CHAPTER 5

EVOLVING IoT STANDARDS

For many years, embedded systems have been deployed as specialized vertical applications with unique functions and attributes. As the need arises for broad-scale deployment, with the ensuing requirement of being able to easily connect these embedded machines to control systems and to users that require interaction with them, standards become fundamentally important. This chapter provides a short survey of some key evolving standards that can be used to support IoT applications. Mainstream layer 1/2 communication standards (specifically, Zigbee, Bluetooth, and long-term evolution [LTE]) and layer 3 communication standards (specifically, IPv6, Mobile IPv6, and IPv6 technologies directly applied to the IoT) are discussed in Chapters 6 through 9; this chapter, therefore, covers the multitude of other support standards that come into play in the deployment of IoT and machine-to-machine (M2M) services (also known as machine-type communication [MTC] in third-generation partnership project [3GPP] environments).

5.1 OVERVIEW AND APPROACHES

Despite technological advances in many supporting technologies that are advancing IoT concepts, difficulties associated with interworking and multisupplier approaches still hamper the cost-effective implementation and rollout of the technology. When there is insufficient standardization, capability mismatches between different devices

Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications,
First Edition. Daniel Minoli.

easily arise. While IoT systems can utilize existing Internet protocols, as mentioned earlier, in a number of cases the power-, processing-, and capabilities-constrained IoT environment can benefit from additional protocols that help optimize the communications and lower the computational requirements. Developers have expressed the desire for having the IoT utilize existing Internet protocol stack, to a large extent and to the degree possible. However, one should expect some challenges and modifications because of the larger capability variations than in the current Internet, and because of the fact that there is no human in the loop for most applications (M2M), although humans may be in the loop in human-to-machine (H2M) situations. Also, as hinted in Chapter 4, power consideration drive the need for leaner protocol stacks.

Standards covering many of the underlying technologies are important because proprietary solutions fragment the industry. Standards are particularly critical when there is a requirement to physically or logically connect entities across an interface. Some areas requiring standardization include, but are not limited to, the following (1–3):

- Developing IP/routing/transport/web protocols subsets that scale down to IOT devices; specifically, lightweight routing protocols for the IoT;
- Describing architectures that employ gateways and middleware;
- Developing mobility management;
- Internetworking of IoT things;
- Lightweight implementations of cryptographic stacks; and building a suitable security infrastructure: end-to-end security capabilities for the IoT things;
- Developing standards for applications, specifically, data formats; and
- Discouraging on domain-specific solutions.

There is a practical desire, motivated by financial consideration, to build optimized solutions that can solve the problem in a particular setting, but these solutions may not be general enough for all situations. Such “point solutions” invariably leads to interoperability problems. Some observers make the case that Internet protocols were successful because they were good enough, scalable, and useful, not because they were particularly optimized for any hardware back in the early days (3).

Fortunately, several global organizations are currently working on global M2M standards. Several standardization efforts are underway addressing layer-specific protocols, optimized architectures, and policy, including but not limited to the following:

- The Internet Engineering Task Force (IETF) IPv6 routing protocol for low power and lossy networks (RPL)/routing over low power and lossy networks (ROLL);
- IETF constrained application protocol (CoAP);

- IETF constrained RESTful environments (CoRE);
- IETF IPv6 over low power WPAN (6LoWPAN);
- 3GPP MTC; and
- ETSI M2M. Recall that M2M involves communication without (or only limited) human intervention where the human is not the input agent but possibly (but not always) the output agent. For example, ETSI TS/TR 102 addresses M2M architecture and services (e.g., smart metering, e-health, auto, and city).

A number of specific considerations need to be taken when designing protocols and architectures for interconnecting smart objects to the Internet. Key concerns are scalability, power efficiency, interworking between different technologies and network domains, usability and manageability, and security and privacy (4). IoT standardization deals with physical interfaces, access connectivity (e.g., low power IEEE 802.15.4-based wireless standards such as IEC62591, 6LoWPAN, and ZigBee Smart Energy (SE) 2.0, DASH7/ISO/IEC 18000-7), networking (such as IPv6), and applications. IETF 6LoWPAN, ROLL, and CoRE aim at making IPv6 work well on constrained devices. 3GPP MTC seeks to include scalability in LTE. ETSI M2M aims at making devices communicate to service platforms and applications (5). Other activities include:

- IEEE 802: addresses LANs, WLANs, and PANs (personal area networks), particularly the IEEE802.15.4 wireless standards such as IEC62591, 6LoWPAN, and ZigBee, ZigBee IP (ZIP), ZigBee SE 2.0—IEEE 802 now includes over 100 standards. Specifically, the ZigBee Alliance's ZIP standard is a first definition of an open standards-based IPv6 stack for smart objects, the goal being to bring IPv6 network protocols over 802.15.4 wireless mesh networks to reality.
- IEEE P2030/SCC21: addresses smart grid (SG) interoperability.
- Emerging IEEE P1901.2 standard for Orthogonal Frequency Division Multiplexing (OFDM)-based communication over power lines and offers guaranteed interoperability. This standard is key to fostering SG deployments.
- ETSI TS/TR 102: addresses M2M architecture, services, smart metering, e-health, auto, and city.
- 3GPP SA1-SA3: addresses services, architecture, and security.
- JTC1 SC 6 and China NITSC: address sensor networks.
- TIA: TR-50: addresses smart device communications.
- CENELEC: addresses device addressability.

In summary, three major strands of press time standardization include the following: (i) ETSI: for end-to-end framework for M2M; (ii) 3GPP: to enable operators to support services; and (iii) IEEE: to optimize the radio access/physical layer.

5.2 IETF IPv6 ROUTING PROTOCOL FOR RPL ROLL

Low power and lossy networks (LLNs) are[1] a class of networks in which both the routers and their interconnect are constrained. LLN routers typically operate with constraints on processing power, memory, and energy (battery power); their interconnects are characterized by high loss rates, low data rates, and instability. LLNs comprise a few dozen routers up to thousands of routers. Supported traffic flows include point-to-point (between devices inside the LLN), point-to-multipoint (from a central control point to a subset of devices inside the LLN), and multipoint-to-point (from devices inside the LLN toward a central control point). The IPv6 Routing Protocol for LLNs (RPL) is a mechanism proposed by the IETF to support multipoint-to-point traffic from devices inside the LLN toward a central control point, as well as point-to-multipoint traffic from the central control point to the devices inside the LLN (6).

LLNs consist largely of constrained nodes (with limited processing power, memory, and sometimes energy when they are battery operated or energy scavenging). These routers are interconnected by lossy unstable links, resulting in relatively high packet loss rates and typically supporting only low data rates. Another characteristic of such networks is that the traffic patterns are not simply point-to-point, but in many cases point-to-multipoint or multipoint-to-point. Furthermore, such networks may potentially comprise up to thousands of nodes. These characteristics offer unique challenges to a routing solution. To address these issues, the IETF ROLL Working Group has defined application-specific routing requirements for an LLN routing protocol; it has also specified the RPL. A set of IETF companion documents to the basic specification provides further guidance in the form of applicability statements specifying a set of operating points appropriate to the building automation, home automation, industrial, and urban application scenarios.

