UNIT :3
THENETWORKLAYER

Constrained Nodes

In IoT solutions, different classes of devices coexist. Depending on its functions in anetwork, “thing” architecture may or may not offer similar characteristics compared to agenericPC orserver in an ITenvironment.

Another limit is that this network protocol stack on an IoT node may be required tocommunicate through an unreliable path. Even if a full IP stack is available on the node, thiscauses problems such as limited or unpredictable throughput and low convergence when atopologychangeoccurs.

Finally, power consumption is a key characteristic of constrained nodes. Many IoTdevices are battery powered, with lifetime battery requirements varying from a few monthsto 10+ years. This drives the selection of networking technologies since high-speed ones,such as Ethernet, Wi-Fi, and cellular, are not (yet) capable of multi-year battery life. Currentcapabilities practically allow less than a year for these technologies on battery-powerednodes. Of course,power consumptionis muchless of a concern on nodes that do notrequirebatteriesas anenergy source.

The power consumption requirements on battery-powered nodes impact communicationintervals. To help extend battery life, one could enable a “low-power” mode instead of onethatis“alwayson.”Anotheroptionis“alwaysoff,”whichmeanscommunicationsareenabledonly whenneededtosenddata.

While it has been largely demonstrated that production IP stacks perform well inconstrainednodes.IoTconstrainednodes canbeclassifiedas follows:

- **Devices that are very constrained in resources, may communicate infrequently totransmit a few bytes, and may have limited security and management capabilities:**ThisdrivestheneedfortheIPadaptationmodel,wherenodescommunicatethroughgateways andproxies.
- **Devices with enough power and capacities to implement a stripped-down IP stackor non- IP stack:** In this case, you may implement either an optimized IP stack anddirectly communicate with application servers (adoption model) or go for an IP ornon-IPstackandcommunicatethroughgatewaysandproxies (adaptationmodel).
- **Devices that are similar to generic PCs in terms of computing and power resourcesbuthaveconstrainednetworkingcapacities,suchasbandwidth:**ThesenodesusuallyimplementafullIPstack(adoptionmodel),butnetworkdesignandapplication behaviorsmustcopewiththebandwidthconstraints.

The definition of constrained nodes is evolving. The costs of computing power, memory,storageresources,andpowerconsumptionaregenerallydecreasing.Atthesame time, networking technologies continue to improve and offer more bandwidth and reliability. Inthefuture,the pushtooptimizeIPforconstrainednodeswilllessenastechnologyimprovementsandcostdecreasesaddressmanyofthesechallenges.

ConstrainedNetworks

In the early years of the Internet, network bandwidth capacity was restrained due totechnical limitations. Connections often depended on low-speed modems for transferringdata. However, these low-speed connections demonstrated that IP could runover low-bandwidth networks.

Buttoday,theevolutionofnetworkinghasseentheemergenceofhigh-speedinfrastructures. However, high-speed connections are not usable by some IoT devices in thelast mile. The reasons include the implementation of technologies with low bandwidth,limited distance and bandwidth due to regulated transmit power, and lack of or limitednetworkservices.

When link layer characteristics that we take for granted are not present, the networkis constrained. A constrained network can have high latency and a high potential for packetloss. Constrained networks have unique characteristics and requirements. In contrast withtypical IP networks, where highly stable and fast links are available, constrained networksarelimitedbylow-power,lowbandwidthlinks(wirelessandwired).Theyoperatebetweena few kbps and a few hundred kbps and may utilize a star, mesh, or combined networktopologies,ensuring properoperations.

With a constrained network, in addition to limited bandwidth, it is not unusual forthe packet deliveryrate (PDR) to oscillate between low and high percentages. Large burstsof unpredictable errors and even loss of connectivity at times may occur. These behaviourscan be observed on both wireless and narrowband power-line communication links, wherepacketdeliveryvariationmay fluctuategreatly duringthecourseofaday.

