International Journal of Contemporary Computer Research (IJCCR), Vol.1 Issue.1 (June, 2024)

Adapting TCP/IP for IoT: Challenges, Solutions, and the Role of Information-Centric Network Architecture

Ammar Bathich, Yazeed Al Moaiad
1,2Faculty of Computer & Information Technology
ammar.bathich@mediu.edu.my,
yazeed.alsayed@mediu.edu.my

Abstract— **The "Internet of Things" (IoT), which involves networking a potentially large number of resource-constrained devices, has garnered increasing attention in recent years. Presently, IoT systems predominantly rely on TCP/IP protocols, with a specific emphasis on IPv6. However, empirical evidence suggests that the original design of the TCP/IP protocol stack is not well-suited for the IoT environment. In response, the Internet Engineering Task Force (IETF) has dedicated considerable effort to adapting the protocol stack to align with IoT deployment scenarios. These endeavours have led to augmentations of existing protocols within the TCP/IP suite and the creation of several novel protocols. Despite these modifications, persistent challenges continue to emerge. This paper conducts an analysis of the technical challenges associated with applying the TCP/IP protocol to the IoT environment. Additionally, it provides an overview of diverse solutions proposed by the IETF. The contention put forth is that existing IP-based solutions exhibit either inefficiency or inadequacy in supporting IoT applications. As a proposed alternative, we advocate for the adoption of the Information-Centric Network architecture as a more effective solution to address the complexities of IoT networking.**

Keywords: Internet of Things (IoT), TCP/IP protocols, IPv6, Information-Centric Network architecture

I. OVERVIEW

The term "Internet of Things" (IoT) generally denotes the interconnection of diverse computing devices for monitoring and control applications. Modern IoT systems adopt the open standards of the TCP/IP protocol suite to accommodate device and application heterogeneity. However, IoT networks differ fundamentally from traditional wired computer networks, posing significant challenges to the application of TCP/IP technologies. This paper aims to systematically identify these challenges and articulate future directions to address them.

IoT networks often feature numerous low-end, resource-constrained devices designed with a focus on low manufacturing and operational costs. These devices typically have limited computing power and operate over extended periods on battery power. IoT networks employ low-energy Layer-2 technologies with smaller MTUs and

lower transmission rates, presenting an immediate challenge for IoT network protocol design. Additionally, power-saving measures in IoT nodes, deployment in environments without wired infrastructure, and reliance on wireless mesh technologies further challenge TCP/IP protocol architecture.

The paper discusses specific challenges, including adapting packet size to constrained links, addressing issues related to mesh networks, optimizing broadcast and multicast in battery-powered networks, implementing scalable routing mechanisms, and accommodating diverse data delivery requirements.

Furthermore, IoT applications heavily interact with sensors and actuators, requiring efficient and scalable support for naming configuration, discovery, security, and resource-oriented communication interfaces like REST. Existing solutions, such as DNS-based naming services, content caches, proxies, and channel-based security protocols, face limitations in IoT environments. The paper delves into these issues, exploring why current solutions may be insufficient and providing insights for the design of future IoT network architectures.

The remainder of the paper discusses each identified issue in detail, examining the architectural reasons behind the challenges when applying TCP/IP to the IoT. It surveys existing solutions standardized or under active development at the IETF, analyzing their limitations and offering insights and directions for the design of future IoT network architectures.

II. CHALLENGES IN THE NETWORK LAYER

The Internet Protocol (IP), particularly IPv6, was initially designed for the conventional internet environment,

where desktops and laptops communicated with wire-connected servers. In this section, we explore how the assumptions made by IP regarding host and network properties, which were valid in the context of traditional internet usage, no longer hold in the Internet of Things (IoT) realm. We also examine the adjustments made to IP and its companion protocols to align them with the unique requirements of IoT.