Existing routing protocols include OSPF/IS-IS (open shortest path first/intermediate system to intermediate system), OLSRv2 (optimized link state routing protocol version 2), TBRPF (topology-based reverse path forwarding), RIP (routing information protocol), AODV (ad hoc on-demand distance vector), DYMO (dynamic MANET on-demand), and DSR (dynamic source routing). Some of the metrics to be considered for IoT applications include the following:

- Routing state memory space—limited memory resources of low power nodes;
- Loss response—what happens in response to link failures;
- Control cost—constraints on control traffic;
- Link and node cost—link and node properties are considered when choosing routes.

The existing protocols all fail one or more of these goals for IoT applications. For example, for *protocol state memory size* OSPF/IS-IS fails; for *loss* OSPF/IS-IS fails;

[1] This discussion is based on and summarized from the IETF document draft-ietf-roll-rpl-19 [6]; it included to motivate the reader to consult the full document and/or related IETF documents for an inclusive view of the issue.

for *control* OSPF/IS-IS fails; for *link cost* OSPF/IS-IS would pass; and for *node cost* OSPF/IS-IS fails (see Reference 7 for additional information). Hence, the need for a new protocol.

In order to be useful in a wide range of LLN application domains, RPL separates packet processing and forwarding from the routing optimization objective. Examples of such objectives include minimizing energy, minimizing latency, or satisfying constraints. An RPL implementation, in support of a particular LLN application, will include the necessary objective function(s) as required by the application.

Consistent with the layered architecture of IP, RPL does not rely on any particular features of a specific link layer technology. RPL is designed to be able to operate over a variety of different link layers, including ones that are constrained, potentially lossy, or typically utilized in conjunction with highly constrained host or router devices, such as but not limited to low power wireless or PLC (power line communication) technologies.

RPL operations, however, require bidirectional links. In some LLN scenarios, communication links may exhibit asymmetric properties. Therefore, the reachability of a router needs to be verified before the router can be used as a parent. RPL expects an external mechanism to be triggered during the parent selection phase in order to verify link properties and neighbor reachability. Neighbor unreachability detection (NUD) is such a mechanism, but alternates are possible, including bidirectional forwarding detection described in RFC 5881 and hints from lower layers via layer 2 triggers. In general, a detection mechanism that is reactive to traffic is favored in order to minimize the cost of monitoring links that are not being used.

RPL also expects an external mechanism to access and transport some control information, referred to as the “RPL Packet Information,” in data packets. The RPL packet information enables the association of a data packet with an RPL instance and the validation of RPL routing states. The IPv6 Hop-by-Hop RPL option is an example of such a mechanism. The mechanism is required for all packets except when strict source routing is used which, by nature, prevents endless loops and alleviates the need for the RPL packet information. Future companion specifications may propose alternate ways to carry the RPL packet information in the IPv6 packets and may extend the RPL packet information to support additional features.

RPL provides a mechanism to disseminate information over the dynamically formed network topology. The dissemination enables minimal configuration in the nodes, allowing nodes to operate mostly autonomously.

In some applications, RPL assembles topologies of routers that own independent prefixes. Those prefixes may or may not be aggregatable depending on the origin of the routers. A prefix that is owned by a router is advertised as “on-link.”

RPL also introduces the capability to bind a subnet together with a common prefix and to route within that subnet. A source can inject information about the subnet to be disseminated by RPL, and that source is authoritative for that subnet. Because many LLN links have non-transitive properties, a common prefix that RPL disseminates over the subnet must not be advertised as on-link.

RPL may, in particular, disseminate IPv6 neighbor discovery (ND) information prefix information option (PIO) and the route information option (RIO). ND

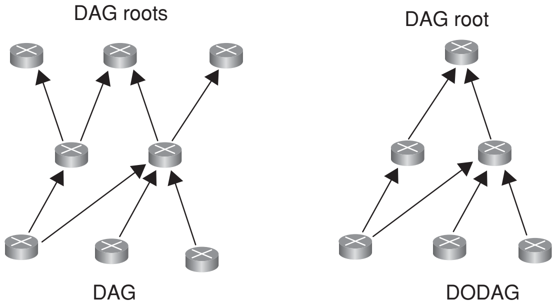


FIGURE 5.1 DAGs and DODAGs.

information that is disseminated by RPL conserves all its original semantics for router to host, with limited extensions for router to router, though it is not to be confused with routing advertisements and it is never to be directly redistributed in another routing protocol. An RPL node often combines host and router behaviors.

Some basic definitions in RPL are as follows (see Fig. 5.1):

- Directed acyclic graph (DAG) is a directed graph with no cycles.
- Destination-oriented DAG (DODAG) is a DAG rooted at a single destination.

RPL defines optimization objective when forming paths toward roots based on one or more metrics. Metrics may include both link properties (reliability, latency) and node properties (e.g., powered on not). RPL defines a new ICMPv6 message with three possible types:

- DAG information object (DIO)—carries information that allows a node to discover an RPL instance, learn its configuration parameters, and select DODAG parents;
- DAG information solicitation (DIS)—solicit a DODAG information object from an RPL node;
- Destination advertisement object (DAO)—used to propagate destination information upward along the DODAG.

A node rank defines a node's relative position within a DODAG with respect to the DODAG root.

The approach in RPL is to build a topology (instance) where routes to these nodes are optimized (namely, DODAG(s) rooted at these nodes). DODAG construction proceeds as follows (7):

- Nodes periodically send link-local multicast DIO messages;
- Stability or detection of routing inconsistencies influence the rate of DIO messages;

- Nodes listen for DIOs and use their information to join a new DODAG, or to maintain an existing DODAG;
- Nodes may use a DIS message to solicit a DIO;
- Based on information in the DIOs, the node chooses parents that minimize path cost to the DODAG root.

RPL is optimized for many-to-one and one-to-many traffic patterns. Routing state is minimized: stateless nodes have to store only instance(s) configuration parameters and a list of parent nodes. The protocol takes into account both link and node properties when choosing paths. Additionally, link failures do not trigger global network re-optimization. The reader is referred to the draft specification discussed in the key reference (6) for an extensive discussion of the capabilities, formats, and procedures of this protocol.

