Architectures for Internet Connectivity

Kashyap, Shruthi; Rao, Vijay; Venkatesha Prasad, Ranga Rao; Staring, Toine

DOI
10.1007/978-3-030-85836-0_3

Publication date
2021

Document Version
Final published version

Published in
SpringerBriefs in Applied Sciences and Technology

Chapter 3
Architectures for Internet Connectivity

In the cordless kitchen, the appliances should be able to connect to the Internet when they are powered, i.e. when they are placed on top of the PTx. It has to be ensured that the appliances maintain the Internet connectivity as long as they are powered on, irrespective of what communication interface is used between the PTx and the appliance. Internet connectivity is not required when they are away from the PTx. One solution to providing connectivity is to install Wi-Fi modules in the appliances. However, there would be drawbacks as mentioned in Chap. 1, which are summarized here:

(a) The Wi-Fi module gets powered only when the appliance is placed on top of the PTx. Powering the appliances with batteries is not desirable as batteries need to be regularly charged and/or replaced.
(b) When the PTx goes into standby, the appliance will be switched off and the Wi-Fi module in the appliance will not be awake to support Internet connectivity. The PTx could supply standby power to the appliance, however, this will not be efficient in terms of power consumption.
(c) The cost of the appliances would increase due to the additional Wi-Fi module and battery.

Hence, having a dedicated Wi-Fi module for every appliance will be unnecessary. To overcome these drawbacks, the Wi-Fi module could be installed in the PTx (or kitchen countertop) instead, and this connection could be shared by all the appliances that use the PTx. To enable this, the existing NFC channel between the appliance and PTx could be used for transmitting the Internet-related information so that the cordless appliances are indirectly connected to the network. Using this solution, the PTx can keep its Wi-Fi module on during standby and wake up the appliance whenever it receives a message for the appliance from a remote user. This would also reduce the cost of the appliance.

Based on this solution, two main architectures can be considered for Internet connectivity: Proxy architecture and Bridge architecture. Both these architectures are to support the TCP/IP protocol. TCP is chosen as the transport layer protocol because

S. Kashyap et al., *Cook Over IP*,
SpringerBriefs in Applied Sciences and Technology,
https://doi.org/10.1007/978-3-030-85836-0_3

the Internet applications of the cordless kitchen like remote user control, recipe and software uploads require reliable connections. TCP is best suited for such scenarios as it provides a reliable, ordered and error-checked delivery of packets between communicating applications. This chapter explains and evaluates these architectures by discussing their advantages and disadvantages in detail.

3.1 Proxy Architecture

In this architecture, the PTx is installed with a Wi-Fi module or Ethernet connection, so it holds the full TCP/IP stack required for Internet connectivity. The cordless appliance only implements the application layer and sends its application data to the PTx through the NFC channel. The PTx acts like a proxy to the appliance by processing the TCP/IP packets for it, as shown in Fig. 3.1.

In the proxy architecture, the PTx represents the appliance on the Internet. So the TCP session initiation/termination, data packet processing and acknowledgment handling are done by the PTx, as shown in Fig. 3.2. The appliance only sends/receives the application payload. When the PTx receives a TCP/IP packet from the end-user device, it immediately sends out an ACK to the end-user device and then sends the application data to the appliance. It does not wait to check if the application data is delivered correctly to the appliance. Advantages of using this architecture are listed below.

- The implementation of the appliance is simple as it only needs the application layer.
- There is less load on the NFC channel as the appliance sends/receives only the application data. This results in lower system latency.
- Lower cost of the appliance as there is no Wi-Fi module or battery.

However, this architecture has the following disadvantages:

- Reliability is dependent on the PTx implementation because the PTx is responsible for creating and processing the TCP/IP packets of the appliance.