A. Small MTU

IoT networks frequently feature constrained, low-energy links with very small Maximum Transmission Units (MTUs). For instance, the IEEE 802.15.4-2006 standard specifies a maximum physical layer frame size of merely 127 bytes. This stands in stark contrast to current IP networks, which assume a minimum MTU of 1500 bytes or higher. IPv6, designed long before the conception of IoT, presents challenges for small-MTU links due to its fixed-length header and the requirement for a minimum MTU size of 1280 bytes. To address this, 6LoWPAN introduces an adaptation layer between the link and network layers, implementing header compression and link-layer fragmentation to alleviate protocol overhead and provide the illusion of a larger MTU size. However, these adaptations introduce complexity and overhead, highlighting the mismatch between the original design and IoT requirements.

B. Multi-link Subnet

The existing subnet model of IPv4 and IPv6 assumes two types of Layer-2 networks: multi-access links and point-to-point links. However, IoT mesh networks form a collection of Layer-2 links without intervening Layer-3 devices, creating a multi-link subnet model not anticipated by the original IP addressing architecture. This mismatch leads to technical challenges related to TTL/Hop-Limit handling and link-scoped multicast, disrupting legacy protocols like ARP, DHCP, and Neighbor Discovery. Resolving these issues necessitates either relying on Layer-2 mechanisms to transparently unite multiple links or partitioning the mesh network into multiple subnets, each with its prefixes, introducing additional complexity in network configuration.

C. Multicast Efficiency

While many IP-based protocols rely on IP multicast for group notifications and queries, supporting multicast in constrained IoT mesh networks poses challenges. Issues include the lack of link-layer ACK for multicast, variations in data transmission rates among recipients, intermittent node sleeping modes, and the need for multicast packets to traverse multiple hops, potentially overloading network resources. Redesigning legacy protocols to minimize multicast use becomes essential for effective application in constrained IoT environments.

D. Mesh Network Routing

Typical IoT networks exhibit either star or peer-to-peer (mesh) topologies, with routing configurations differing for each. Star networks involve a hub node acting as a default gateway, suitable for limited deployment scales. In contrast, mesh networks enable larger coverage but require efficient routing mechanisms. Two approaches, mesh-under and route-over, address routing at the link and network layers, respectively. IEEE 802.15.5 supports link-layer routing for mesh networks, while the IETF's RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) serves as the standard solution for route-over routing. Both approaches present challenges related to address allocation, routing table maintenance, and header size, emphasizing the persistent routing challenges in IP-based IoT mesh technologies.

III. Challenges in the Transport Layer

In the TCP/IP architecture, the transport layer, primarily managed by TCP, is responsible for congestion control and reliable delivery, particularly suited for long-lived point-to-point connections with minimal latency requirements. However, TCP faces inefficiencies when dealing with diverse communication patterns inherent in IoT applications. Challenges include the impracticality of maintaining long-lived connections due to energy constraints, the unacceptable overhead of establishing connections for small data amounts, and the low-latency requirements of certain applications. In lossy wireless networks, TCP's in-order delivery and retransmission mechanisms may lead to head-of-line blocking, causing unnecessary delays. Additionally, link-layer automatic repeat request (ARQ) in wireless MAC protocols may further impact TCP performance. While some industrial IoT standards still mandate TCP support, an increasing number of IoT protocols, such as BACnet/IP and CoAP, opt for integrating transport functionalities into the application layer using UDP. This transformation turns the transport layer into a multiplexing module, highlighting the need for application-level framing to embed application semantics into network-level packets, a feature lacking in the current TCP/IP architecture.

IV. Challenges in the Application Layer

Most IoT applications follow a resource-oriented request-response model similar to today's web services using the REST architecture. The Constrained Application Protocol (CoAP), a UDP-based protocol, has been developed to facilitate REST-style communication for IoT applications. However, gaps in lower layers of the TCP/IP architecture, including resource discovery, caching, and security, prompt the implementation of REST at the application layer.

