operator module. Thus, the tick time of the Translator is typically set to a lower value, e.g., one third of LoRa’s receiving window.

4.2.3 Transfer worker

This module handles communication via the primary socket that the system uses to communicate with NFD. Upon receiving a packet from NFD, it is added to the appropriate queue for the Translator to process when it is available. The Translator also utilizes the socket to directly send packets to NFD. By allowing the Translator to access the socket, the management within the system is simplified, as it removes some of the needs for synchronization between those parts of the program.

4.2.4 Relay listener

Due to the fact that NFD prevents Interest packets from being forwarded back to the interface that they are received from when the Face is a UDP tunnel, the Relay

Listener module was designed. It managed a UDP socket similarly to the Transfer Worker, but packets were never sent upwards to NFD via the Relay Listener. When a packet is received by the Relay Listener, it is added to a queue for the Translator to process. The Interest flows on a Producer, Relay, and Consumer are illustrated in Fig. 4.

4.3 Emulator

We designed two types of emulators: local emulators and remote emulators. Locally, we designed three emulators to imitate behaviors of a consumer, a producer, and a relay node. These emulators allow us to test various components of our main system without the need for a physical device, and to isolate the higher layers of the network stack to find where issues are within the system. Regarding remote emulators, we first designed a remote producer that serves a producer application to reply with Data packets remotely without LoRa daemon and upper layers. Second, we designed a second producer that sits upon the Translator module. These emulators can be used for troubleshooting remote communication of the system. Both types of emulators can be used not only in our system, but also in other NDN-based applications without wireless devices for quick debugging and troubleshooting.

4.3.1 Consumer emulator

The Consumer emulator creates an NDN Interest packet with a sequence number, then encodes it to the packet format required by upper layers, and then sends the packet into the LoRa daemon through a UDP socket. The packet is then handled by the LoRa daemon and goes to NFD, and further arrives at the NDN application layer. The returning data packet from the application layer is returned to LoRa daemon, and further returns it to the emulator. Figure 5a illustrates the process. The emulator can log the Round Trip Times (RTTs) of packets through the system, allowing for benchmarking of the system's higher layers to fine tune the timers that exist in the system for synchronization. Also, it allows us to see any issue with the higher layers much more quickly than by running the entire system.

4.3.2 Producer emulator

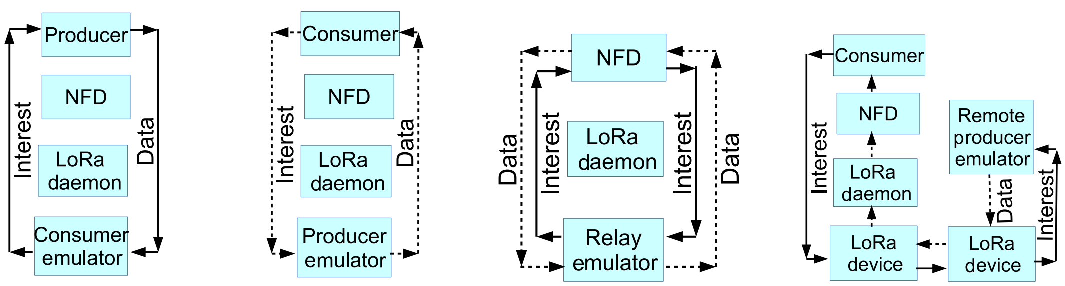
The Producer emulator listens on a UDP socket for Interest packets from the LoRa daemon. Upon receiving an Interest packet, the emulator decodes it and uses NDN's Type Length Value (TLV) encoding to create an Interest object. It then uses that Interest to create a matching data object and returns that data packet to the LoRa daemon. Figure 5b illustrates the process.

4.3.3 Relay emulator

The Relay emulator handles both pass-through Interest and data packets. First, it generates an Interest packet and pushes it to LoRa daemon and further to NFD. Then NFD returns the same Interest. Second, the emulator generates a data packet and pushes it to LoRa daemon and NFD. NFD returns the data packet. We created this emulator to test an issue that we encountered with multi-hop communication, and it showed that without proper configuration of our Interest and data packets, the PIT entries on the intermediary nodes would likely expire before the data packet reached the Relay node. Note that both Interest and data packets are not going to the application in this emulation. Figure 5c illustrates the relaying process of both Interest and data packets.

