CHAPTER 4

Building a Web of Things

The massive number of Internet of Things (IoT) end devices described in the preceding chapter will be producing and consuming prodigious amounts of simple data in the form of terse chirps (see Chapter 2). But if these chirps, as described, lack the traditional overhead trappings of well-known protocols such as TCP/IP, how can they be moved across the traditional Internet—or indeed, *across any network at all*?

The majority of IoT end devices will, by design, be cheap, limited in power and memory, and rudimentary. They will not be capable of managing and controlling their own networking as IP devices are expected to do. This networking task will fall to the class of devices called *propagator nodes*. These nodes are technologically a bit more like traditional networking equipment such as switches and routers, but they operate in a more broadly purposed way. IoT chirp-based traffic will be bundled, pruned, converted, and forwarded as necessary to move it throughout the network via a variety of protocols and interfaces. Propagator nodes must include a chirp packet translation into an IP packet because some packets are intended for external consumption. And, of course, the same process will happen in reverse for IP traffic destined for chirp–based end devices.

Most importantly, it will be possible for the function of some classes of propagator nodes to be influenced by agents residing within the integrator functions described in Chapter 5. Biasing the networking activity of the propagator nodes will serve to create software–defined publish/subscribe relationships across the IoT. These logical relationships won't be based on physical network topologies, but on neighborhoods of interest and affinities.

As an example, the affinity group for portable diesel generators operated by a global enterprise may be international. The resulting rollup of chirps through propagator nodes to "small data" to big data creates the power of the emerging Internet of Things (see Figure 4-1).

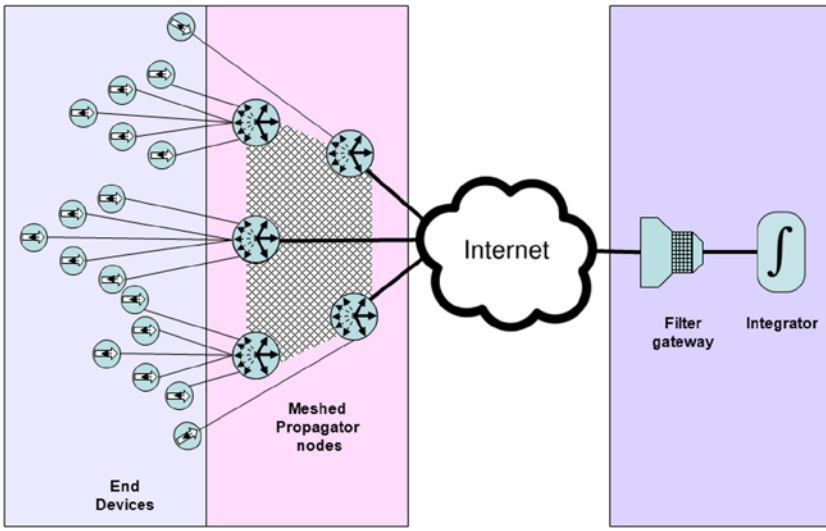

Figure 4-1. *Propagator nodes create the web of the Internet of Things, connecting end devices, other propagator nodes, and integrator functions (some with filter gateways)*

Versatility in Function and Form

Propagator nodes incorporate a number of novel concepts. Their most basic function will be to transport traffic on behalf of a huge range of end devices and other elements, including translation and gateway services to place aggregated chirp messages onto traditional networks such as TCP/IP. They will be expected to build an independent web of interconnection immediately upon power-up, discovering and connecting to nearby end devices, adjacent propagator nodes, and other elements.

In many ways, propagator nodes are the *key building blocks* of the emerging Internet of Things because their functionality allows the vast majority of end devices to remain simple and cheap, even disposable. With propagator nodes in place, IoT networks may be scaled to huge sizes, with each individual propagator node supporting many low-cost chirp–based end devices. Propagator nodes also provide connections to (and through) the traditional Internet to integrator functions in which meaning may be extracted from the "small data" of the end devices.