A. Resource Discovery

Resource-oriented communication requires a resource discovery mechanism. Traditional IP networks employ DNS-based Service Discovery (DNS-SD), but this falls

short in supporting IoT applications. IoT resource discovery demands a more generalized approach, with CoAP adopting a URI-based naming scheme to identify resources. CoRE-RD, a CoAP-based resource discovery mechanism, utilizes less constrained resource directory (RD) servers to store metainfo about resources hosted on devices. Synchronization mechanisms like MPL and the Home Networking Control Protocol (HNCP) address challenges posed by link-local multicast inefficiencies. An efficient IoT network architecture should incorporate resource discovery as a core functionality.

B. Caching

The TCP/IP communication model assumes simultaneous online presence of both client and server. However, in IoT scenarios, constrained devices frequently enter sleep mode, leading IoT applications to rely on caching and proxying for efficient data dissemination. Application-level caching implemented by CoAP and HTTP has limitations in dynamic network environments. Pervasive opportunistic caching within the network, integrated into the forwarding process, and a fundamental change in the security model are needed for efficient and flexible caching in the IoT environment.

C. Security

Security is crucial for IoT applications interacting with the physical world. The prevalent channel-based security model (e.g., TLS and DTLS) introduces overhead in establishing secure channels, requires maintaining channel states, and does not ensure end-to-end security once data exits the channel. Object-based security, an alternative proposed at the IETF, secures the application data unit directly, providing necessary authentication information. This model addresses limitations associated with channel-based security, offering a more suitable approach for securing IoT applications.

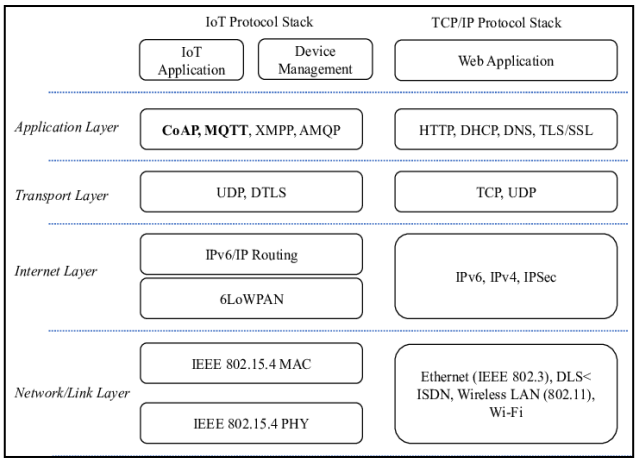

Figure 1: A typical architecture for IoT systems

V. REBUILD THE ARCHITECTURE

The well-known maxim of indirection suggests that "all problems in computer science can be solved by another level of indirection." However, it doesn't address the challenge of dealing with an excessive number of levels of indirection, precisely the situation in the current IoT network architecture.

In Figure 1, we observe the layered structure of an IP-based IoT stack. Typically, IoT applications, aiming to support the REST interface, utilize messaging protocols like CoAP or HTTP. These applications often interact with common services situated above the messaging layer, such as the CoAP Resource Directory and object security support. The transport layer incorporates TLS and DTLS to secure the communication channel. Furthermore, various infrastructural services, including ICMP, DHCP, Neighbour Discovery (ND), DNS, and RPL, are essential to facilitate IP network communications.

The layered structure of an IP-based IoT (Internet of Things) stack typically follows the principles of the OSI (Open Systems Interconnection) model, which consists of several distinct layers, each responsible for specific functions. Application Layer is the top layer, responsible for defining how IoT devices interact with applications and services. It includes protocols for various IoT applications, such as CoAP (Constrained Application Protocol) or HTTP (Hypertext Transfer Protocol). Sitting just below the application layer, the transport layer ensures end-to-end communication between devices. In the context of IP-based IoT, this layer often involves the use of TCP (Transmission Control Protocol) or UDP (User Datagram Protocol).