4.3.4 Remote emulator

The LoRa Remote Producer emulator runs on a different machine that runs LoRa daemon. This Producer acts like a node running the LoRa daemon with NFD and NDN



(a) Consumer emulator (b) Producer emulator (c) Relay emulator (d) Remote producer emulator

Fig. 5 Different types of LoRa daemon emulators.

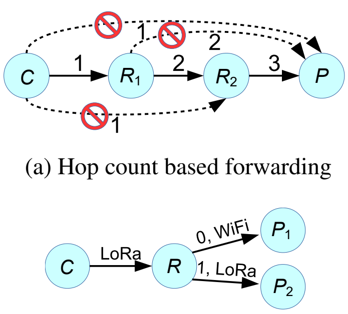
Producer. However, this emulator does not use NFD and the application layers. To do this, it receives packets from LoRa, returns a data packet, and sends it through the LoRa device back to the Consumer node. This allows us to isolate one node to test for issues with the physical transport. Figure 5d illustrates the process. In addition, we also designed a few other remote emulators that can help debug LoRa daemon and benchmark the performance of isolated modules. We skip the details due to space constraints.

4.4 Forwarding strategy

Within the forwarding pipeline of NFD, a flexible mechanism is provided that allows users and developers to define and execute customized forwarding strategies. Forwarding strategies may consider an assortment of information, including application needs, system requirements and capabilities, or current network conditions. Strategies are also able to utilize any available heterogeneous interfaces to provide quick information dissemination via multi-path routing. In our work, we have defined two useful, but simple forwarding strategies: hop count based forwarding and heterogeneous wireless links forwarding.

4.4.1 Hop count based forwarding

Testing multi-hop forwarding over the newly created networking system is required. However, LoRa is capable of covering a very long distance with only a single hop. Due to the fact that wireless signals are broadcast, finding good locations for testing multi-hop communication is not practical or convenient. As shown in Fig. 6a, multiple receivers may receive a one-hop wireless signal simultaneously. With the goal of overcoming this issue, we define a new attribute for point-to-point links using a feature in the Link Protocol (LP) layer of NDN. Using this feature, we defined a HopCount field within the LP layer. This HopCount value is incremented by one as the Interest packet travels from one link to another. Based on this value, we can define the multi-hop forwarding strategy in this way: (1) Line up one consumer (C), two relay nodes (R_1 and R_2), and one producer (P); (2) let relay nodes R_1, R_2, and producer P only accept Interests with HopCount values 1, 2, and 3, respectively. As a result, all other broadcasting messages not satisfying this condition will be filtered out as illustrated in Fig. 6a. Utilization of this forwarding strategy is useful in limiting the message forwarding pathways. Thus, it is key in debugging and evaluating multi-hop behaviors in a complicated wireless environment. Hop count forwarding can be used for other scenarios. For instance, users can define a forwarding strategy which only accepts Interests within 10 hops and consequently the Interests can only "float" in that constrained area. The similar idea can be applied to physical distance, and users can define a distance based forwarding strategy to limit Interests within a certain radius from the information requesting source.



(a) Hop count based forwarding

(b) Interoperability

Fig. 6 Forwarding strategies.

4.4.2 Interoperability over heterogeneous links

NDN is able to utilize all available physical interfaces on a node. They are made visible to the forwarding strategies in the form of NFD Faces. Forwarding strategies are able to select any number of Faces to send an interest through. In our network, two wireless interfaces are available to each device: LoRa and WiFi. To demonstrate the capabilities of NDN to select which interface to send packets to dynamically, we created a forwarding strategy that is able to alternate forwarded Interests between being forwarded via LoRa and WiFi links. This mechanism utilizes a boolean flag to track the next interface to send a packet to. Once a Face is used to forward an Interest, the flag is toggled, which signals that the other interface should be used to send the next packet. Figure 6b shows the flag value and how the packets are forwarded on each link. While this implementation is simple, it illustrates the capabilities for intelligent forwarding strategies to be implemented. A more powerful forwarding strategy would utilize network information and other metrics to make forwarding decisions. Another forwarding strategy we have implemented is to send an Interest to both WiFi and LoRa links simultaneously for maximizing the possibility of information retrieval.

