## Title
Evolving intelligent devices for the future via named data networking



## Journal
XRDS Crossroads The ACM Magazine for Students, 26(1)

## ISSN
1528-4972

## Authors
Zhang, Zhiyi
Lu, Edward
Guan, Yu
et al.

## Publication Date
2019-09-17

## DOI
10.1145/3351482

Peer reviewed

# Evolving Intelligent Devices for the Future via Named Data Networking

**As the numbers and capabilities of networked devices continue to grow, they will play an increasingly important role in daily life. Ensuring security and usability will be the first and foremost challenge; Named Data Networking can help address this challenge through localized trust, usable security, and autoconfiguration.**

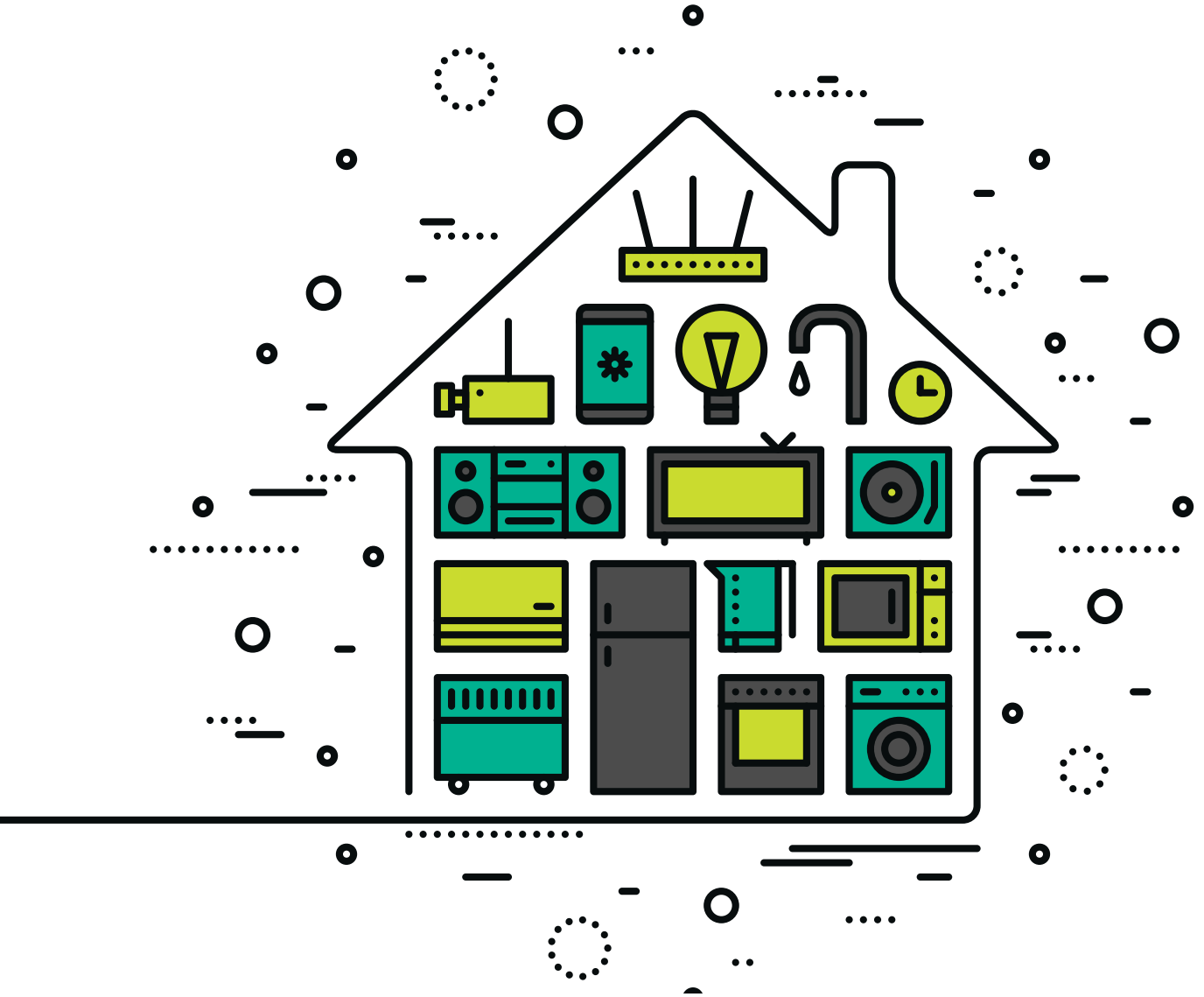
*By Zhiyi Zhang, Edward Lu, Yu Guan, Tianxiang Li, Xinyu Ma, Zhaoning Kong, and Lixia Zhang*

DOI: 10.1145/3351482

Smart devices, such as smartphones, smartwatches, smart refrigerators, and smart light bulbs are all around us today, and are expected to continue to grow in number and capability. The most important question to focus on now is not how to further improve these devices' according to quantitative performance measure, but what is the best way to make use of their ever-increasing capabilities to enrich the quality of our daily lives. We make our case by examining internet of things (IoT) devices in home environments.