Security is a critical concern in IoT, and this layer includes protocols like TLS (Transport Layer Security) and DTLS (Datagram Transport Layer Security) to establish secure communication channels between devices. The network layer manages the routing of data packets between devices. In IP-based IoT, it involves the use of IP (Internet Protocol) for addressing and routing. Infrastructure Layer includes various infrastructure services that facilitate communication within the IP network. Common protocols in this layer include ICMP (Internet Control Message Protocol), DHCP (Dynamic Host Configuration Protocol), Neighbour Discovery (ND), DNS (Domain Name System), and RPL (Routing Protocol for Low-Power and Lossy Networks).

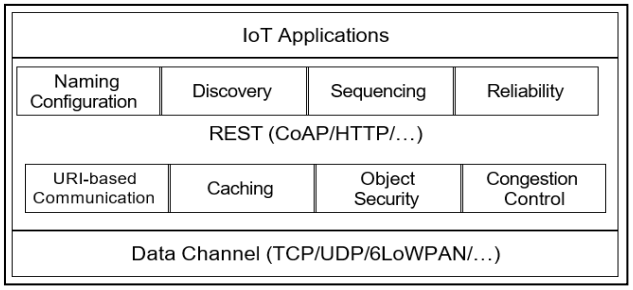

Figure 2: An IoT stack from the application's perspective

A re-evaluation of the network stack from the application's perspective, focusing on core functionalities, yields a different picture, as shown in Figure 2. Unlike the "everything over IP" paradigm, IoT applications converge on a new paradigm of "everything over REST." At the base, an IoT stack may utilize diverse data transport options like UDP and 6LoWPAN. In the centre of the stack, a RESTful messaging protocol encompasses all service components operating over a unified abstraction of the application data unit (ADU) defined by IoT applications. This re-evaluation reveals a fundamental misalignment between the expectations of IoT applications and the current architectural layered view.

From the application's perspective, an IoT (Internet of Things) stack is designed to facilitate communication and interaction between IoT devices and applications. The architecture may be viewed as a stack of layers, each serving specific functions to enable seamless connectivity and data exchange. Here's an overview of an IoT stack from the application's perspective:

At the top of the stack are IoT applications that define the specific use cases and functionalities. These applications interact with the underlying layers to send and receive data from IoT devices. This layer implements a RESTful messaging protocol, such as CoAP (Constrained Application Protocol) or HTTP (Hypertext Transfer Protocol), to enable communication between IoT devices and applications. It provides a standardized way for applications to request and exchange data. Common services, like the CoAP Resource Directory, are often included in this layer. These services assist in resource discovery, security support, and other common functionalities.

The transport layer handles the actual transfer of data between devices. Protocols like UDP (User Datagram Protocol) and 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) are commonly used to transmit data efficiently in IoT environments. Security is crucial in IoT applications. This layer includes mechanisms for ensuring the integrity and confidentiality of individual Application Data Units (ADUs). It may involve digital signatures and encryption for secure data transmission.

Congestion Control Module layer may implement congestion control algorithms to manage data flow efficiently in diverse network environments. It adapts to different conditions and ensures optimal communication. Naming Configuration: Naming configurations and resource discovery mechanisms assist in identifying and locating IoT resources within the network. This is crucial for IoT applications to operate effectively. Large data that cannot fit into a single Application Data Unit (ADU) may be chopped into smaller segments using a sequencing mechanism, this ensures efficient data handling. To meet application demands, this layer supports packet retransmission and ordering, ensuring reliable data transmission.

By considering the IoT stack from the application's perspective, emphasis is placed on the functionalities required for efficient communication, security, and management of IoT data in diverse and dynamic network environments. The design aligns with RESTful principles and seeks to address the specific needs of IoT applications in a layered and modular structure.