Unstable link layer environments create other challenges in terms of latency andcontrol plane reactivity. One of the golden rules in a constrained network is to “underreactto failure.” Due to the low bandwidth, a constrained network that overreacts can lead to anetworkcollapse—whichmakes theexisting problemworse.

Control plane traffic must also be kept at a minimum; otherwise, it consumes thebandwidththatisneededbythedatatraffic.Finally,onehastoconsiderthepowerconsumption in battery-powered nodes.Any failure or verbose control plane protocol mayreducethelifetimeofthebatteries.

Tosummarize,constrainednodesandnetworksposemajorchallengesforIoTconnectivity in the last mile. This in turn has led various standards organizations to work onoptimizingprotocols forIoT.

IPVersions

For 20+years, theIETFhasbeenworkingontransitioningtheInternetfromIPversion 4 to IP version 6. The main driving force has been the lack of address space in IPv4 asthe Internet has grown. IPv6 has a much larger range of addresses that should not beexhausted for the foreseeable future. Today, both versions of IP run over the Internet, butmosttrafficis still IPv4based.

While it may seem natural to base all IoT deployments on IPv6, you must take intoaccount current infrastructures and their associated lifecycle of solutions, protocols, andproducts.IPv4isentrenchedinthesecurrentinfrastructures,andsosupportforitisrequired in most cases. Therefore, the Internet of Things has to follow a similar path as theInternetitselfandsupportbothIPv4andIPv6versions concurrently.

Techniques such as tunnelling and translation need to be employed in IoT solutionsto ensure interoperability between IPv4 and IPv6. A variety of factors dictate whether IPv4,IPv6, or both can be used in an IoT solution. Most often these factors include a legacyprotocol or technology that supports only IPv4. Newer technologies and protocols almostalways support both IP versions. The following are some of the main factors applicable toIPv4andIPv6supportinanIoTsolution:

- **ApplicationProtocol:**IoT devicesimplementingEthernetorWi-Fi interfacescancommunicate over both IPv4 and IPv6, but the application protocol may dictate thechoice of the IP version. For example, SCADA protocols such as DNP3/IP (IEEE 1815),Modbus TCP, or the IEC 60870-5-104 standards are specified only for IPv4. So, thereare no known production implementations by vendors of these protocols over IPv6today.ForIoTdeviceswithapplicationprotocolsdefinedbytheIETF,suchasHTTP/HTTPS, CoAP, MQTT, and XMPP, both IP versions are supported. The selectionoftheIPversionisonly dependentontheimplementation.
- **Cellular Provider and Technology:** IoT devices with cellular modems are dependenton the generation of the cellular technology as well as the data services offered bythe provider. For the first three generations of data services—GPRS, Edge, and 3G—IPv4isthebaseprotocolversion.Consequently,ifIPv6isusedwiththesegenerations, it must be tunneled over IPv4. On 4G/LTE networks, data services canuseIPv4orIPv6as abaseprotocol,depending ontheprovider.

- **SerialCommunications:**Manylegacydevicesincertainindustries,suchasmanufacturing and utilities, communicate through serial lines. Data is transferredusing either proprietary or standards based protocols, such as DNP3, Modbus, or IEC60870-5-101. In the past, communicating this serial data over any sort of distancecould be handled by an analog modem connection. However, as service providersupport for analog line services has declined, the solution for communicating withthese legacy devices has been to use local connections. To make this work, youconnect the serial port of the legacy device to a nearby serial port on a piece ofcommunications equipment, typically a router. This local router then forwards theserial trafficover IP tothe central serverforprocessing. Encapsulation of serialprotocols over IP leverages mechanisms such as raw socket TCP or UDP. While rawsocket sessions can run over both IPv4 and IPv6, current implementations are mostlyavailableforIPv4only.
- **IPv6 Adaptation Layer:** IPv6-only adaptation layers for some physical and data linklayers for recently standardized IoT protocols support only IPv6. While the mostcommonphysicalanddatalinklayers(Ethernet,Wi-Fi,andsoon)stipulateadaptationlayersforbothversions,newertechnologies,such asIEEE802.15.4(Wireless Personal Area Network), IEEE 1901.2, and ITU G.9903 (Narrowband PowerLine Communications) only have an IPv6 adaptation layer specified. This means thatany device implementing a technology that requires an IPv6 adaptation layer mustcommunicate over an IPv6-only sub network. This is reinforced by the IETF routingprotocolforLLNs, RPL,which is IPv6only.