Home IoT devices work for and are controlled by end users. Therefore, IoT devices in the future should work in a user-centric and user-friendly way. More specifically, they should have automatic configuration, strong user privacy and system security, an integrated framework that offers good application support, and provide all of the above within a home environment that is completely controlled by the home users.

Today's home IoT devices are far from meeting these requirements. Most (if not all) of them are controlled by cloud servers. Therefore, whenever the cloud servers or network connectivity is down, smart home devices' capabilities become limited. Furthermore, controlling home devices remotely poses potential user privacy risks, as user information from private homes can be exposed to other parties intentionally or otherwise, with or without the users' awareness. In addition, today's smart devices are interconnected by TCP/IP networking. The TCP/IP protocol stack was developed almost 40 years ago for networking mainframe computers and is not a good fit for networking today's IoT devices. For example, when a single device has multiple interfaces, as many devices do today (e.g., WIFI and Bluetooth low energy), TCP/IP has a difficult time utilizing all the interfaces, because each interface has a different IP address. As another example, applications use names to communicate but the IP sends packets to destination addresses, meaning some kind of mapping services must be provided to translate between them. Worse yet, since TCP/IP has no built-in security protection, home devices can be easily compromised and turned to zom-



bies in DDoS attacks, as we have seen in the Mirai attack [1].

Named Data Networking (NDN) [2] is a new Internet protocol currently in development that can support the evolution of IoT devices.

## DESIRED FEATURES OF FUTURE IOT DEVICES

There are several desirable features in IoT networks, which we believe are necessary to create high quality IoT services.

First, auto configuration is of vital importance to system usability and user experience. Auto configuration is a wide concept which overlaps with many features like networking and security. To be more specific, auto configuration at least contains networking configuration, security bootstrapping, and service discovery. One principle is configuration should require the least user involvement, both at device installation time and during the whole lifespan of the IoT device.

The second required property is security, which includes security bootstrapping, trust management, access control, user privacy, and DDoS defense. Such security support should be provided from hardware (e.g., hardware-based cryptographic support, such as random number generator) to software (e.g., crypto support APIs), and from the network layer (e.g., verifiable network packets) to the application layer (e.g., access control). Without security protection being designed-in, IoT devices are subject to unauthorized control, private data leakage (e.g., unauthorized access or eavesdropping on sensitive data), and many other threats.

Last but not least, a highly desirable property is an integrated IoT framework with open application layer support. Instead of today's stovepipe market, where different home IoT service providers sell their own proprietary solutions, an integrated IoT framework should be defined with open and unified application support. From an application developer's perspective, the support should include (1) APIs for usable resource discovery with unified resource identifier naming schema, (2) ease-of-use bootstrapping interfaces for device profiling and security bootstrapping APIs, (3) integrated trust management APIs, and (4) easy-to-use access control APIs.

## CHALLENGES OF IOT NETWORKING IN THE CURRENT TCP/IP ARCHITECTURE

Today's IoT networking are mainly based on the TCP/IP architecture, which was developed for the wired internet decades ago. The difference between wireless IoT networks and the traditional internet architecture poses a number of challenges with respect to the network

layer, transport layer, application layer, and lastly with security in general.

Looking at the network requirement from a high-level perspective, IP adopts a host-oriented communication model, which diverges from the data-centric nature of the applications. Applications focus on the retrieval of data using a unified resource identifier, while the IP network layer focuses on packet delivery to hosts identified by addresses. This mismatch of what applications want and what IP provides leads to additional complexity in the system (i.e. domain name system (DNS) to resolve application level names to IP addresses). From a low-level perspective, IoT networks with resource-constrained devices often rely on low-energy link layer technologies with a small maximum transmission unit (MTU); this conflicts with the IP protocol's relatively large header (IPv6, which is used for IoT environment, has a minimal header length of 40 bytes). Furthermore, mesh network routing under IP also causes challenges for maintaining routing information for each host in a multi-link environment.