5.3 CONSTRAINED APPLICATION PROTOCOL (CoAP)

5.3.1 Background

The IETF constrained RESTful environments (CoRE) Working Group has recently undertaken standardization work the CoAP. CoAP is a simple application layer protocol targeted to simple electronic devices (e.g., IoT/M2M things) to allow them to communicate interactively over the Internet. CoAP is designed for low power sensors (especially wireless sensor network [WSN] nodes described in Chapters 3 and 4) and for actuators that need to be controlled or monitored remotely, using IP/Internet networks. CoAP can be seen as a specialized web transfer protocol for use with constrained networks and nodes for M2M applications, such as smart energy and building automation. CoAP operates with HTTP (hypertext transfer protocol) for basic support with the web, allowing proxies to be built providing access to CoAP resources via HTTP in a uniform way, while also supporting multicast and enjoying low overhead CoAP can run on most devices that support user datagram protocol (UDP) or a similar protocol. Some key aspects of the protocol are as follows: (i) minimal complexity for the mapping with HTTP; (ii) low header overhead and low parsing complexity; (iii) support for the discovery of resources; (iv) simple resource subscription process; and (v) simple caching based on max-age.

CoAP makes use of two message types, requests and responses, using a simple binary base header format. The base header may be followed by options in Internet control message protocol (ICMP)-style type-length-value format. CoAP is by default bound to UDP and, optionally, to transmission control protocol (TCP). Any bytes after the headers in the packet are considered the message body if any. The length of the message body is implied by the datagram length. When bound to UDP, the entire message must fit within a single datagram. When used with 6LoWPAN as defined in RFC 4944, messages fit into a single IEEE 802.15.4 frame.

The constrained nodes for which CoAP is targeted often have 8-bit microcontrollers with small amounts of ROM and RAM, while networks such as 6LoWPAN often have high packet error rates and a typical throughput of 10s of Kbps. CoAP provides a method/response interaction model between application end-points, supports built-in resource discovery, and includes key web concepts such as URIs (uniform resource identifiers) and content-types. CoAP easily translates to HTTP for integration with the web while meeting specialized requirements such as multicast support, very low overhead, and simplicity for constrained environments (8).

The use of Web Services (WS) on the Internet has become ubiquitous in most applications; it depends on the fundamental representational state transfer (REST) architecture of the web (see Section 5.4). The CoRE working group[2] aims at realizing the REST architecture in a suitable form for constrained IoT/M2M nodes (e.g., 8-bit microcontrollers with limited RAM and ROM) and IoT/M2M networks (e.g., 6LoWPAN). Constrained networks such as 6LoWPAN support the expensive fragmentation of IPv6 packets into small link-layer frames. One design goal of CoAP has been to keep message overhead small, thus limiting the use of fragmentation.

One of the main goals of CoAP is to design a generic web protocol for the special requirements of this constrained environment, especially considering energy, building automation, and other M2M applications. The objective of CoAP is not to statically compress HTTP, but rather to realize a subset of REST common with HTTP, but optimized for M2M applications. Although CoAP can be used for compressing simple HTTP interfaces, it also offers features for M2M such as built-in discovery, multicast support, and asynchronous message exchanges. CoAP has the following main features:

- Constrained web protocol fulfilling M2M requirements;
- UDP binding with optional reliability supporting unicast and multicast requests;
- Asynchronous message exchanges;
- Low header overhead and parsing complexity;
- URI and content-type support;
- Simple proxy and caching capabilities;
- A stateless HTTP mapping, allowing proxies to be built providing access to CoAP resources via HTTP in a uniform way or for HTTP simple interfaces to be realized alternatively over CoAP; and
- Security binding to datagram transport layer security (DTLS).

The interaction model of CoAP is similar to the client/server model of HTTP. However, M2M interactions typically result in a CoAP implementation acting in both client and server roles (called an end-point). A CoAP request is equivalent to that of HTTP and is sent by a client to request an action (using a method code) on a resource (identified by a URI) on a server. The server then sends a response with a response code; this response may include a resource representation. Unlike

[2] This discussion is based on and summarized from the IETF document draft-ietf-core-coap-09 (8); it is included to motivate the reader to consult the full document and/or related IETF documents for an inclusive view of the issue.

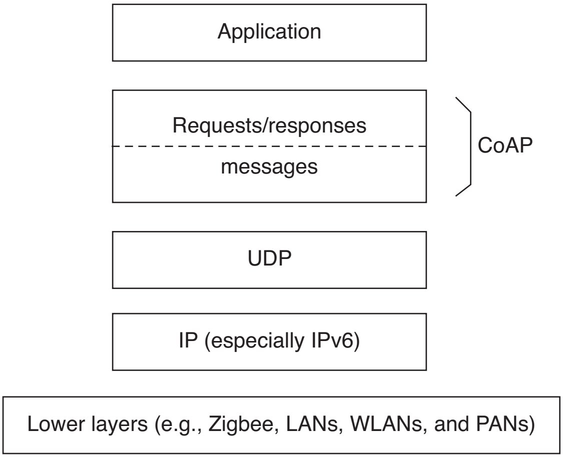


FIGURE 5.2 Abstract layering of CoAP.

HTTP, CoAP deals with these interchanges asynchronously over a datagram-oriented transport such as UDP. This is done logically using a layer of messages that supports optional reliability (with exponential back-off). CoAP defines four types of messages: confirmable (CON), non-confirmable (NON), acknowledgement, reset; method codes and response codes included in some of these messages make them carry requests or responses. The basic exchanges of the four types of messages are transparent to the request/response interactions.

One could think of CoAP logically as using a two-layer approach, a CoAP messaging layer used to deal with UDP and the asynchronous nature of the interactions, and the request/response interactions using method and response codes (see Fig. 5.2). CoAP is, however, a single protocol, with messaging and request/response just features of the CoAP header. Figure 5.3 depicts the overall protocol stack that is being considered in the CoAP context.

The reader is referred to the draft specification discussed in the key reference (8) for an extensive discussion of the capabilities, formats, and procedures of this protocol. A short summary follows.

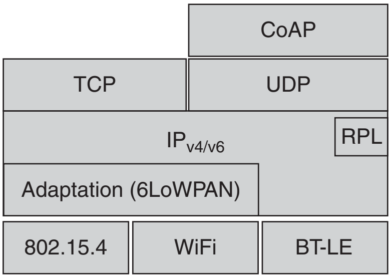


FIGURE 5.3 Overall protocol stack in CoAP’s environment.

5.3.2 Messaging Model

The CoAP messaging model is based on the exchange of messages over UDP between end-points. It uses a short fixed-length binary header (4 bytes) that may be followed by compact binary options and a payload. This message format is shared by requests and responses. Each CoAP message contains a message ID used to detect duplicates and for optional reliability.

Reliability is provided by marking a message as CON. A CON message is retransmitted using a default timeout and exponential back-off between retransmissions, until the recipient sends an acknowledgement message (ACK) with the same message ID from the corresponding end-point. When a recipient is not able to process a CON message, it replies with a reset message (RST) instead of an ACK. A message that does not require reliable delivery, for example, each single measurement out of a stream of sensor data, can be sent as a NONmessage. These are not acknowledged, but still have a message ID for duplicate detection. When a recipient is not able to process a NON message, it may reply with an RST.