6LoWPAN

While the Internet Protocol is key for a successful Internet of Things, constrained nodes andconstrained networks mandate optimization at various layers and on multiple protocols ofthe IP architecture. Some optimizations are already available from the market or underdevelopment by the IETF. Figure 2.12 highlights the TCP/IP layers where optimization isapplied.

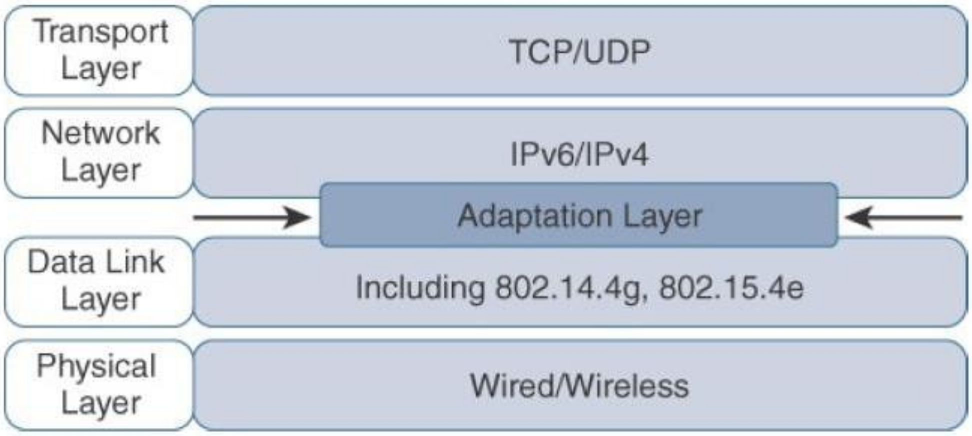

Figure2.12:OptimizingIPforIoTUsinganAdaptationLayer

In the IP architecture, the transport of IP packets over any given Layer 1 (PHY) andLayer 2 (MAC) protocol must be defined and documented. The model for packaging IP intolower-layerprotocols isoftenreferredtoas an *adaptationlayer*.

Unless the technology is proprietary, IP adaptation layers are typically defined by anIETF working group and released as a Request for Comments (RFC). An RFC is a publicationfromtheIETFthatofficiallydocumentsInternetstandards,specifications,protocols,procedures,andevents.Forexample,RFC864describeshowanIPv4packetgetsencapsulated over an Ethernet frame, and RFC 2464 describes how the same function isperformedforanIPv6packet.

IoT-relatedprotocolsfollowasimilarprocess.Themaindifferenceisthatanadaptation layer designed for IoT may include some optimizations to deal with constrainednodes and networks.The main examples of adaptation layers optimized for constrainednodes or “things” are the ones under the 6LoWPANworking group andits successor, the6Loworking group.

The initial focus of the 6LoWPAN working group was to optimize the transmission ofIPv6packetsoverconstrainednetworkssuchasIEEE802.15.4.Figure2.13showsanexample of an IoT protocol stack using the 6LoWPAN adaptation layer beside the well-knownIPprotocolstackforreference.