Architecting Trees and Leaves

The basic principles of the propagator nodes are drawn again from natural phenomena. If the billions of end devices are viewed as the "leaves" of the IoT, the propagator nodes may be seen as the "limbs" and "trunks" that connect them.

The typical tree in nature (see Figure 4-2) is structured: individual leaves do not connect to one another directly because *they have nothing of value for one another*. Instead, the branches, limbs, and trunks of the tree serve to bring water and nutrients to

the leaves and carry manufactured food from leaves to roots. From tiny shrubs to mighty redwoods, trees scale because they are structured based on this basic flow: the input and output of untold millions of end points is organized for maximum efficiency.

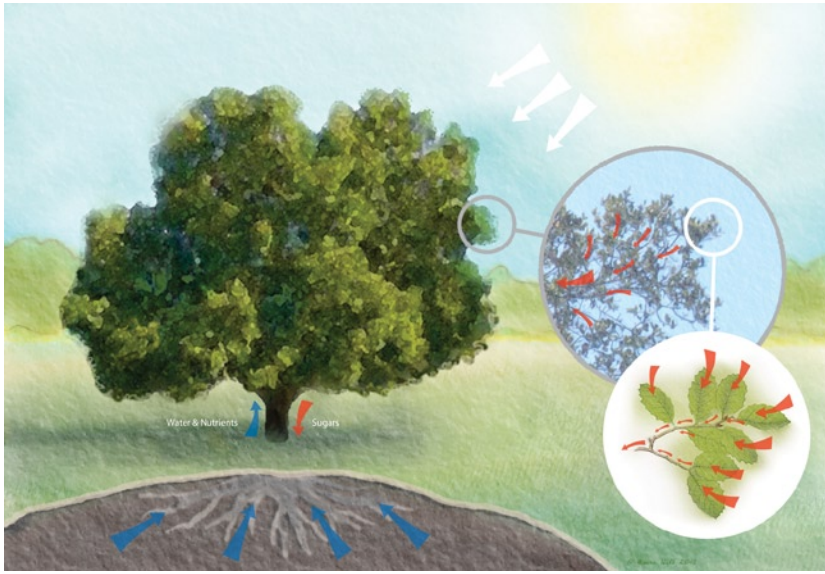

Figure 4-2. *Trees are inherently structured: no leaf connects directly to another. Instead, flows are organized through trunks, limbs, and branches. Internet of Things traffic will be similarly structured to present data only where it has meaning. Nature's arteries—rivers, trees, and so on—generally tend to be self-forming, dynamic, recursive tree-like topologies. Obviously, there is no computing resource available in nature to calculate a graph-based organization. Routing is simpler within this organic structure of recursive branching. (See the "Why Trees Scale" sidebar)*

SIDEBAR: WHY TREES SCALE

A structured tree-like network (which can be referred to as "Order *n*" [O(*n*)]), is linear with tree depth and therefore scales even in very large sizes. It scales because Moore's Law, which is also linear, is useful only when applied to counteract degradation effects, is also O(*n*). Anything more, for example, O(*n*-squared) systems, simply cannot scale with linear efficiency improvements and compensations. In nature, such systems would eventually reach extinction because of natural selection, which through trial and error over generations, would prune inefficient transport of nutrients from the root to leaves and vice versa (up and down the tree's network). Hence, O(*n*) systems prevail in nature, in large part because they are inherently efficient—and thus scalable.

> The flow of data in an IoT context is also inherently hierarchical and tree-like. At the root, there is the tree trunk and its more-focused flow. The tree trunk then again branches out into roots and tendrils.
>
> At the other end, branches assimilate small data emanating from the "leaves" (end devices). The entire process of how small data (from a myriad of end devices) is assimilated, pruned, modified, and then forwarded is hierarchical in the IoT, just as in trees.
>
> One can imagine the two tree-like structures (roots and branches) coalescing into one central location: the trunk. This is where the big data services, such as integrator functions, reside.