Regarding the transport layer, TCP was designed for reliable byte-stream delivery over point-to-point connections between IP nodes that are always-on. However, IoT networking does not assume nodes are always-on in general; some may be sleeping to conserve energy. On top of this, a majority of IoT communication involves a small amount of data transfer, which renders TCP's handshake process for connection setup and teardown costly. Lastly, TCP treats the network as a virtual pipe and performs loss recovery end-to-end which causes additional delay and overhead in lossy wireless environments. It misses the opportunity of utilizing in-network storage to buffer packets at each hop so that a lost packet can be retrieved from the cache right before the loss point.

In terms of application layer requirements, IoT applications adopt the resource-oriented communication model. For example, smart-home monitors can request data generated by different sensors in a room. The first requirement is resource discovery. Resource discovery in IoT covers a broader scope; it can refer to services, devices, and data. This requires a generic approach for identifying heterogeneous IoT resources. For example, CoAP adopted a URI-based naming scheme for resource identification [3]. However, a fundamental limitation of these TCP/IP based solutions is that TCP/IP cannot use application-layer names directly for resource discovery; instead they have to deploy mapping services, such as DNS-based Service Discovery (DNS-SD) [4], increasing the system dependency and complexity.

**Future IoT devices will be more powerful, and their increasing capabilities can enable a new generation of applications.**

Lastly, there is the security question. TCP/IP was not designed with security and has thus been enhanced with a point-to-point channel-based protection model (e.g., TLS and DTLS), which secures the communication channel between two end-points. This model does not meet the security requirements of the IoT environment for several reasons. Establishing a secure channel may need multiple rounds of handshakes for channel authentication and security parameter negotiation—a task that can be time and energy consuming. Also, both ends of a communication channel need to maintain a certain state before the communication terminates, adding pressure on memory usage if a device communicates with many others, as well as energy consumption if the security associations require continued connectivity. It should be noted, IoT messages lose security protection once they are outside of the channel. For example, application data cached in middle boxes requires an additional trust relationship to be established between the resource owner and the middle boxes. These limitations drive the need for an object-based security model, which can secure the application data directly and allows each piece of data to be verified regardless of where it is stored or how it is retrieved.

### A BRIEF INTRODUCTION TO NDN

NDN is a proposed future internet architecture, which we believe provides an alternative to TCP/IP for networking future IoT devices. There are three key concepts related to NDN: naming, stateful forwarding, and data-centric security.

NDN is a data-centric network architecture, where applications "pull" data from the network by using the application-level names of the data. These names are defined by applications and are directly used by NDN for network layer packet delivery. NDN's networking model is in sharp contrast to the node-centric nature of TCP/IP, in which data is exchanged through point-to-point connections, and in which application layer identifiers are invisible to the network layer.

To pull data from the network, NDN needs two types of packets: interest packets and data packets. Data consumers send interests containing the name of the data they want to retrieve, and the network takes the responsibility to fetch the corresponding data packets from either the data's producer or any network device that happens to have a copy of the data (e.g., in cache). Within this process, no IP addresses are involved. Interests are forwarded in the network based on their name by the NDN forwarder at each node. Once an interest reaches a data packet with the matching name, the data packet will be sent back following the reverse path back to the consumer, in a hop-by-hop manner based on the traces left by the interest packet; the data packet can also be cached at the intermediate hops.

Each data packet is secured directly by the producing applications: The producer cryptographically signs the data packet, binding its name to the content. This enables consumers to fetch data packets from anywhere in the network, then use the producer's public key to verify the authenticity of each data packet. The producer may also encrypt the content of its data packets to achieve data confidentiality and make the decryption key available only to the

parties who are authorized with the access to the data.

## THE EVOLVING FUTURE OF IOT DEVICES WITH NDN

NDN can more naturally and robustly support auto-configuration, security, an integrated IoT framework, and heterogeneous network connectivity than ICP/TP.

With built-in security and naming schemes, autoconfiguration in NDN helps applications to bootstrap not only networking, but also security. Specifically, in an NDN-based home IoT network, as soon as a new device gets connected, it will obtain a unique and verifiable name under this IoT network, obtain a digital certificate for its own identity, and install the home's trust anchor certificate together with a set of security policies. The name and the certificate enables $D_{new}$ to be discovered and authenticated by other devices in the same home, and the trust anchor enables $D_{new}$ to verify packets received from others.