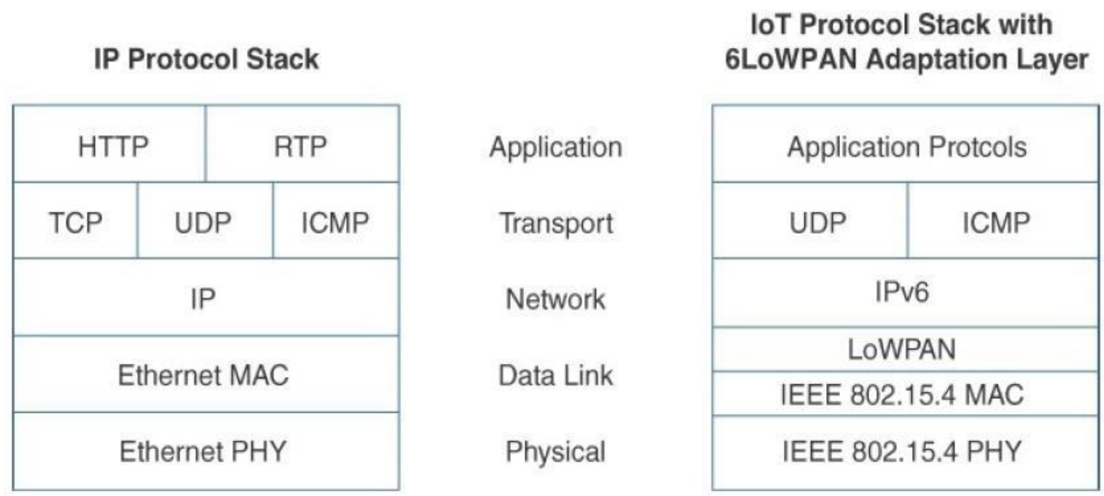

Figure2.13:ComparisonofanIoTProtocolStackUtilizing6LoWPANandanIPProtocolStack

The 6LoWPAN working group published several RFCs, but RFC 4994 is foundationalbecause it defines frame headers for the capabilities of header compression, fragmentation,and mesh addressing. These headers can be stacked in the adaptation layer to keep theseconceptsseparatewhileenforcingastructuredmethodforexpressingeachcapability .Depending on the implementation, all, none, or any combination of these capabilities andtheir corresponding headers can be enabled. Figure 2.14 shows some examples of typical6LoWPAN headerstacks.

Figure2.146LoWPANHeaderStack

HeaderCompression

IPv6headercompressionfor6LoWPANwasdefinedinitiallyinRFC4944andsubsequently updated by RFC 6282. This capability shrinks the size of IPv6's 40-byte headersand User Datagram Protocol's (UDP's) 8-byte headers down as low as 6 bytes combined insome cases. Note that header compression for 6LoWPAN is only defined for an IPv6 headerandnotIPv4.

The 6LoWPAN protocol does not support IPv4, and, in fact, there is no standardizedIPv4 adaptation layer for IEEE 802.15.4. 6LoWPAN header compression is stateless, andconceptually it is not too complicated. However, a number of factors affect the amount ofcompression,suchasimplementationofRFC4944versusRFC6922,whetherUDPisincl uded,andvarious IPv6addressingscenarios.

At a high level, 6LoWPAN works by taking advantage of shared information known by allnodes from their participation in the local network. In addition, it omits some standardheader fields by assuming commonly used values. Figure 2.15 highlights an example thatshowstheamountof reductionthatispossiblewith6LoWPANheadercompression.