No doubt other types of nutrient transport technologies exist in nature in smaller plants, but none has demonstrated the majestic scaling seen in trees. For the same reasons, tree-like structured networks will prevail at the edges of the IoT. Unlike a natural "tree," human networks, with their unlimited peer-to-peer interactions, create the need for constant computing and updating for additions and perturbations. This is the major driver for using networking protocols (such as TCP/IP) with traditional Internet end points such as smartphones and PCs.

On Behalf of Chirps at the Edge

But the majority of end devices in the Internet of Things will communicate via the lightweight chirp protocol, as described in Chapter 3. chirp protocol includes only minimal addressing and error detection (Chapter 2). Therefore, global naming, full TCP/IP formatting, and protocol services must be applied elsewhere if the data is to pass to-and-from end devices.

Propagator nodes provide these services on behalf of end devices, so at least one propagator node must be in the communications path of any chirp end device so that the simple chirp transmissions may be transformed in order to be carried over the Internet and then interpreted by an integrator function (each of which are based on traditional TCP/IP). Propagator nodes perform this important function of grooming traffic for transmission to-and-from the traditional Internet and other TCP/IP networks.

Because they link the end-device chirp world and the broader IP-based network, propagator nodes will often be equipped with multiple wired and/or wireless interfaces. The "range" of a propagator node depends on the type of connections it has to its end devices. In a home network example, a propagator node might have two wireless interfaces (one to communicate with chirp–based end devices using infrared LEDS and a second IP-based connection such as Wi-Fi) to communicate with a standard 802.11 access point.

Isolating and Securing the Edge

As described here, the propagator nodes' prime function is linking the chirp–based end-devices to one or more integrator functions. These integrator functions are reached via the IP network, with the propagator nodes performing the bridging between the chirp subnetwork and its IP parent network. Without accredited propagator nodes as

the "middle man," chirp devices are unreachable from the IP side of the network. This is intentional: chirp devices become inherently secure if they are invisible in the IP addressable space of devices. Propagators are thus essential to providing the final level of control of mission-critical remote systems.

Autonomy and Coordination

With no practical way for the Internet of Things to be engineered in an overall top-down way (maximally efficient) nor to be effectively over-provisioned to the edge (minimally efficient), propagator nodes must be designed to independently develop *reasonably* efficient network architectures. This will require a balance between autonomy and cooperation that may be provided only by the use of robot-like intelligence distributed in each propagator node.

We'll examine first the general techniques used by all propagator nodes in creating the IoT architecture and then later explore different classes and modes of operation in their specific applications.

Upon power-up, each propagator node will assess its surroundings for possible connections to other IoT devices, including the type, characteristics, and functionality of adjacent communicating elements. These connections might be end devices, other propagator nodes, or integrator functions (and their associated filter gateways, as discussed in Chapter 5). Depending on the specific interfaces available (discussed below), these connections can be wired or wireless, via the traditional Internet, or even internal to the propagator node depending on packaging choices (see below). This same start-up procedure is repeated if a primary networking link is lost.

In some ways, this is similar to the discovery mechanisms used by other networking devices such as switches and routers. But unlike most IP-based devices, the overall structure of the network is "bottom-up," with each individual node having the power to create a structured (loop-free) path through negotiation with its peers. This is distinct from many traditional networks, which are engineered top-down using the capabilities designed into IP for the task (and shouldering the IP overhead to do so).

Structuring a Networking Path

The propagator node begins its startup by looking for a path to one or more integrator functions. Occasionally, this will be a direct wired or wireless connection (perhaps through the integrator functions' associated filter gateway). But in the majority of cases, there will be no directly connected integrator function. (Integrator functions are always connected via IP, so at least one propagator node in the path must be equipped to convert traffic from chirp to IP and back, as described below.)