Since CoAP is based on UDP, it also supports the use of multicast IP destination addresses, enabling multicast CoAP requests.

5.3.3 Request/Response Model

CoAP request and response semantics are carried in CoAP messages, which include either a method code or response code, respectively. Optional (or default) request and response information, such as the URI and payload content-type, are carried as CoAP options. A token option is used to match responses to requests independent of the underlying messages.

A request is carried in a CON or NON message, and if immediately available, the response to a request carried in a CON message is carried in the resulting ACK message. This is called a piggy-backed response. If the server is not able to respond immediately to a request carried in a CON message, it simply responds with an empty ACK message so that the client can stop retransmitting the request. When the response is ready, the server sends it in a new CON message (which then in turn needs to be acknowledged by the client). This is called a separate response. Likewise, if a request is sent in a NON message, then the response is usually sent using a new NON message, although the server may send a CON message.

CoAP makes use of GET, PUT, POST, and DELETE methods in a similar manner to HTTP.

5.3.4 Intermediaries and Caching

The protocol supports the caching of responses in order to efficiently fulfill requests. Simple caching is enabled using freshness and validity information carried with CoAP responses. A cache could be located in an end-point or an intermediary.

Proxying is useful in constrained networks for several reasons, including (i) network traffic limiting, (ii) to improve performance, (iii) to access resources of sleeping devices, or (iv) for security reasons. The proxying of requests on behalf of another CoAP end-point is supported in the protocol. The URI of the resource to request is included in the request, while the destination IP address is set to the proxy.

The reader is referred to the draft specification discussed in the key reference (8) for an extensive discussion of the capabilities, formats, and procedures of this protocol.

5.4 REPRESENTATIONAL STATE TRANSFER (REST)

As noted, CoAP uses REST techniques. REST was first described in 2000 by Roy Fielding in his University of California dissertation which analyzed a set of web-focused software architecture principles for distributed computing. REST aims at supporting scalability of component interactions, generality of interfaces, and independent deployment of components. Hence, it defines a set of architectural principles by which one can design WS that focus on a system's resources, including how resource states are addressed and transferred over HTTP by a plethora of clients written in different languages (9). Stated differently, REST is an architectural style of large-scale networked software that takes advantage of the technologies and protocols of the World Wide Web; it describes how distributed data objects, or resources, can be defined and addressed, stressing the easy exchange of information and scalability (10). A REST-based WS follows four basic design principles:

- Use HTTP methods explicitly.
- Be stateless.
- Expose directory structure-like URIs.
- Transfer XML, JavaScript Object Notation (JSON), or both.

5.5 ETSI M2M

ETSI recently created a dedicated Technical Committee, with the mission to develop standard M2M communications. The group seeks to provide an end-to-end view of M2M standardization and is expected to co-operate closely with ETSI's ongoing activities on next-generation networks (NGNs), radio communications, fiber optics and powerline, as well as collaboration with 3GPP standards group on mobile communication technologies. The reference model used in this text is the M2M model developed by this group, as defined in various evolving standards, including the ETSI M2M Release 1 standards described in ETSI TS 102 689 (requirements), ETSI TS 102 690 (functional architecture), and ETSI TS 102 921 (interface descriptions). ETSI has also published a number of documents defining common use cases. These documents were cited in other chapters and are not re-listed here.

Key elements in the M2M environment include the following (11):

- M2M device: A device capable of replying to request for data contained within those device or capable of transmitting data contained within those devices autonomously;
- M2M area network (device domain): A network that provides connectivity between M2M devices and M2M gateways, for example, a PAN;
- M2M gateway: A gateway (say a router or higher layer network element) that uses M2M capabilities to ensure M2M devices interworking and interconnection to the communication network;
- M2M communication networks (network domain): A wider-range network that supports communications between the M2M gateway(s) and M2M application; examples include but are not limited to xDSL, LTE, WiMAX, and WLAN; and
- M2M applications: Systems that contain the middleware layer where data goes through various application services and is used by the specific business-processing engines.

The reader is referred to the architecture specification cited above for an extensive discussion of the M2M environment.

5.6 THIRD-GENERATION PARTNERSHIP PROJECT SERVICE REQUIREMENTS FOR MACHINE-TYPE COMMUNICATIONS

5.6.1 Approach

Current mobile networks are optimized for human-to-human (H2H) traffic and not for M2M/MTC interactions; hence, optimizations for MTC are advantageous. For example, one needs lower costs to reflect lower MTC ARPUs (average revenue per user); also, there is a need to support triggering. Hence, 3GPP has started work on M2M specification in 2010 for interoperable solutions, particularly in the 3G/4G/LTE context. Table 5.1 provides a superset of specifications that are applicable to MTC services. Figure 5.4 depicts the service model, while Figure 5.5 depicts the architecture. In that architecture, the interfaces are as follows:

- MTCu: provides MTC devices access to the 3GPP network for the transport of user traffic;
- MTCi: the reference point for MTC server to connect the 3GPP network via 3GPP bearer service; and
- MTCsms: the reference point for MTC server to connect the 3GPP network via 3GPP SMS.

The key document *3rd Generation Partnership Project Service Requirements for Machine Type Communications—Release 10* focused on overload and congestion control, extended access barring (EAB), low priority access, APN (access point

TABLE 5.1 3GPP Specifications Related to MTC

3GPP Specifications	Specifications Associated with or Affected by MTC Work
22.011	Service accessibility
22.368	Service requirements for MTC; stage 1
23.008	Organization of subscriber data
23.012	Location management procedures
23.060	General packet radio service (GPRS); service description; stage 2
23.122	Non-access-stratum (NAS) functions related to mobile station (MS) in idle mode
23.203	Policy and charging control architecture
23.401	GPRS enhancements for evolved universal terrestrial radio access network (E-UTRAN) access
23.402	Architecture enhancements for non-3GPP accesses
23.888	System improvements for MTC
24.008	Mobile radio interface layer 3 specification; core network protocols; stage 3
24.301	NAS protocol for evolved packet system (EPS); stage 3
24.368	NAS configuration management object (MO)
25.331	Radio resource control (RRC); protocol specification
29.002	Mobile application part (MAP) specification
29.018	GPRS; serving GPRS support node (SGSN)—visitors location register (VLR); Gs interface layer 3 specification
29.060	GPRS; GPRS tunneling protocol (GTP) across the Gn and Gp interface
29.118	Mobility management entity (MME)—VLR SGs interface specification
29.274	3GPP EPS; evolved GTP for control plane (GTPv2-C); stage 3
29.275	Proxy mobile IPv6 (PMIPv6)-based mobility and tunneling protocols; stage 3
29.282	Mobile IPv6 vendor-specific option format and usage within 3GPP
31.102	Characteristics of the universal subscriber identity module (USIM) application
33.868	Security aspects of MTC
36.331	Evolved universal terrestrial radio access (E-UTRA); RRC; protocol specification
37.868	RAN improvements for MTC
43.868	GERAN improvements for MTC
44.018	Mobile radio interface layer 3 specification; RRC protocol
44.060	GPRS; MS–base station system (BSS) interface; radio link control/medium access control (RLC/MAC) protocol
45.002	Multiplexing and multiple access on the radio path

name)-based congestion control, and downlink throttling (12). For MTC communication, the following communication scenarios are identified and described in the Release 10 document:

(i) MTC devices communicating with one or more MTC server;
(ii) MTC devices communicating with each other.