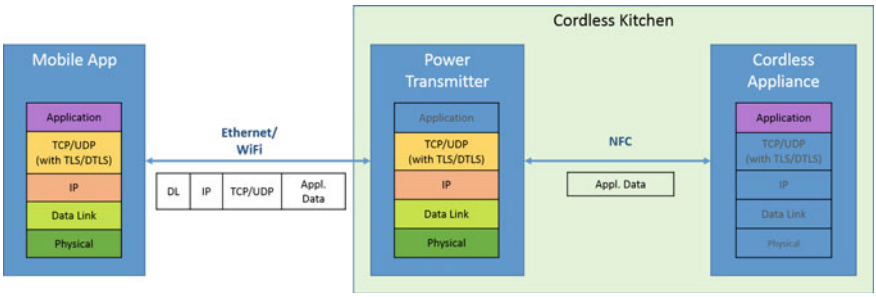

Fig. 3.1 Proxy architecture for Internet connectivity

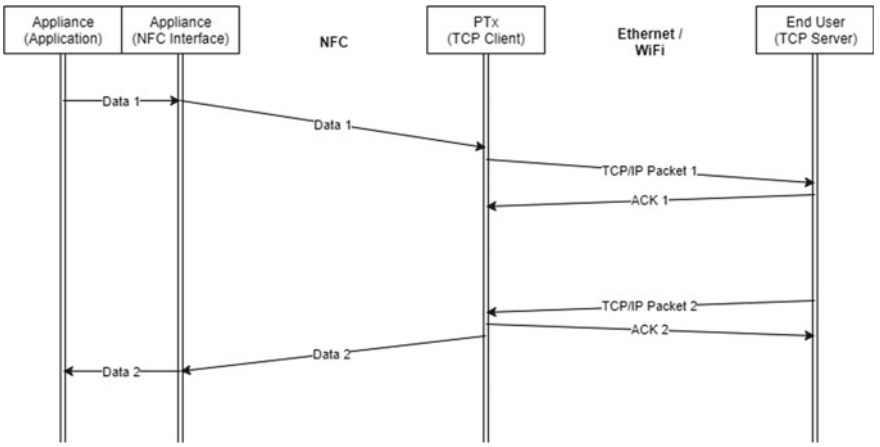

Fig. 3.2 TCP sequence diagram of proxy architecture

- The PTx sends an ACK irrespective of whether the data is delivered to the appliance or not. A special handshake mechanism could be implemented where the PTx waits until the appliance sends an ACK for the data it received. This would increase the latency and also the complexity of implementation.
- The data is not end-to-end protected by the appliance. When the PTx and the appliance are from different manufacturers, the appliance needs to trust the PTx with its application data. There would be possibilities of PTx using the appliance's data for its business purpose without the consent of the user, for example, analyzing user behavior, extracting the appliance's implementation details and sending the packets to a malicious server/user. It is possible to use data encryption techniques to increase the security, however, the PTx would still have the control of processing the TCP/IP packets of the appliance.
- Another disadvantage could be that the PTx manufacturers might not be willing to implement this architecture. There are no advantages for the PTx in this architecture because it only acts like a proxy and has the burden of processing Internet packets for the appliance.
- Appliance is not visible on the network as it does not have its own IP address.

3.2 Bridge Architecture

In this architecture, the PTx contains the Wi-Fi module or Ethernet connection but it acts like a bridge by processing only the data link and the physical layers for the appliance. The higher layers of the TCP/IP stack are implemented by the appliance, as shown in Fig. 3.3. Thus, the appliance does not have to depend on the PTx for TCP/IP packet processing.