Where there is no locally connected integrator function, the propagator node will exchange information with any other propagator node that is connected, wirelessly or wired. Each propagator node will build its own table of adjacencies, a logical network tree, so this information may be shared to permit the independent intelligence in each propagator node to determine a reasonably efficient path to one or more integrator functions.

Routes are weighted based on the number of "hops" (node-to-node connections) required to reach the integrator functions and may also consider adjacent propagator node loading and bandwidth available. Trade-offs are made between taking a more reliable but circuitous (more hops) route to the destination integrator function versus a more direct, but more loaded, connection. Similar to cars in rush-hour traffic, freeways are less efficient when crowded than a more circuitous city street route.

As seen in Figure 4-3, some propagator nodes may discover direct paths to the Internet (top node), which will *usually* provide the best path to one or more integrator nodes using an IP connection. But many propagator nodes will not have a direct path to the Internet and will instead connect via adjacent propagator nodes using either chirp or IP protocols.

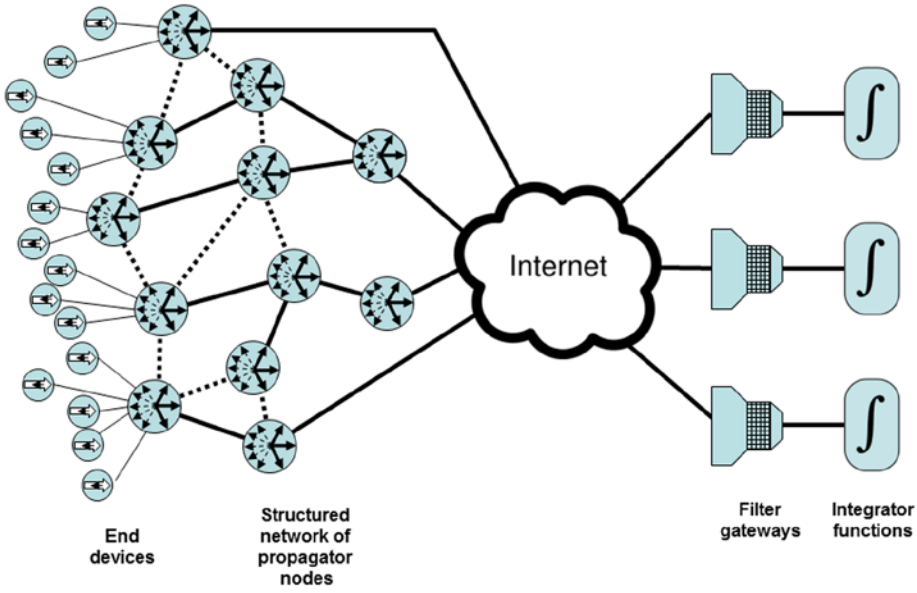

Figure 4-3. *When functioning generically, individual propagator nodes consider path information shared by adjacent nodes in building a reasonably efficient path to one or more integrator functions*

Many alternate paths may also be discovered (dotted lines); each individual propagator node will choose only one primary connection based on the information on speed, congestion, number of hops (node-to-node connections), past reliability, and so on provided by adjacent propagator nodes via housekeeping frames (see below). Alternate paths are kept in reserve in case of path or intermediate node failure, or significant speed/quality changes.

The establishment of a path to one or more integrator functions defines the "arrow" of transmission introduced in Chapter 2 and described more fully in Chapter 6. The path definition allows the propagator node to make the basic routing decisions for traffic destined for end devices versus integrator functions. This tree-based calculation maps to both the physical and the logical subnetwork of chirp devices.

The "arrow" may be loosely thought of as an overall *inherent* direction similar to upstream/uphill or downstream/downhill flows in nature. The task of the propagator nodes is to organize themselves to provide efficient flows in both directions, possibly through alternative paths.