5 Evaluation

5.1 Experimental environment

To evaluate our system, Raspberry Pis (RPis) equipped with LoRa devices were used. RPis include WiFi onboard, but a LoRa module is not included on the device. Thus, separate LoRa devices are connected to the RPis to perform experimentation. The Operating System (OS) installed on the RPis was Rasbian, and NFD version 0.6.0 was compiled for the software system. Anker PowerCore 20100 portable chargers were used to power the hardware system. Several experiments were performed in the field to evaluate the system performance. Figure 7 shows a map of the testing area and the hardware device. The distance between the connected locations on the map is approximately 1 km. These experiments were typically used to gather data on the RTTs of packets, and to see what the relationship is between the size of the packet and the RTT. These experiments should show how well the system will work in a real deployment. For simple understanding of the results, the average RTT of 10 packets is used at each packet size in the following graphs.

5.2 One-hop experiment

In Fig. 8a, the line using circles as points shows the RTTs of packets versus their size over a raw LoRa link. This experiment evaluates the system over a range of 1 km, and the devices are within line of sight of each other. These results demonstrate that the RTTs of packets are fairly regular around 2 s. Also, it is noted that as the packet size increases, so does the RTT of the packet increase slightly. This is due to an increase in the time to transmit the packet. Throughout this experiment, there is a timeout rate of 3.79% of packets. This experiment gathers a baseline of LoRa's transmission speed without any time overhead from our full LoRa daemon system.

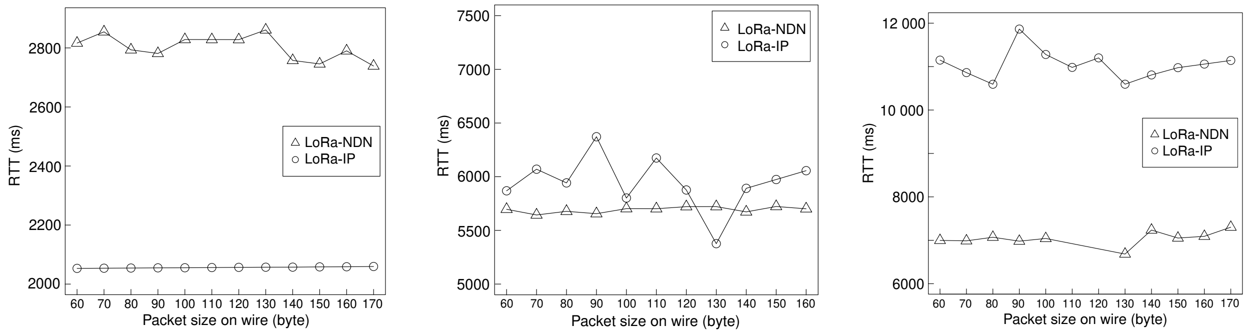
Fig. 7 Experimental map and device.

In Fig. 8a, the line with triangles as points shows the RTTs of NDN packets that are being transmitted via the LoRa daemon over the same 1 km distance, maintaining line of sight. The default NDN headers were used for the packets, and the size of the packets was increased by adding a number of arbitrary characters to the end of the name of the packet. An NDN packet header and data signature is about 60 bytes. Thus, no Data packet is able to pass through the network with a size smaller than 60 bytes. This is why the LoRa-NDN line starts around packet size 60 bytes. The average RTT for the single-hop LoRa-NDN experiment was around 2.8 s. The 0.8 s increase in delay was caused by the timed wait mechanism length in the translator (see Section 4.2.2). We also discovered that LoRa is unable to carry a packet that is more than 170 bytes in size over a 1 km distance.