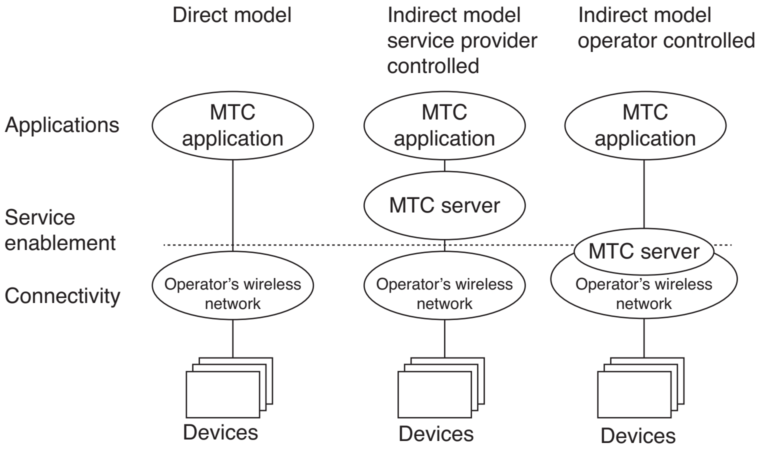
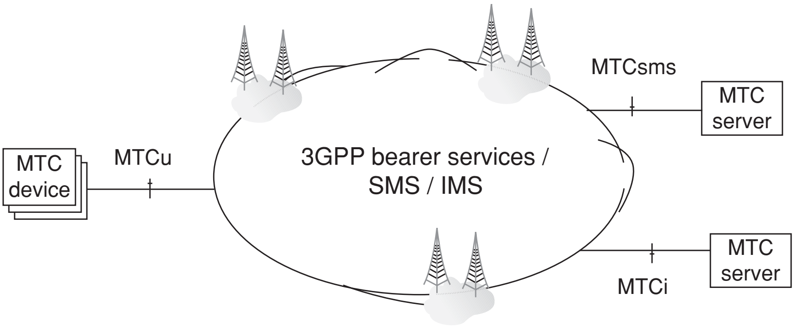

Direct model
Indirect model service provider controlled
Indirect model operator controlled
Applications
Service enablement
Connectivity
MTC application
MTC application
MTC application
MTC server
MTC server
Operator's wireless network
Operator's wireless network
Operator's wireless network
Devices
Devices
Devices

FIGURE 5.4 M2M in 3GPP—service models.



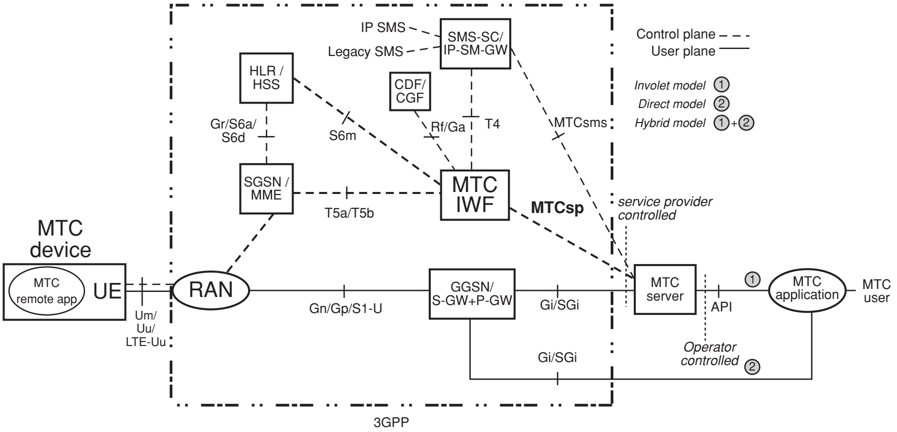


FIGURE 5.5 M2M in 3GPP—Architecture.

For MTC devices communicating with one or more MTC servers, the following use cases exist:

(a) MTC server controlled by the network operator; namely the MTC server is located in the operator domain. Here
   - The network operator offers API (e.g., Open Systems Architecture [OSA]) on its MTC server(s)
   - MTC user accesses MTC server(s) of the network operator via API

(b) MTC server not controlled by the network operator; namely MTC server is located outside the operator domain. Here
   - The network operator offers the network connectivity to the MTC server(s) located outside of the network operator domain

The communication scenario where the MTC devices communicate directly without intermediate MTC server is not considered in this release of the specification.

MTC applications do not all have the same characteristics. This implies that not every system optimization is suitable for every MTC application. Therefore, MTC features are defined in Release 10 to provide structure for the different system optimization possibilities that can be invoked. Such MTC features are offered on a per subscription basis. MTC features can be individually activated. The following MTC features have been defined:

- Low mobility
- Time controlled
- Time tolerant
- Packet switched (PS) only (here the MTC feature PS only is intended for use with MTC devices that only require packet switched services)
- Small data transmissions
- Mobile originated only
- Infrequent mobile terminated
- MTC monitoring
- Priority alarm
- Secure connection
- Location-specific trigger
- Network provided destination for uplink data
- Infrequent transmission

5.6.2 Architectural Reference Model for MTC

The latest Release 11 (an extensive document) focuses on numbers and addressing, on improvements of device triggering, and on interfaces between MTC server and mobile network (13, 14). Referring to Figure 5.5, MTCsp is a new control interface for interactions with MTC server; MTC-IWF is a new interworking function between (external) MTC server and operator core network handling security, authorization, authentication, and charging.

The end-to-end application, between the user equipment (UE) used for MTC and the MTC application, uses services provided by the 3GPP system, and optionally services provided by an MTC server. The 3GPP system provides transport and communication services (including 3GPP bearer services, IMS, and SMS) including various optimizations that can facilitate MTC. Figure 5.5 shows UE used for MTC connecting to the 3GPP network (UTRAN, E-UTRAN, GERAN, I-WLAN, and so on) via the Um/Uu/LTE-Uu interface. The architecture encompasses a number of models as follows:

- Direct model—direct communication provided by the 3GPP operator: The MTC application connects directly to the operator network without the use of any MTC server;
- Indirect model—MTC service provider controlled communication: The MTC server is an entity outside of the operator domain. The MTCsp and MTCsms are external interfaces (i.e., to a third-party M2M service provider);
- Indirect model—3GPP operator controlled communication: The MTC server is an entity inside the operator domain. The MTCsp and MTCsms are internal to the public land mobile network (PLMN);
- Hybrid model: The direct and indirect models are used simultaneously in the hybrid model, for example, connecting the user plane using the direct model and doing control plane signalling using the indirect model.