The basic routing algorithm will also be weighted to make some specific decisions, such as preferring a wired connection to wireless if other factors are equal, but will also take note of path quality information from adjacent nodes (reliability over time, and so on).

Path decisions are revisited periodically to encompass perturbations and failures, the addition of new network elements, and updated path quality information. With each propagator node and transmission path added to the network, the immediately adjacent propagator nodes will reexamine their path analysis in order to maintain reasonably efficient paths to one or more integrator functions. Propagator nodes also perform a fresh search of possible adjacent nodes at regular intervals to discover potential new paths and new adjacent propagator nodes.

Structuring a Tree—with Redundancy

In a logical view, the typical Internet of Things relationship will be one or a few integrator functions to thousands or millions of end devices. Given the basic premise that only a branching tree may scale to the huge network size inherent in the Internet of Things, the most efficient overall network topology will thus take the form of a tree with limbs and branches at the high-volume "end device" edge of the network.

But there may be many possible paths discovered by each propagator node as it examines the data provided by adjacent propagator nodes. Without an overall computation of the entire network structure (which would be impossible), some method is required to avoid circular routing paths.

Individual propagator nodes create the necessary tree structure as they populate their routing tables with potential paths. Alternate possible paths that are deemed less desirable due to hop count, bandwidth, or quality history are noted but not activated. Instead, they are retained as potential backup paths if the primary chosen path is lost for a significant amount of time, as shown in Figure 4-4.

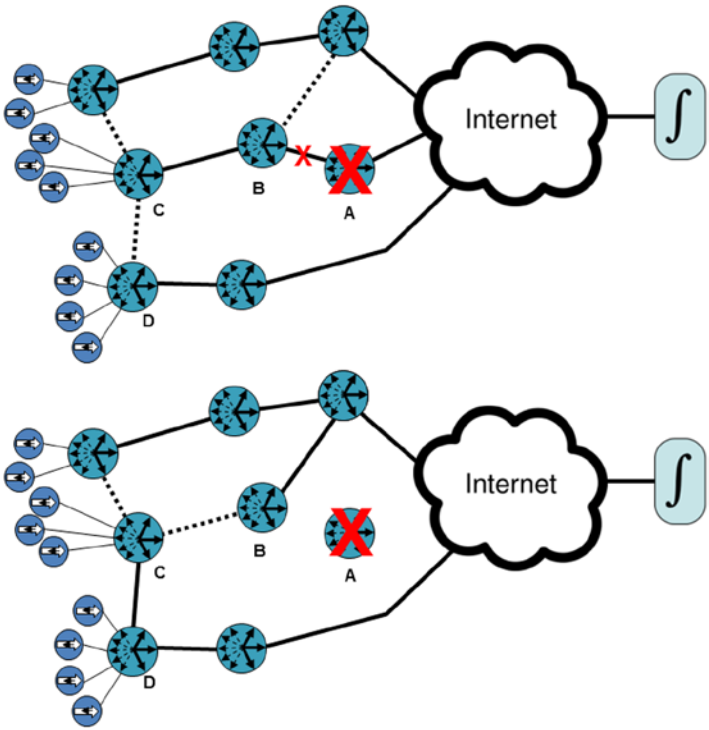

Figure 4-4. *Alternate paths that are not in use are maintained in propagator node routing tables for use as alternate paths in case of the failure of a data path or adjacent propagator node. In the example above, a failure at Node A causes Node B to activate a back-up route. But Node C makes its own routing decision and may select Node D as a better choice rather than following its previous "parent." The tree structure (no loops) is maintained in the new routing*

In this way, failover to an alternate path may take place reasonably quickly. Routing path data will be regularly "aged out" of the table as required to keep the information on possible redundant paths as current as possible.

Housekeeping

As noted previously, a small amount of link and path quality data must be regularly exchanged between adjacent nodes as a housekeeping message. In order for this to be reasonably efficient, there are two classes of information exchanged, circulated to all known adjacent nodes.