The REST layer includes several sub-modules implementing critical functionalities:

1. A URI-based communication mechanism delivering application-layer data to network destinations.
2. A caching mechanism for efficient data dissemination.
3. An object security mechanism for protecting the integrity and confidentiality of individual ADUs.
4. A congestion control module implementing multiple algorithms for different network environments.
5. Naming configuration and resource discovery for assisting application operations.
6. A sequencing mechanism for chopping large data that cannot fit into a single ADU.
7. A reliability mechanism supporting packet retransmission and ordering according to the application’s demand.

Currently, all these functionalities (including the REST interface itself) are implemented by application layer protocols. However, some functionalities could be more effective if moved into the core network. For example, congestion control could benefit from feedback from network and link layers, and caching could be more efficient if caches are ubiquitous inside the network. To utilize in-network caching, URI-based forwarding, REST interface, and object security should also be supported at the network layer. This protocol stack optimization leads to a simpler and more efficient architecture, resembling the Information-Centric Network (ICN) vision.

ICN architectures like NDN [16, 31] not only provide native support for functionalities demanded by IoT applications but also address lower-layer network challenges. They apply the same ADU across layers, giving packet flow control back to applications. ICN does not impose artificial requirements on minimum MTU, and its simplified stack reduces packet header size. It is inherently multicast-friendly due to pervasive caching, and its data-oriented communication avoids addressing and routing issues for sensor nodes. Data-centric security avoids the overhead of channel-based security solutions, fitting IoT devices with limited resources and intermittent connectivity. The architectural simplicity leads to smaller code size for application software, lower energy and memory footprint

for devices, and better utilization of network resources compared to the current IP-based IoT stack. The potentials of IoT over ICN have garnered attention in the IRTF ICNRG [32], and we anticipate it becoming an active research topic as interest in IoT technologies continues to grow.

VI. CONCLUSION

When the TCP/IP protocol stack was initially developed in the early 1980s, the aim was to connect mainframe computers through wired connectivity. Although the protocol stack evolved post the IP specification, the fundamental assumption behind the architecture design remained unchanged. IoT networks represent a new application type where the IP architecture cannot easily fit without significant modifications to the protocol stack.

In this paper, we explored the challenges of applying TCP/IP to IoT networks arising from the network and transport layers. We discussed how application layer protocols like CoAP provide solutions for functionalities that lower layers fail to support. The mismatch was made more evident by comparing the current IoT stack with the desired architecture from the application's perspective. We proposed an architectural change moving REST-related components into the core network layer, eventually arriving at a more efficient architecture compared to existing application layer solutions. This new IoT stack would embrace the ICN design and implement required functionalities natively and more efficiently inside the network.

REFERENCES

[1] Z. Sheng et al., "A Survey on the IETF Protocol Suite for the Internet ofThings: Standards, Challenges, and Opportunities," IEEE Wireless Commun., vol. 20, no. 6, 2013, pp. 91–98.

[2] B. Ahlgren et al., "A Survey of Information-Centric Networking," IEEE Commun. Mag., vol. 50, no. 7, 2012, pp. 26–36.

[3] A. Lindgren et al., "Applicability and Trade-Offs of Information-Centric Networking for Effi cient IoT," IETF Internet Draft, Jan. 2015.

[4] Y. Zhang et al., "ICN Based Architecture for IoT — Requirements and Challenges," IETF Internet Draft, Nov. 2014.

[5] J. Quevedo, D. Corujo, and R. Aguiar, "Consumer Driven Information Freshness Approach for Content Centric Networking," Proc. IEEE NOM, 2014.

[6] M. Amadeo et al., "Named Data Networking for IoT: An Architectural Perspective," Proc. European Conf. Networks and Commun., Bologna, Italy, 2014.

[7] E. Baccelli et al., "Information Centric Networking in the IoT: Experiments with NDN in the Wild," ACM Conf. Information-Centric Networking, 2014.

[8] S. Li et al., "A Comparative Study of MobilityFirst and NDN Based ICN-IoT Architectures," Proc. 10th IEEE Int'l. Conf. Heterogeneous Networking for Quality, Reliability, Security and Robustness, 2014, pp. 158–63.