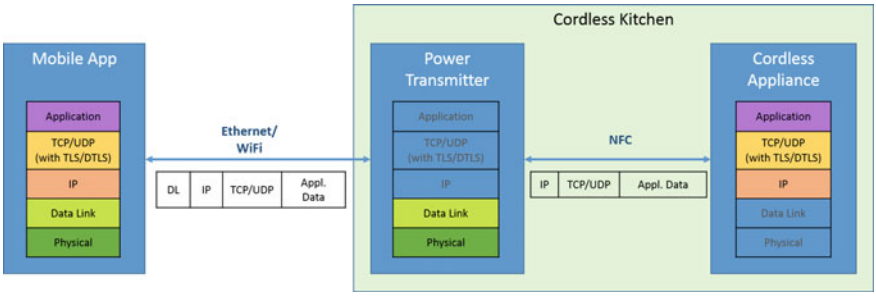

Fig. 3.3 Bridge architecture for Internet connectivity

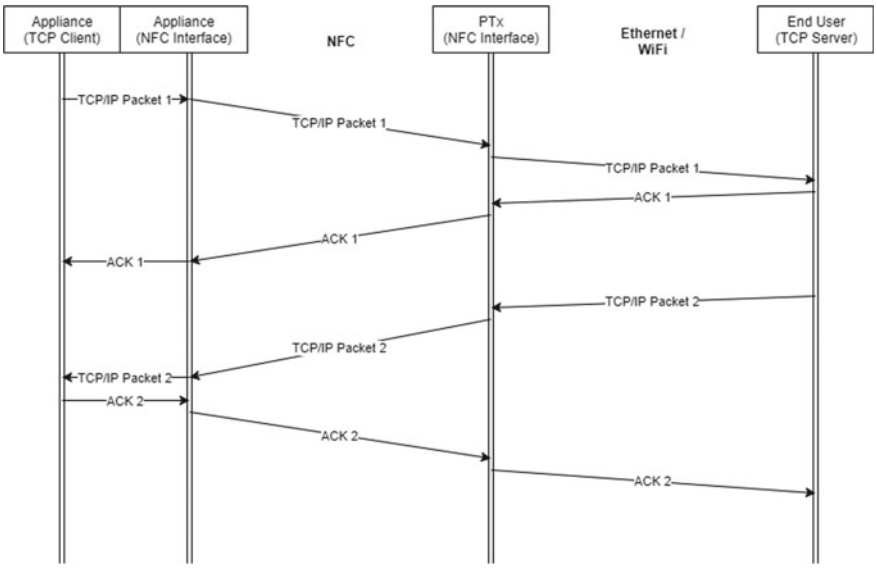

Fig. 3.4 TCP sequence diagram of bridge architecture

Figure 3.4 shows an example TCP sequence diagram using the bridge architecture. In this, the appliance is visible in the network and has a TCP/IP stack of its own. It is completely responsible for TCP session initiation/termination, data packet processing and acknowledgment handling. The PTx merely acts like a bridge by forwarding the appliance's packets to the end-user device. The advantages of using this architecture are as follows:

- The appliance has more control in the process of Internet connectivity. It is only dependent on the PTx for forwarding its TCP/IP packets.
- The data communication can be made more secure by using cryptographic protocols like the Transport Layer Security (TLS) in the appliance stack to ensure data privacy.

- The burden on the PTx is less as it does not have to process the TCP/IP packets for the appliance.
- The appliance will be visible on the network as it will have its own IP address.

Some of the disadvantages of this architecture are listed below.

- The load on the NFC channel increases due to the overhead introduced by the TCP/IP protocol. This will have a large impact on the latency of the system. As the Internet applications of the cordless kitchen are soft and firm real time, it is very important to have minimal latency and a good response time in the applications. Packet compression techniques could be employed to reduce the latency.
- The implementation of the appliance would be complex due to packet processing and tunneling of the TCP/IP protocol over the NFC channel.

3.3 Comparison of Transmission Latency

The size of the application data in the cordless kitchen depends on the kitchen UI protocol being used. A proprietary protocol called Digital Innovation Communications (DICOMM) Protocol is used for the experiments. The approximate message sizes in the JavaScript Object Notation (JSON)-based variant and the Binary variant of the protocol are shown in Table 3.1.

Figure 3.5 shows the latencies of data exchange using the proxy and bridge architectures at an NFC bit rate of 83.2 kbps in the time-slotted mode. A TCP session exchanging a single data packet is considered for the bridge architecture, and the 6LoWPAN header compression results given in [1] are used. It can be seen that without compression, the latency in the bridge architecture is around 170 ms higher than that of the proxy architecture. This is because of the overhead introduced by the TCP/IP protocol. This overhead remains constant for all data sizes, so it will be less significant at higher sizes. For a data size of 1024 bytes, the latency with the bridge architecture is about 44.6% more than that of the proxy architecture, and with header compression this difference is reduced to about 36.6%. It is possible to have long TCP sessions equal to the duration of the cooking session in order to

Table 3.1 Internet application message sizes using the DICOMM UI protocol

Message type	Message size (Bytes)	
	JSON protocol variant	Binary protocol variant
Switch on/off, Set time/temperature, Keep warm on/off, etc.	30–100	10–35
Status information/Notification	250–300	75–100
Recipe upload	350–1000	125–350

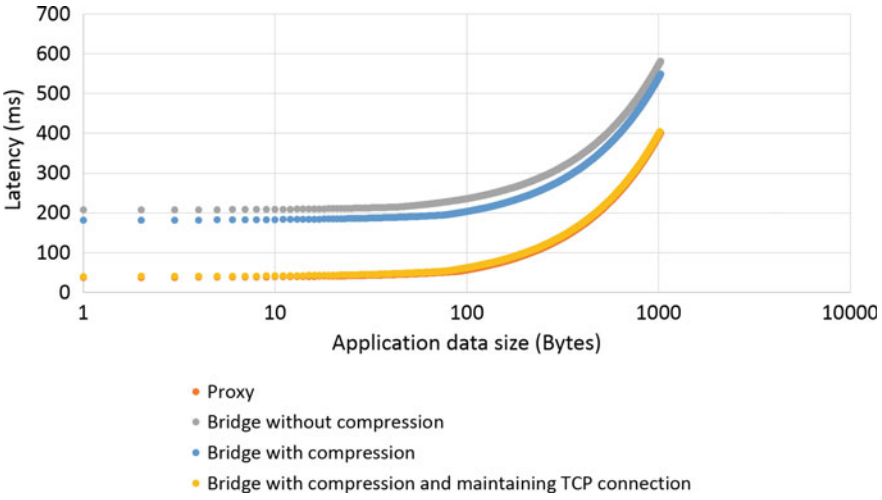

Fig. 3.5 Transmission latency for different application data sizes using proxy and bridge architectures

avoid executing the TCP handshake and termination procedures frequently. In such scenarios the latency obtained with header compression will be very close to that of the proxy architecture as seen in Fig. 3.5.

Reference

1. J. Hui, P. Thubert, Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. RFC **6282**, 1–24 (2011)