Throughout this experiment, the time that the LoRa Operator listens for packets was set to 1.5 s. This experiment demonstrates the overhead that NFD and our system add to LoRa when the LoRa Operator's receiving time is set relatively high. This experiment showed a timeout rate of 6.77%. While this is worse than raw LoRa, it is still well within the boundaries of a usable system. The timers that are used to synchronize the internal transport mechanisms are likely causing this increased timeout rate. This is due to the fact that the timers can cause instances where a node might not be receiving for a split second. If that time overlaps with the time when a preamble of an incoming packet should be received, that packet will be lost.

5.3 Multi-hop experiment

Two hops: Because LoRa is capable of broadcasting, we also evaluated it in a two-hop environment. Figure 8b illustrates the change in RTT as the size of packets changes for NDN and an IP-like protocol that we implemented for testing. The results of this experiment show that the RTT of packets through two LoRa hops is approximately equal to two times the RTT of LoRa



(a) One hop (b) Two hops (c) Three hops

Fig. 8 RTT for one and multi-hop experiments.

over one hop as expected. These results seem to indicate that our LoRa system can perform as well as the IP-like solution, however, this contradicts the observation of the single hop experiment. LoRa-NDN exhibits a timeout rate of 12.72%, while the IP-like protocol shows a timeout rate of 40.18%. The RTTs of the IP-like protocol also show higher fluctuation than the NDN-based solution. Further investigation into the exact cause is required.

Three hops: To investigate the system further, we performed experiments via three LoRa hops using NDN and comparing it to our IP-like protocol. In the test of NDN, we utilized our hop-based forwarding mechanism to control the path of Interests until they reached the producer. For the IP-like experiment, a symmetric routing pathway was set on the relay node between the producer and consumer nodes. In Fig. 8c, the line with triangles shows the RTTs for LoRa-NDN. They average about 7 s, much lower than the RTTs of LoRa-IP (shown by the line with circles) which exhibit an average RTT of 11 s. These results are surprising, but are explainable with the features of NDN. IP-type packets are required to travel a pre-configured path back to the requester. However, when deployed in a broadcast-based environment, NDN exhibits unique benefits. NDN allows packets to return directly to the consumer while following the most direct physical path to the consumer among the routers that have forwarded the respective Interest packets. This allows for optimal performance of the network. The experiments in a multi-hop environment show that this new information centric wireless network is readily capable of covering a large area of several kilometers with just a few hops through device-to-device communication. When compared to a more conventional IP-based design, this new architecture exhibits great capabilities to enable fast and efficient information dissemination and collection.

5.4 Forwarding strategy for interoperability

To show NDN's capability to choose which interface to forward an Interest to, we implemented a forwarding strategy that sends every other Interest packet via a different interface. These experiments were performed in a short-range environment. The strategy was configured to alternate sending Interests between WiFi and LoRa. Figure 9 shows the RTTs of packets when sent within a Y topology (see Fig. 6b) with this strategy installed and running on the middle node. Figure 6b shows that the system is capable of switching between both transmission mediums reliably. The two media are easily differentiable, as the delay via WiFi is around 3 s, while is much lower than that of the LoRa link. This demonstrates that even with LoRa's capabilities to communicate at high-range, its latency is large. This experiment shows a timeout rate of about 3%, showing that under the right conditions, this system is capable

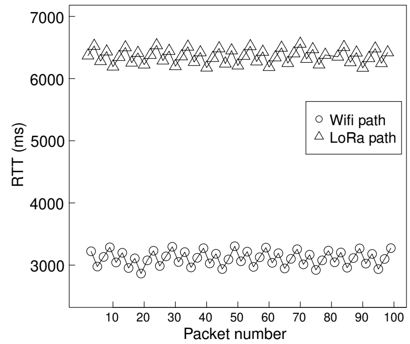


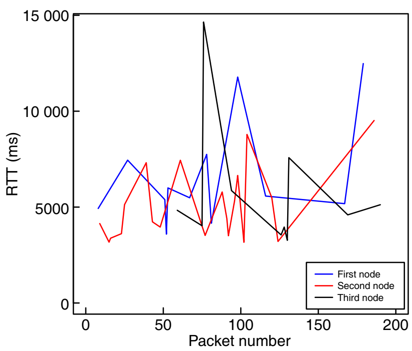
Fig. 9 RTT in a Y topology.

of reliable performance. While this strategy is not an intelligent implementation, it does show NDN's ability to frictionlessly bridge heterogeneous wireless links and maximize the efficiency of information retrieval. Also notice that when retrieving data, an end user does not need to specify a potential location where the data may be stored. The network can interpret the semantics of Interest names and follow a pre-defined forwarding strategy to facilitate information dissemination.