[9] K. V. Katsaros et al., "Information-Centric Networking for Machine-toMachine Data Delivery: A Case Study in Smart Grid Applications," IEEE Network, vol. 28, no. 3, 2014, pp. 58–64.

[10] N. Fotiou and G. C. Polyzos, "Realizing the Internet of Things Using Information-Centric Networking," Proc. 10th IEEE Int'l. Conf. Heterogeneous Networking for Quality, Reliability, Security and Robustness, 2014, pp. 193–94.

[11] W. Shang et al., "Securing Building Management Systems Using Named Data Networking," IEEE Network, vol. 3, no. 28, 2014, pp. 50–56.

[12] J. Burke et al., "Securing Instrumented Environments over Content-Centric Networking: The Case of Lighting Control and NDN," Proc. IEEE NOMEN Wksp., 2013.

[13] J. Burke et al., "Secure Sensing over Named Data Networking," Proc. IEEE Network Computing and Applications, 2014, pp. 175–80.

[14] S. Vural et al., "In-Network Caching of Internet-of-Things Data," Proc. IEEE ICC, 2014.

[15] X. Vasilakos, K. Katsaros, and G. Xylomenos, "Cloud Computing for Global Name-Resolution in Information-Centric Networks," Proc. 2nd Symp. Network Cloud Computing and Applications, 2012, IEEE, 2012, pp. 88–94.

[16] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking Named Content. In Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT '09, pages 1–12, New York, NY, USA, 2009. ACM.

[17] C. A. Kent and J. C. Mogul. Fragmentation considered harmful. SIGCOMM Comput. Commun. Rev., 25(1):75–87, Jan. 1995.

[18] E. Kim, D. Kaspar, C. Gomez, and C. Bormann. Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing. RFC 6606 (Informational), May 2012.

[19] N. Kushalnagar, G. Montenegro, and C. Schumacher. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. RFC 4919 (Informational), Aug. 2007.

[20] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944 (Proposed Standard), Sept. 2007. Updated by RFCs 6282, 6775.

[21] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor Discovery for IP version 6 (IPv6). RFC 4861 (Draft Standard), Sept. 2007. Updated by RFCs 5942, 6980, 7048.

[22] E. Rescorla and N. Modadugu. Datagram Transport Layer Security Version 1.2. RFC 6347 (Proposed Standard), Jan. 2012.

[23] G. Selander, J. Mattsson, F. Palombini, and L. Seitz. Object Security for CoAP. draft-selander-ace-object-security-03 (work in progress), Oct. 2015.

[24] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann. Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs). RFC 6775 (Proposed Standard), Nov. 2012.

[25] Z. Shelby, K. Hartke, and C. Bormann. The Constrained Application Protocol (CoAP). RFC 7252 (Proposed Standard), June 2014.

[26] Z. Shelby, M. Koster, C. Bormann, and P. van der Stok. CoRE Resource Directory. draft-ietf-core-resource-directory-05 (work in progress), Oct. 2015.

[27] M. Stenberg and S. Barth. Distributed Node Consensus Protocol. draft-ietf-homenet-dncp-12 (work in progress), Nov. 2015.

[28] M. Stenberg, S. Barth, and P. Pfister. Home Networking Control Protocol. draft-ietf-homenet-hncp-10 (work in progress), Nov. 2015.

[29] D. Thaler. Multi-Link Subnet Issues. RFC 4903 (Informational), June 2007.

[30] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550 (Proposed Standard), Mar. 2012.

[31] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, kc claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh. Named Data Networking (NDN) Project. Technical Report NDN-0001, October 2010.

[32] Y. Zhang, D. Raychadhuri, L. Grieco, E. Baccelli, J. Burke, R. Ravindran, and G. Wang. ICN based Architecture for IoT - Requirements and Challenges. draft-zhang-iot-icn-challenges-02 (work in progress), Aug. 2015.