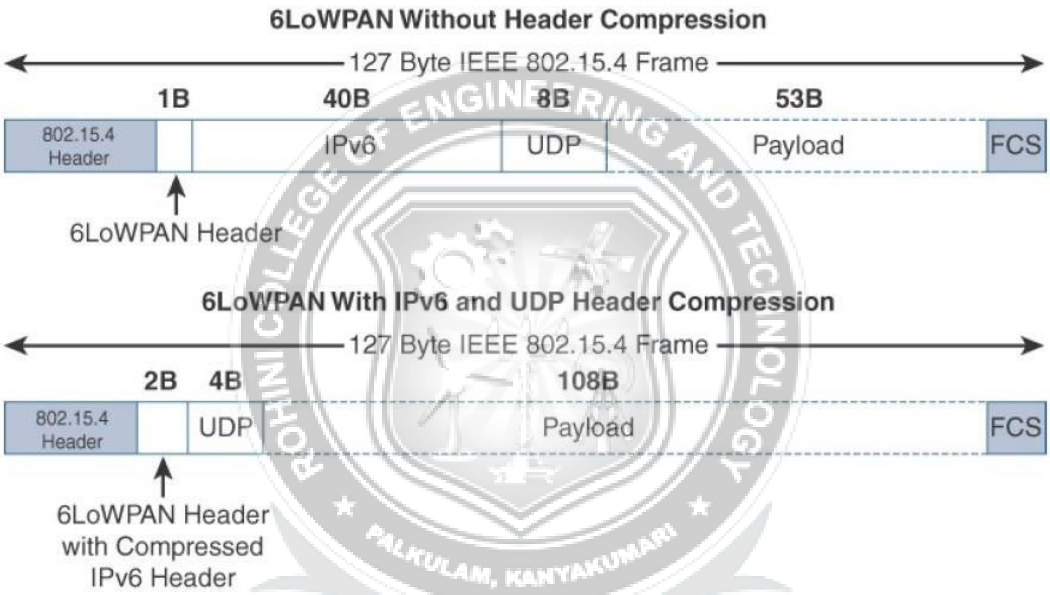

Figure2.156LoWPANHeaderCompression

AtthetopofFigure2.15,youseea6LoWPANframewithoutanyheadercompression enabled: The full 40- byte IPv6 header and 8-byte UDP header are visible. The6LoWPAN header is only a single byte in this case. Notice that uncompressed IPv6 and UDPheaders leave only 53 bytes of data payload out of the 127- byte maximum frame size in thecaseofIEEE802.15.4.

The bottom half of Figure 2.15 shows a frame where header compression has beenenabledforabest-casescenario.The6LoWPANheaderincreasesto2bytestoaccommodate the compressed IPv6 header, and UDP has been reduced in half, to 4 bytesfrom 8. Most importantly, the header compression has allowed the payload to more thandouble, from 53 bytes to 108 bytes, which is obviously much more efficient. Note that the 2-byteheadercompressionappliestointra-cellcommunications,whilecommunicationsexternaltothecellmayrequiresomefieldof theheaderto notbecompressed.

Fragmentation

The maximum transmission unit (MTU) for an IPv6 network must be at least 1280 bytes. Theterm *MTU* defines the size of the largest protocol data unit that can be passed. For IEEE802.15.4, 127 bytes is the MTU. This is a problem because IPv6, with a much larger MTU, iscarried inside the 802.15.4 frame with a much smaller one. To remedy this situation, largeIPv6packetsmustbefragmentedacrossmultiple802.15.4framesatLayer2.

The fragment header utilized by 6LoWPAN is composed of three primary fields: DatagramSize, Datagram Tag, and Datagram Offset. The 1-byte Datagram Size field specifies the totalsizeoftheunfragmentedpayload.DatagramTagidentifiesthesetoffragmentsforapayload. Finally, the Datagram Offset field delineates how far into a payload a particularfragmentoccurs.Figure2.16providesanoverviewofa6LoWPANfragmentation header.