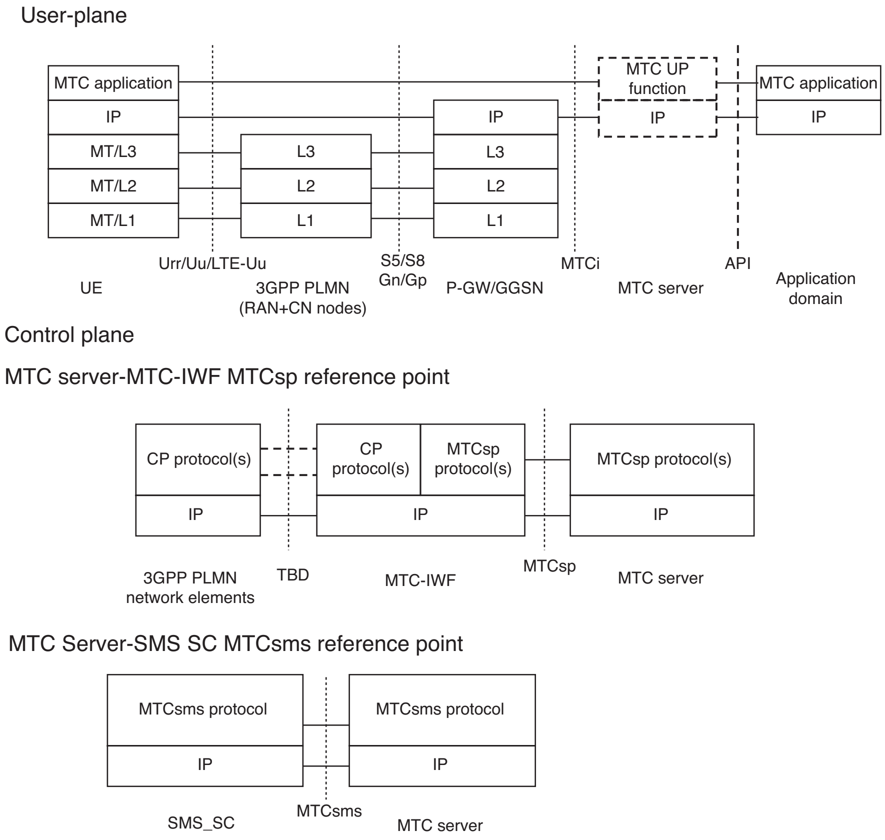
Some believe that there may be E.164 telephone number issues as related to M2M: in several countries, regulators have indicated that there are not enough (mobile) numbers available for M2M applications. 3GPP postulates that solutions will have to support 100× more M2M devices than devices for H2H communications. Proposed solutions include: (i) mid-term solution: special M2M number ranges with longer telephone numbers (e.g., 14 digits); (ii) long-term solution: no longer provide E.164 telephone numbers for M2M applications.

Figure 5.6 provides a view to the various protocol stacks defined in Release 11.

The reader is referred to the Technical Report by 3GPP (13) for an extensive discussion of architectural aspects and system requirements for MTC/M2M communication. The second part of Chapter 6 discusses some 3GPP networks that may come into play in the MTC/M2M context.

5.7 CENELEC

Recently, the European Committee for Electrotechnical Standardization (CENELEC) has accepted the transport profile of Siemens' distribution line carrier communication protocol, CX1, as a standardization proposal. The standard aims at supporting open



User-plane

Control plane

MTC server-MTC-IWF MTCsp reference point

MTC Server-SMS SC MTCsms reference point

FIGURE 5.6 User and control plane stack for MTC architecture—as described in Release 11—Technical Report 3GPP TR 23.888 V1.7.0 (2012–08).

and fault tolerant communication via powerline in intelligent power supply grids. As the basis for the transmission protocol, which uses the low voltage network as a communication channel for data of grid sensors and smart meters, the transport profile has been designed to ensure interoperability in accordance with EU Mandate M/441. CENELEC TC 13 was planning to forward the CX1 transport profile to TC 57 of the International Electrotechnical Commission (IEC) as a proposal for inclusion in the IEC standardization process. CX1 is already used to connect meters and other intelligent terminal devices in Siemens' SG metering systems, such as in the load switching devices that will replace household ripple control receivers. The systems collect energy consumption data and network information, which are then relayed to a control center for further processing (15). CX1 utilizes spread spectrum modulation, in which multiple frequencies within the same frequency band are used simultaneously to transmit a single signal. This means that interference, which often occurs at certain frequencies, has only a negligible effect on signal transmission. In addition, the communication protocol can handle any change in the physical communication parameters of a low voltage power supply grid, such as signal attenuation, noise, network disruption and signal coupling, as well as operational changes in network configuration. The protocol can also be integrated into existing IEC protocol-based network automation and energy management infrastructures.

5.8 IETF IPv6 OVER LOWPOWER WPAN (6LoWPAN)

6LoWPAN is an IPv6 adaption layer for low power wireless PAN (LoWPAN).

IPv6-over-IEEE 802.15.4 described in RFC 4944 specifies how IPv6 is carried over an IEEE 802.15.4 network with the help of an adaptation layer which sits between the MAC layer and the IP network layer. As it should be clear at this juncture, a link in a LoWPAN is characterized as lossy, low power, low bit-rate, short range, with many nodes saving energy with long sleep periods.

It turns out that multicast as used in IPv6 ND described in RFC 4861 is not desirable in such a wireless low power and lossy network. Moreover, LoWPAN links are asymmetric and non-transitive in nature. A LoWPAN is potentially composed of a large number of overlapping radio ranges. Although a given radio range has broadcast capabilities, the aggregation of these is a complex non-broadcast multi-access (NBMA) structure with generally no LoWPAN-wide multicast capabilities. Link-local scope is in reality defined by reachability and radio strength. Thus, one can consider a LoWPAN to be made up of links with undetermined connectivity properties, along with the corresponding address model assumptions defined therein. Hence, there is work underway to develop optimizations to IPv6 ND (RFC 4861) specifically aimed at low power and lossy networks such as LoWPANs (16).

This topic is covered in Chapter 9, after the reader has acquired some background on IPv6.