A "full" housekeeping message contains a complete "snapshot" of information on adjacencies and link paths from each node and is generated and broadcast every 60 to 600 seconds. The full housekeeping message would typically be in the range of 1,000 to 2,000 bytes of data. A "light" housekeeping packet includes only changes from the last "full" update and is generated every 15 to 60 seconds, with a size of 10 to 100 bytes of data.

When there have been no changes, this lightweight packet provides a confirmation to adjacent propagator nodes that the broadcasting propagator node is still functioning.

This means that a new propagator node joining the network must wait for a full housekeeping packet before it can complete its path analysis. But even light housekeeping packets provide useful information by indicating the presence of an adjacent propagator node.

Propagator nodes also maintain tables of the identities of attached end devices and report this information to adjacent nodes via full housekeeping packets.

By Any Means

To this point, there has been no distinction made between the different possible networking protocols used for connections between propagator nodes. This has been intentional, as the general network decision-making is the same. Individual link paths are abstracted as different channels, each with its own weighting, see Figure 4-5. In some cases, the link between propagator nodes may be simple chirp protocols; in other cases, full TCP/IP connections via the traditional Internet.

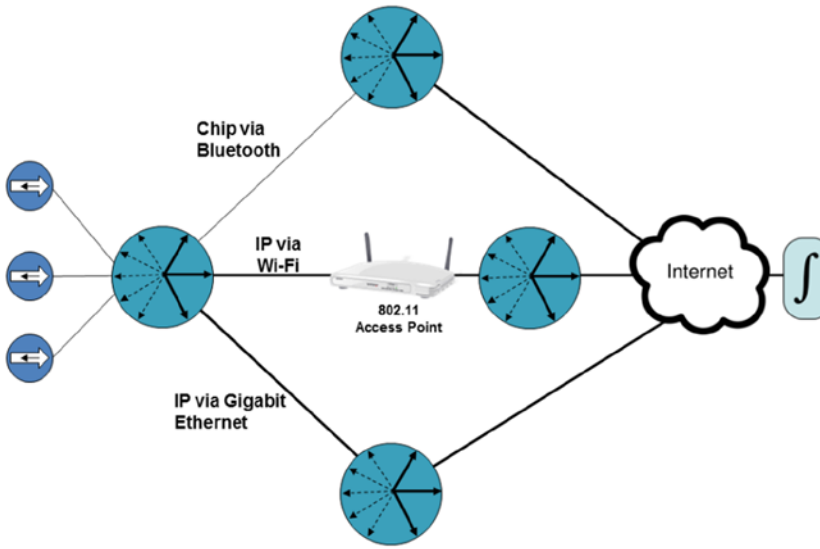

Figure 4-5. *Propagator nodes treat every possible link type as a different channel, abstracting the route-decision algorithm from specific protocols. Although operating at vastly different speeds, any of the three links from the propagator node at left could be used as a path to an integrator function via the Internet*

In the latter case, the propagator node will typically use Dynamic Host Configuration Protocol (DHCP) so that it can communicate on the network using the IP. Housekeeping packets between propagator nodes will be encapsulated within TCP/IP (along with all the other traffic via that path, of course). As noted before, chirp protocol communication and the associated end devices are isolated “behind” the propagator node network.

Take Out the Thrash

As with any networking system, it is critical to avoid hysteresis and thrashing in the structure of the networking "tree" formed by propagator nodes that might be caused by rapidly changing or inconsistent path quality. This might be caused by degraded or failed wireless links or the loss of one or more propagator nodes due to power outage or equipment failure. These changes in the availability and/or quality of individual links or nodes will "ripple" through adjacent propagator nodes in the path of the logically structured tree. The ability to manage such perturbations is especially critical in the Internet of Things, where communications at the edge of the network may be intermittent and of low quality as a matter of course.