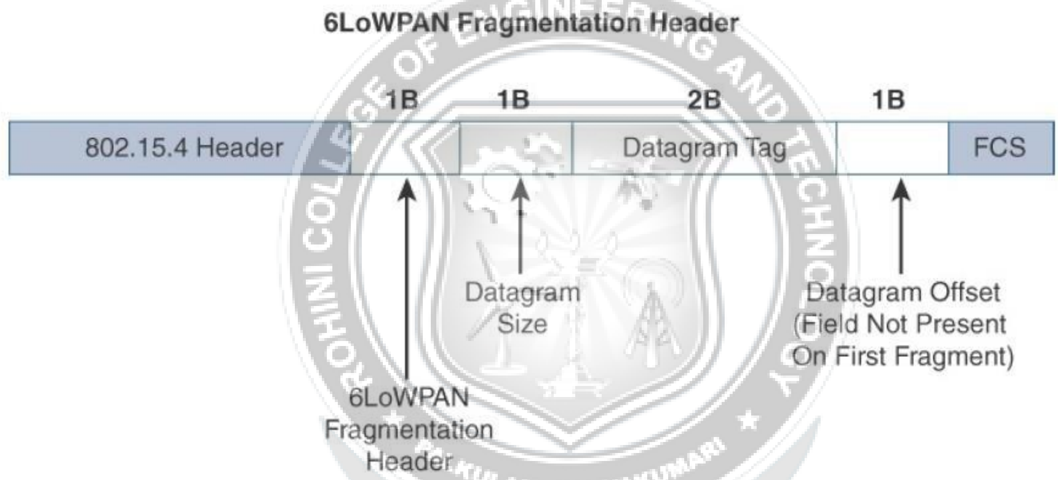

Figure2.166LoWPANFragmentationHeader

In Figure 2.16, the 6LoWPAN fragmentation header field itself uses a unique bit valueto identify that the subsequent fields behind it are fragment fields as opposed to anothercapability, such as header compression. Also, in the first fragment, the Datagram Offset fieldis not present because it would simply be set to 0. This results in the first fragmentationheader for an IPv6 payload being only 4 bytes long. The remainder of the fragments have a5-byteheaderfieldsothattheappropriateoffsetcanbespecified.

MeshAddressing

The purpose of the 6LoWPAN mesh addressing function is to forward packets overmultiple hops. Three fields are defined for this header: Hop Limit, Source Address, andDestination Address. Analogous to the IPv6 hop limit field, the hop limit for mesh addressingalso provides an upper limit on how many times the frame can be forwarded. Each hopdecrements this value by 1 as it is forwarded. Once the value hits 0, it is dropped and nolongerforwarded.

TheSourceAddressandDestinationAddressfieldsformeshaddressing areIEEE 802.15.4addressesindicatingtheendpointsofanIPhop.Figure2.17detailsthe6LoWPANmeshaddressing headerfields.

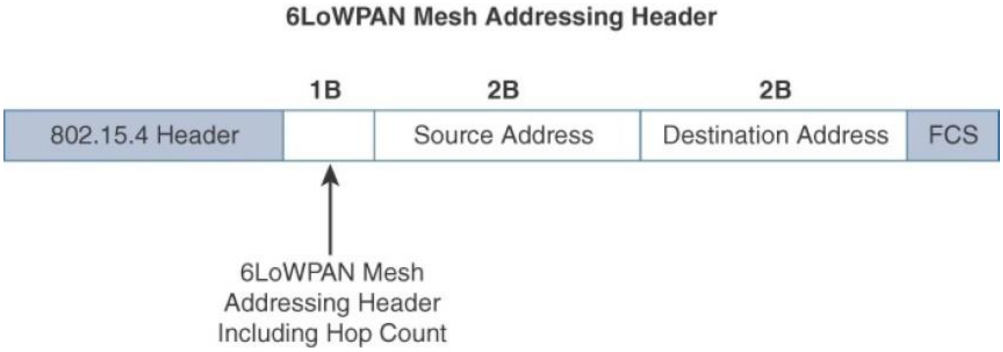

Figure2.17:6LoWPANMeshAddressingHeader

Note that the mesh addressing header is used in a single IP subnet and is a Layer 2type of routing known as mesh-under. RFC 4944 only provisions the function in this case asthe definition of Layer 2 mesh routing specifications was outside the scope of the 6LoWPANworkinggroup,andtheIETFdoesn'tdefine"Layer2routing."Animplementationperforming Layer 3 IP routing does not need to implement a mesh addressing header unlessrequiredby agiventechnology profile.