5.9 ZigBee IP (ZIP)

ZigBee is a wireless PAN IEEE 802.15.4 standard, which we cover in Chapter 6. Here we simply make some passing reference to the ZigBee Alliance's ZIP standard, which is a first definition of an open standards-based IPv6 stack for smart objects. The goal is to extend the use of IP networking into resource-constrained devices over a wide range of low power link technologies. The effort related to ZIP development has made significant progress to bring IPv6 network protocols over 802.15.4 wireless mesh networks to reality. ZIP is a protocol stack based on IETF- and IEEE-defined standards such as 6LoWPAN and IEEE 802.15.4 to be used for the Smart Energy 2.0 (SE 2.0) profile.

ZIP enables low power 802.15.4 nodes to participate natively with other IPv6-enabled WiFi, Homeplug, and Ethernet nodes without the complexity and cost of application layer gateways. To accomplish this, the ZIP stack incorporates a number of standardized IETF protocols including 6LoWPAN for IP header compression and ND, and RPL for mesh routing. ZIP further employs other IETF standards to support network joining procedures, service discovery, and TLS/SSL-based security mechanisms (17). At press time, the ZIP specification was nearing release of its 0.9 draft and had already progressed through numerous certification events. In particular, there has been interest in validating that ZIP will comfortably support SEP2 unicast and multicast messaging over an 802.15.4-based HAN mesh. It was anticipated that production ready, certified stacks would be available mid-2013. Early implementers included Cisco, Exegin, and Grid2Home, among others. Proponents expect that ZIP-based product offerings would soon be interoperating within the SG.

5.10 IP IN SMART OBJECTS (IPSO)

The IPSO Alliance is an advocate for IP-networked devices for use in energy, consumer, healthcare, and industrial applications. The objective of the Alliance is not to define technologies or standards, but to document the use of IP-based technologies defined at the standard organizations such as IETF with focus on support by the Alliance of various use cases. The IPSO Alliance is a non-profit association of more than 60 members at press time from leading technology, communications, and energy companies around the world. The mission is to provide a foundation for industry growth through building stronger relationships, fostering awareness, providing education, promoting the industry, generating research, and creating a better understanding of IP and its role in connecting smart objects. Goals include (18):

- Promote IP as the premier solution for access and communication for smart objects.
- Promote the use of IP in smart objects by developing and publishing white papers and case studies and providing updates on standards progress from associations like IETF, among others, and through other supporting marketing activities.
- Understand the industries and markets where smart objects can have an effective role in growth when connected using the Internet protocol.
- Organize interoperability tests that will allow members and interested parties to show that products and services using IP for smart objects can work together and meet industry standards for communication.
- Support IETF and other standards development organizations in the development of standards for IP for smart objects.

APPENDIX 5.A: LEGACY SUPERVISORY CONTROL AND DATA ACQUISITION (SCADA) SYSTEMS

This appendix provides a short summary of SCADA, a legacy, but widely deployed system used to monitor and control a plant or equipment in industries such as but not limited to energy, oil and gas refining, water and waste control, transportation, and telecommunications. This section is summarized and synthesized from reference (19) from the National Communications System (NCS). M2M approaches seek to enhance, modernize, and extend the basic concepts found in SCADA (M2M is not intended to be directly interoperable with SCADA but can be supported with proxies/gateways.)

A SCADA system gathers remote operational information, transfers the information to a central site, then alerts a management station that an event has occurred, carrying out necessary analysis and control. These systems can be relatively simple, such as one that monitors environmental conditions of a small office building, or very complex, such as a system that monitors all the activity in a nuclear power plant or the activity of a municipal water system. Traditionally, SCADA systems have made use of public switched network (PSN) facilities for monitoring purposes; wireless technologies are now being widely deployed for purposes of monitoring.

A SCADA system encompasses the transfer of data between a SCADA central host computer and a number of remote terminal units (RTUs) and/or programmable logic controllers (PLCs); the central host typically supports operator terminals. Specifically, a SCADA system consists of:

- One or more field data interface devices, usually RTUs or PLCs, which interface to field sensing devices and local control switchboxes and actuators;
- A communications system used to transfer data between field data interface devices and control units and the computers in the SCADA central host; the communication may use telephone, cable, radio, cellular, satellite, etc. or any combination of these;
- A central host computer server or servers (sometimes called a SCADA center, master station, or master terminal unit [MTU]);
- A collection of standard and/or custom software systems [sometimes called human machine interface (HMI) software or man machine interface (MMI) software] used to provide the SCADA central host and operator terminal application, support the communications system, and monitor and control remotely located field data interface devices.

There have been three generations of SCADA systems:

- First generation—monolithic approach
- Second generation—distributed approach
- Third generation—networked approach

In a SCADA system, the RTU accepts commands to operate control points, sets analog output levels, and responds to requests. The RTU provides status, as well as discrete and accumulated data to the SCADA master station. The data representations sent are not identified in any fashion other than by unique addressing. The addressing is designed to correlate with the SCADA master station database. The RTU has no knowledge of which unique parameters it is monitoring in the real world; it simply monitors certain points and stores the information in a local addressing scheme. Each protocol consists of two message sets or pairs. One set forms the master protocol, containing the valid statements for master station initiation or response, and the other set is the RTU protocol, containing the valid statements an RTU can initiate and respond to. In most but not all cases, these pairs can be considered a poll or request for information or action and a confirming response. The SCADA protocol between master and RTU forms a viable model for RTU-to-intelligent electronic device (IED) communications. Currently, there are several different protocols in use; the most common are:

- IEC 60870-5 series, specifically IEC 60870-5-101 (commonly referred to as 101) and
- Distributed network protocol version 3 (DNP3).

IEC 60870-5 Series

IEC 60870-5 specifies a number of frame formats and services that may be provided at different layers. IEC 60870-5 is based on a three-layer enhanced performance architecture (EPA) reference model (see Fig. 5A.1) for efficient implementation within RTUs, meters, relays, and other IEDs. Additionally, IEC 60870-5 defines basic application functionality for a user layer; such user layer is situated between the open system interconnection (OSI) application layer and the application program. This user layer adds interoperability for such functions as clock synchronization and file transfers.

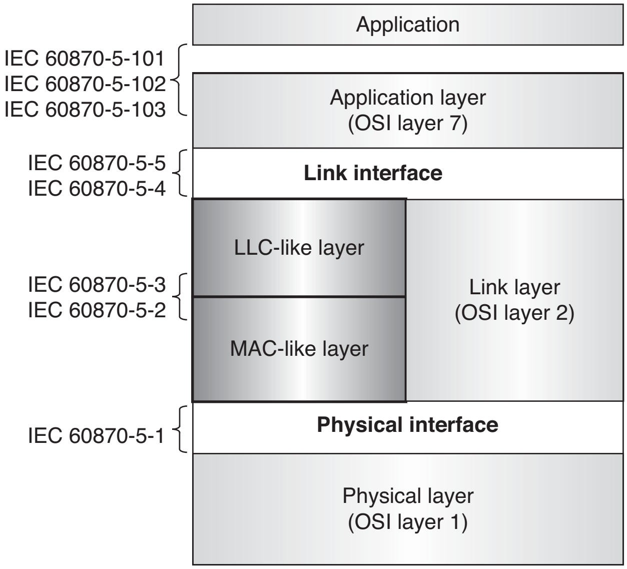


FIGURE 5A.1 SCADA protocols and EPA.