By communicating with named, secured packets, as opposed to channel-based security like TLS and IPSec, NDN secures the content being transferred regardless of the involved devices. This, together with NDN's stateful forwarding, brings several benefits: (1) secured data can be reused by any device that desires the data; (2) packet verification can be performed before applications receive any data, and thus trust management can be enforced starting from the network layer, independent from an individual application's implementation; and (3) cryptographic keys in NDN are named, so their usages can be defined and reasoned about using the semantics embedded in the packet name and producer name, which helps automate trust management [5]. In designing security solutions for home users, we simply cannot overemphasize the importance of usability. Smart-home users will be able to keep their home truly private when a home IoT system is built with stronger resiliency against failures that today's cloud-based IoT systems experience, and with strong cryptographic protection that even the non tech-savvy can manage.

From the network management perspective, bringing the application namespace into the network layer helps intrusion detection. Unlike with numerical source and destination IP addresses, an NDN router can obtain far more insight about its ongoing traffic because of the visible data names and its forwarding state, which helps facilitate intrusion detection and DDoS mitigation [6].

NDN can enable support for an integrated IoT framework. As discussed, NDN integrates security into applications' data packet generation, and these application-generated packets are directly used for network communication. This enables an integrated IoT framework [7] with application support over NDN, without each application having to build their own trust management, access control, and service discovery. With well-defined naming conventions, devices from different vendors and applications developed by different parties can communicate using a unified namespace at the network layer. Also, since all data packets can be verified upon reception, an integrated trust management and access control system can be built below the application layer. From an application developer's perspective, such support can be implemented in an IoT software development kit or library.

There are advantages NDN brings in supporting heterogeneous network connectivity. NDN, being a data-centric network architecture, naturally supports broadcast and multicast-based IoT network systems. Consider a WiFi-connected air conditioner that needs to determine the temperature of a bedroom. Instead of figuring out one of the IP addresses of the temperature sensor in the bedroom (there may be multiple of them) and setting up the connection to request the data, in an NDN-connected home, a node can directly issue an interest "/home/bedroom/temperature" to fetch the temperature data from any temperature sensor which has data under this prefix. With respect to heterogeneous network connectivity support, compared with IP, which is strictly bound to a network interface controller (NIC) (two NICs cannot have the same IP address), an NDN name prefix can be registered on different network interfaces. For example, a forwarding-enabled device, when it hears an interest packet with prefix "/home/bedroom," can forward this interest to all interfaces that are registered with this prefix. Prefix registration and best path selection can be achieved through broadcast-based self-learning (in a similar way to ethernet self-learning).

## LOOKING AHEAD

Future IoT devices will be more powerful, and their increasing capabilities can enable a new generation of applications. Nevertheless, IoT networking and security must be well-defined to serve as the basis for these new opportunities. NDN can help meet these requirements with its built-in security, naming schema, and data-centric networking. With efforts from both industry and academia, NDN and future IoT devices can help humans to achieve a better quality of life.


**References**

[1] Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J.A., Invernizzi, L., Kallitsis, M. and Kumar, D. Understanding the mirai botnet. *26th {USENIX} Security Symposium*, 2017, 1093–1110.

[2] Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Crowley, P., Papadopoulos, C., Wang, L. and Zhang, B. Named data networking. *ACM SIGCOMM Computer Communication Review 44*, 3 (2014), 66-73.

[3] Shelby, Z., Hartke, K., and Bormann, C. The constrained application protocol (CoAP). RFC 7252. June 2014; https://www.rfc-editor.org/info/rfc7252. DOI: 10.17487/RFC7252.

[4] Cheshire, S. and Krochmal, M. DNS-based service discovery.RFC 6763. Feb. 2013; https://www.rfc-editor.org/info/rfc6763. DOI 10.17487/RFC6763.

[5] Yu, Y., Afanasyev, A., Clark, D., Jacobson, V. and Zhang, L. Schematizing trust in named data networking. *Proceedings of the ACM ICN*. (2015).

[6] Zhang, Z, et al. "Expect More from the Networking: DDoS Mitigation by FITT in Named Data Networking." *arXiv preprint arXiv:1902.09033*(2019).

[7] Zhang, Z., Vasavada, V., Osterweil, E. and Zhang, L. NDNoT: a framework for named data network of things. *Proceedings of the ACM ICN*, 200-201 (2018).



**Biographies**

Zhiyi Zhang is a Ph.D. candidate in the Internet Research Lab at UCLA.

Edward Lu is an undergraduate student in the Internet Research Lab at UCLA.

Yu Guan is a Ph.D. candidate from Peking University, China.

Tianxiang Li is a Ph.D. student in the Internet Research Lab at UCLA.

Xinyu Ma is a Ph.D. student in the Internet Research Lab at UCLA.

Zhaoning Kong is a master's student in the Internet Research Lab at UCLA.

Lixia Zhang is the leader of the Internet Research Lab at UCLA, and a Principal Investigator of the Named Data Networking Project.