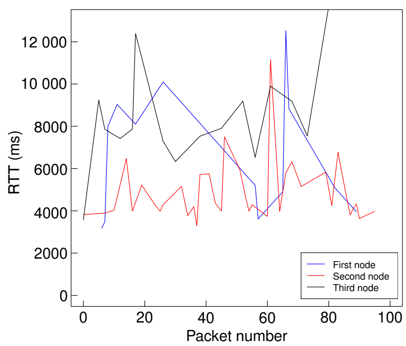
5.5 In-network caching

One hop caching: With the goal of investigating the functionality of NDN's PIT and CS mechanisms, and their capabilities to minimize network congestion and improve or at least maintain performance within a wireless environment, we performed several experiments that utilized four nodes: three of which were consumers, one of which was a producer. A set of four experiments were performed, each with a different set of conditions. To begin, each consumer sent Interest packets that carried different names roughly at the same time. These conditions resulted in a high number of packet collisions, causing many packets to be lost. The RTTs of these packets are shown in Fig. 10a. During this test, the nodes exhibit timeout rates of 94%, 89.5%, and 94.5%. These extremely high timeout rates are a result of the high number of packet collisions, and the fact that NDN's capabilities are unused because of the differing packet names. In an attempt to allow successful content discovery, we ran a similar experiment with the transmission of Interest packets staggered throughout three seconds. These results are shown in Fig. 10b. The timeout rate of each node is 87%, 61%, and 84%, respectively, which shows an improvement over the previous experiment.

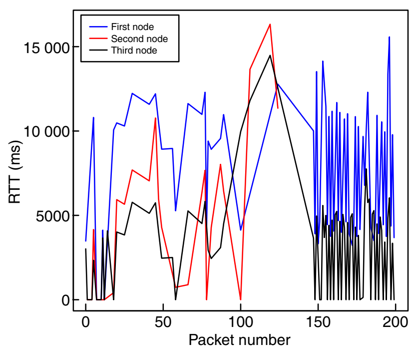
To examine the usefulness of the CS and PIT, and leverage their capabilites, we did a similar set of experiments as above, but used Interest packets that carried the same names. Figure 10c illustrates the results of that experiment when the Interests are sent simultaneously, whereas Fig. 10d shows the results when Interests are staggered. In the experiment when Interests are sent at the same time, the nodes have timeout rates of 64%, 84.5%, and 58%, respectively. This demonstrates that when the PIT and CS are leveraged in a congested environment, the network performance



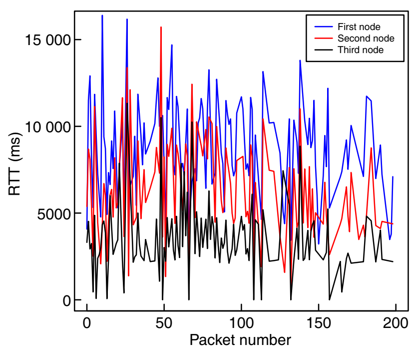
(a) Different names simultaneously



(b) Different names staggered



(c) Same names simultaneously



(d) Same names staggered

Fig. 10 RTT for single hop caching.

and reliability can be greatly improved. When the Interests are staggered, the timeout rates are 34%, 40.5%, and 33.5%, respectively. This illustrates that a more robust collision avoidance protocol could greatly benefit our system. For staggered experiments, the blue line represents the RTTs of packets sent by the node that first sends an Interest of a given name, red the second, and black the third. It is also seen that nodes that send their packets later almost always see lower RTTs. In addition, in Fig. 10c, at packet number 150, the second consumer suddenly drops and we notice a sharp increase of received packets. This indicates that a small number of consumers requesting data will enhance the chance of obtaining data. Also in Fig. 10d, we find that many Interests have almost zero RTTs and this can be explained by the fact that the previously requested data have been received by the LoRa layer, and are being processed by the LoRa daemon while the Interest packet is being processed, causing it to appear that the network is working extremely quickly.