The following descriptions provide the basic scope of each of the five documents in the base IEC 60870-5 telecontrol transmission protocol specification set. Standard profiles are necessary for uniform application of the IEC 60870-5 standards. A profile is a set of parameters defining the way a device acts; such profiles have been created.

- IEC 60870-5-1 (1990–02) specifies the basic requirements for services to be provided by the data link and physical layers for telecontrol applications. In particular, it specifies standards on coding, formatting, and synchronizing data frames of variable and fixed lengths that meet specified data integrity requirements. At the physical layer, the Standard 101 Profile additionally allows the selection of International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) standards that are compatible with Electronic Industries Association (EIA) standards RS-232 and RS-485 and also support fiber optics interfaces.
- IEC-60870-5-2 (1992–04) provides a selection of link transmission procedures using a control field and optional address field; the address field is optional because some point-to-point topologies do not require either source or destination addressing.
- IEC 60870-5-3 (1992–09) specifies rules for structuring application data units in transmission frames of telecontrol systems. These rules are presented as generic standards that may be used to support a variety of present and future telecontrol applications. This section of IEC 60870-5 describes the general structure of application data and basic rules to specify application data units without specifying details about information fields and their contents.
- IEC 60870-5-4 (1993–08) provides rules for defining information data elements and a common set of information elements, particularly digital and analog process variables that are frequently used in telecontrol applications.
- IEC 60870-5-5 (1995–06) defines basic application functions that perform standard procedures for telecontrol systems, which are procedures that reside beyond layer 7 (application layer) of the ISO reference model. These utilize standard services of the application layer. The specifications in IEC 60870-5-5 (1995–06) serve as basic standards for application profiles that are then created in detail for specific telecontrol tasks.

DNP3

DNP3 is a protocol for transmission of point-to-point data using serial communications. It has been used primarily by utilities, but can also be used in other areas. The DNP3 is specifically developed for interdevice communication involving SCADA RTUs and provides for both RTU-to-IED and master-to-RTU/IED. It is based on the three-layer EPA model contained in the IEC 60870-5 standards, with some alterations to meet additional requirements of a variety of users in the electric utility industry. DNP3 was developed with the following goals in mind:

- High data integrity. The DNP3 data link layer uses a variation of the IEC 60870-5-1 (1990–02) frame format FT3. Both data link layer frames and application layer messages may be transmitted using confirmed service.
- Flexible structure. The DNP3 application layer is object based, with a structure that allows a range of implementations while retaining interoperability.
- Multiple applications. DNP3 can be used in several modes, including: (i) polled only; (ii) polled report-by-exception; (iii) unsolicited report-by-exception (quiescent mode); and (iv) a mixture of modes. It can also be used with several physical layers, and as a layered protocol it is suitable for operation over local and some wide area networks.
- Minimized overhead. DNP3 was designed for existing wire-pair data links with operating bit rates as low as 1200 bps and attempts to use a minimum of overhead while retaining flexibility. Selection of a data reporting method, such as report-by-exception, further reduces overhead.
- Open standard. DNP3 is a non-proprietary, evolving standard controlled by a users group whose members include RTU, IED, and master station vendors, and representatives of the electric utility and system consulting community.

REFERENCES

1. Gluhak A, Krco S, et al. A survey on facilities for experimental internet of things research. Communications Magazine, IEEE, November 2011;49(11):58–67.
2. Ladid L. Keynote Speech, International Workshop on Extending Seamlessly to the Internet of Things (ESLOT 2012), in conjunction with IMIS-2012 International Conference, July 4–6, 2012, Palermo, Italy.
3. Arkko J. Interoperability Challenges in the Internet of Things. Interconnecting Smart Objects with the Internet Workshop 2011, 25th March 2011, Prague.
4. Internet Architecture Board, Interconnecting Smart Objects with the Internet Workshop 2011, 25th March 2011, Prague.
5. Kutscher D, Farrell S. Towards an Information-Centric Internet with more Things. Interconnecting Smart Objects with the Internet Workshop 2011, 25th March 2011, Prague.
6. Winter T, editor. ROLL/RPL: IPv6 Routing Protocol for Low Power and Lossy Networks, March 2011, draft-ietf-roll-rpl-19.
7. Kuryla S. RPL: IPv6 Routing Protocol for Low Power and Lossy Networks, Networks and Distributed Systems Seminar, March 1, 2010.
8. Shelby Z, Hartke K, Bormann C, Frank B. Constrained Application Protocol (CoAP). CoRE Working Group, March 12, 2012, Internet-Draft, draft-ietf-core-coap-09.
9. Richardson L, Ruby S. RESTful Web Service, O’Reilly Media, 2007, Sebastopol, CA.
10. Kay R. QuickStudy: Representational State Transfer (REST). ComputerWorld, August 6, 2007.

11. Lin T-M. M2M: Machine to Machine Communication (From ETSI/3GPP Aspect). White Paper, Industrial Technology Research Institute of Taiwan, R.O.C, 2010.
12. 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects; Service Requirements for Machine Type Communications (MTC); Stage 1 (Release 10); Technical Specification 3GPP TS 22.368 V10.1.0 (2010–06).
13. 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects; System Improvements for Machine-Type Communications; Stage 1 (Release 11); Technical Report 3GPP TR 23.888 V1.7.0 (2012–08).
14. Norp T. Mobile Network Improvements for M2M, a 3GPP Perspective. ETSI M2M Workshop, October 2011. TNO, P.O. Box 342, NL-7300 AH Apeldoorn.
15. Mrosik J. International PLC data communication standard for grid automation and smart metering proposed by Siemens. On line Magazine, Nov 15, 2012, http://www. metering .com.
16. Shelby Z, editor. Neighbor Discovery Optimization for Low Power and Lossy Networks (6LoWPAN), Updates: 4944 (if approved), August 24, 2012, IETF draft-ietf-6lowpan-nd-21.
17. Duffy P. Zigbee IP: Extending the Smart Grid to Consumers., Cisco Blog – The Platform, June 4, 2012, Cisco Systems, Inc., 170 West Tasman Dr., San Jose, CA 95134 USA.
18. IPSO Alliance, http://www.ipso-alliance.org/.
19. National Communications System, Supervisory Control and Data Acquisition (SCADA) Systems, Technical Information Bulletin 04-1, NCS TIB 04-1, October 2004, P.O. Box 4052, Arlington, VA 22204-4052. http://www.ncs.gov.