These experiments display the capabilities of NDN's CS and PIT mechanisms to greatly increase network performance in a relatively congested environment when nodes are requesting the same data. Also, these experiments show that when nodes are requesting the same data, nodes that request the data after other nodes have previously requested it will more often than not yield a much lower RTT.

Two hop caching: With the goal of isolating NDN's caching mechanism for evaluation, we configured an upside-down Y topology, where only the middle relay node has caching enabled. One consumer sent Interest packets to retrieve data from the producer. Then the second consumer retrieved the data by sending Interest carrying the same names as the first set of Interests. Figure 11 shows the results gathered by this experiment. The line with triangles in Fig. 11 represents the RTTs of the second consumer's packets, and the line with circles represents the RTTs of the packets that the first consumer sent. While this experiment was performed in a simple network topology, it illustrates the capability of NDN's caching mechanism to provide massive benefits to the network's performance. Furthermore, after the first set of interests was sent, the producer node was powered

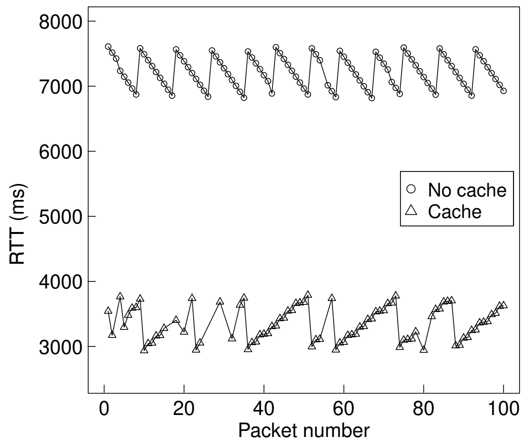


Fig. 11 RTTs of packets in Y topology with/without caching.

off. This demonstrates NDN's capability to retrieve data even through a producer for that data may not be directly reachable, provided that the data have passed through the network at some point in time previously.

6 Conclusion and future work

Our work shows that while the transmission speed of LoRa is slower than that of conventional WiFi, it has a massive increase in range with a minimal loss to reliability. Thus, LoRa has shown itself to be viable as a Device-to-Device (D2D) communication medium in a number of scenarios. NDN's capability to use dynamic forwarding strategies within mobile enironments paired with the ability to use NDN over LoRa, WiFi, and other wireless links enables the creation of an information-centric wireless networking system that is readily deployable and capable of the efficient collection and dissemination of information. Furthermore, because of LoRa's low power usage, the system will be able to be used without a large amount of power. Our future work will involve developing applications for use over this system, such as a sensor network system or an emergency communication application. Also, we plan on further improving our existing LoRa daemon application to make it more efficient, and integrating it directly into NFD to simplify the interface between NFD and our system. Our future plans include further investigation of a more diverse set of wireless transmission technologies in order to make a more diversified, resilitent network that takes advantage of an intelligent forwarding strategy, enabling nodes to dynamically choose the most optimal interface or interfaces that an Interest should be forwarded to.

References

[1] J. Lin, W. Yu, N. Zhang, X. Y. Yang, H. L. Zhang, and W. Zhao, A survey on Internet of things: Architecture, enabling technologies, security and privacy, and applications, *IEEE Int. Things J.*, vol. 4, no. 5, pp. 1125–1142, 2017.

[2] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios, *IEEE Wirel. Commun.*, vol. 23, no. 5, pp. 60–67, 2016.

[3] D. Magrin, M. Centenaro, and L. Vangelista, Performance evaluation of loRa networks in a smart city scenario, in *Proc. 2017 IEEE Int. Conf. on Communications*, Paris, France, 2017.

[4] A. Augustin, J. Z. Yi, T. Clausen, and W. M. Townsley, A study of loRa: Long range & low power networks for the internet of things, *Sensors*, vol. 16, no. 9, p. 1466, 2016.

[5] G. Kokkonis, S. Kontogiannis, and D. Tomtsis, A smart IoT fuzzy irrigation system, *IOSR J. Eng.*, vol. 7, no. 6, pp. 15–21, 2017.

[6] D. G. Reina, M. Askalani, S. L. Toral, F. Barrero, E. Asimakopoulou, and N. Bessis, A survey on multihop ad hoc networks for disaster response scenarios, *Int.J. Distrib. Sens. Netw.*, vol. 2015, p. 647037, 2015.

[7] S. F. Fang, L. D. Xu, Y. Q. Zhu, Y. Q. Liu, Z. H. Liu, H. Pei, J. W. Yan, and H. F. Zhang, An integrated information system for snowmelt flood early-warning based on internet of things, *Inf. Syst. Front.*, vol. 17, no. 2, pp. 321–335, 2015.

[8] R. G. L. Narayanan and O. C. Ibe, A joint network for disaster recovery and search and rescue operations, *Comput. Netw.*, vol. 56, no. 14, pp. 3347–3373, 2012.

[9] S. F. Ochoa and R. Santos, Human-centric wireless sensor networks to improve information availability during urban search and rescue activities, *Inf. Fusion*, vol. 22, pp. 71–84, 2015.

[10] M. Conti and S. Giordano, Mobile ad hoc networking: Milestones, challenges, and new research directions, *IEEE Commun. Mag.*, vol. 52, no. 1, pp. 85–96, 2014.

[11] M. Conti and S. Giordano, Multihop ad hoc networking: The reality, *IEEE Commun. Mag.*, vol. 45, no. 4, pp. 88–95, 2007.

[12] A. Martín-Campillo, J. Crowcroft, E. Yoneki, and R. Martí, Evaluating opportunistic networks in disaster scenarios, *J. Netw Comput. Appl.*, vol. 36, no. 2, pp. 870–880, 2013.

[13] C. Raffelsberger and H. Hellwagner, A hybrid MANET-DTN routing scheme for emergency response scenarios, in *Proc. 2013 IEEE Int. Conf. on Pervasive Computing and Communications Workshops*, San Diego, CA, USA, 2013, pp. 505–510.

[14] H. W. Yuan, T. Song, and P. Crowley, Scalable NDN forwarding: concepts, issues and principles, in *Proc. 2012 21st Int. Conf. on Computer Communications and Networks*, Munich, Germany, 2012, pp. 1–9.

[15] T. Mick, R. Tourani, and S. Misra, Laser: Lightweight authentication and secured routing for NDN IoT in smart cities, *IEEE Internet Things J.*, vol. 5, no. 2, pp. 755–764, 2018.

[16] M. Amadeo, C. Campolo, and A. Molinaro, Priority-based content delivery in the internet of vehicles through named data networking, *J. Sens. Actuator Netw.*, vol. 5, no. 4, p. 17, 2016.

[17] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, Networking named content, in *Proc. 5th Int. Conf. on Emerging Networking Experiments and Technologies*, Rome, Italy, 2009, pp. 1–12.

[18] L. X. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. C. Zhang, Named data networking, *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.

[19] M. Amadeo, C. Campolo, and A. Molinaro, Forwarding strategies in named data wireless ad hoc networks: Design and evaluation, *J. Netw. Comput. Appl.*, vol. 50, pp. 148–158, 2015.

[20] A. Afanasyev, J. X. Shi, B. C. Zhang, L. X. Zhang, I. Moiseenko, Y. D. Yu, W. T. Shang, Y. B. Li, S. Mastorakis, Y. Huang, et al., *NFD Developer's Guide*. NDN, 2016.

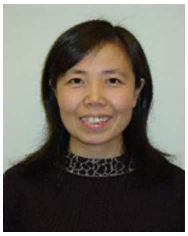
[21] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, and L. X. Zhang, VANET via named data networking, in *Proc. 2014 IEEE Conf. on Computer Communications Workshops*, Toronto, Canada, 2014, pp. 410–415.

[22] L. Wang, R. Wakikawa, R. Kuntz, R. Vuyyuru, and L. X. Zhang, Data naming in vehicle-to-vehicle communications, in *Proc. 2012 Proc. IEEE INFOCOM Workshops*, Orlando, FL, USA, 2012, pp. 328-333.

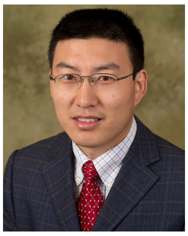
Anthony Dowling is pursuing the PhD degree at Clarkson University. He received the bachelor degree in computer science from Clarkson University in 2019, and the master degree in computer science from Clarkson University in 2020. He worked with the Air Force Research Laboratory Information Directorate in Rome, NY in the summers of three years: 2018–2020, via the Visiting Faculty Research Progrm. His paper "Empowering Named Data Networks for Ad-Hoc Long-Range Communication" won the Charles Kao Best Paper Award at WOCC 2020. His research focuses on Named-Data Networking (NDN) and its uses in various applications, along with the evaluation of NDN's performance in a variety of network environments.

Lauren Huie received the BS degree in electrical engineering from the State University of New York at Binghamton in 2005. In 2007, she received the MS degree in electrical engineering from the Pennsylvania State University. She received the PhD degree from the State University of New York at Binghamton in 2013. She is currently with the Air Force Research Laboratory Information Directorate in Rome, NY. Her research interests include sensor networks, communications, estimation and detection theory, and physical layer security. She is a member of the IEEE.

Laurent Njilla received the BSc degree in computer science from the University of Yaoundé-1 in Yaoundé, Cameroon, the MSc degree in computer engineering from the University of Central Florida in 2005, and the PhD degree in electrical engineering from Florida International University in 2015. He joined the Cyber Assurance Branch of the Air Force Research Laboratory Information Directorate in Rome, NY, as a research electronics engineer in 2015. As a basic researcher, he is responsible for conducting and directing basic research in the area of cyber defense, cyber physical system, cyber resiliency, hardware security, and the application of game theory, category theory, and Blockchain technology. He is the program manager of the Center of Excellence in Cyber Security for the Historically Black Colleges and Universities & Minorities Institutions and the program manager of the Disruptive Information Technology Program at AFRL/RI. He has co-authored over 70 peer-reviewed journal and conference papers with best paper awards. He is a co-inventor of 3 patents and 3 patent applications. He is a co-editor of two books at Wiley-IEEE Press entitled *Blockchain for Distributed System Security* and *Modeling and Design of Secure Internet of Things*. He is a member of the National Society of Black Engineer.



Hong Zhao received the PhD degree in electrical and computer engineering from New Jersey Institute of Technology in 2004. She is a professor of electrical and computer engineering at Fairleigh Dickinson University, NJ, USA. Her research focuses on various aspects of broadband communications and computer security including network traffic/performance/security analysis and modeling, and hardware Trojan detection. She serves as an associate editor of *the Journal on Multidimensional Systems and Signal Processing*, and editor of *the Journal of Computing and Information Technology*. She also serves as vice chair of IEEE North Jersey Section. She has been a TPC member, symposium co-chair, technical paper reviewer, and book reviewer for IEEE conferences, journal magazines, and book publishers. She served as a board chair of the Wireless and Optical Communication Conference (WOCC) (2017–2018). She received AFRL VFRP award in 2014–2016, AFOSR SFFP award in 2017–2019, and 2015 IEEE Region 1 Award for Outstanding Support for the Mission of the IEEE, MGA, REGION 1, and Section.



Yaoqing Liu received the master and PhD degrees from the University of Memphis in computer science in 2011 and 2013, respectively. He is the program coordinator of M.S. in Cybersecurity and Information Assurance program (MSCSIA), and an assistant professor at Gildart Haase School of Computer Sciences and Engineering, Fairleigh Dickinson University (FDU). His research interests are networked systems (security, routing, algorithm, and measurement) and blockchain applications. His publications appear in highly reputed conference proceedings and journals, such as *IEEE INFOCOM*, *ACM SIGCOMM CCR*, and *ACM/IEEE ANCS*. He has been a TPC member and technical paper reviewer for IEEE conferences, journal magazines, and transactions. He is the inventor of four patents. His research work has been selected as one of the 2018 TechConnect Defense Innovation Awards recognizing the potential positive impact for the warfighter and national security.