Fortunately, the generally low data rates and completely uncritical nature of any individual transmission of the Internet of Things (see Chapter 2) means that a relatively high damping may be applied to the path determination algorithms. The goal is only reasonable efficiency, not overall network optimization, again because of the unique nature of the traffic on the IoT and the capability of propagator nodes to monitor, prune, and tune overall network performance. The tree-based topology ensures that individual propagator node decisions to route via one propagator node versus another are driven by the overall tree route efficiency. This is covered in more depth in Chapter 6.

The Power of Bias and the Role of the Integrator Function

The previous general description includes the basic network capabilities common to all propagator nodes. But the greatest power of the Internet of Things will come as integrator functions create vast networks of data streams encompassing very large numbers of end devices. Based on "neighborhoods of interest" and "affinity" (fully described in Chapter 5), the tiny chirps of end devices are aggregated into small data streams at the propagator nodes, coalesced into big data, and then transformed into useful information at the integrator functions.

This is the essence of the publish/subscribe model in the context of the Internet of Things: the end devices simply broadcast data in the form of chirps without any knowledge of how or where this data will be used. The integrator functions independently create neighborhoods of interest by selecting from available data sources.

For efficiency's sake, it makes sense for the path this data takes, from end device through propagator nodes and on toward the integrator functions, to be actively and intelligently managed as a publish/subscribe model defined by the integrator functions.

This will be achieved by a publishing agent within some classes of propagator nodes (Figure 4-6). This publishing agent may be biased by instructions from one or more integrator nodes to create specific data paths and/or bundle chirp data in specific combinations. Because chirp data is inherently self-classified by external markers, publishing agents may act upon the data by type.

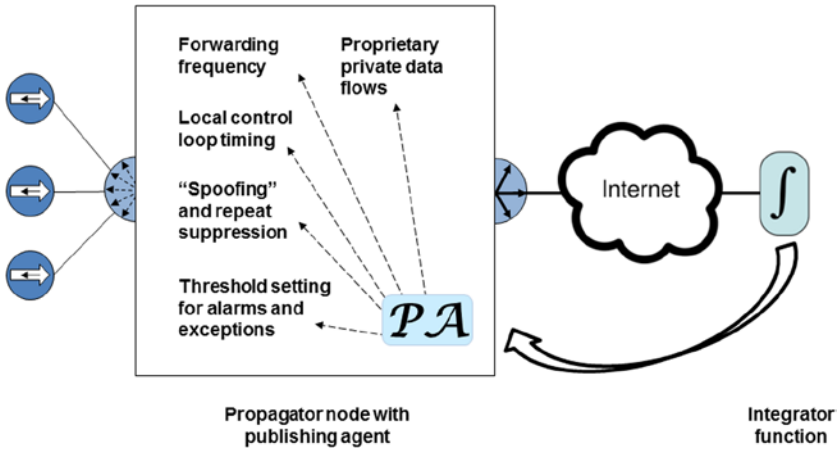

Figure 4-6. *To unlock the full power of the Internet of Things, some classes of propagator node will contain a publishing agent that may be directed by one or more integrator functions to create and modify a publish/subscribe data flow*

Because most of the data paths in these types of relationships will be via the traditional Internet, TCP/IP protocols will be the norm for links from the propagator nodes to integrator functions. This is a logical accompaniment to the publish/subscribe model, in which the end points are known.

The relationship between the integrator function and publishing agent in the propagator node will often be proprietary. For example, a particular manufacturer may provision a publishing agent in its own line of propagator nodes for specific use with that same manufacturer's integrator function. Although the propagator node might also function generically for other Internet of Things traffic, data from specific types of end devices might preferentially be packed for publishing to that integrator function.

Bias and Influence

Although the proprietary relationships described previously will be more typical, there may also be situations where the data being aggregated by a particular propagator node is required by multiple integrator functions for multiple applications or users, either simultaneously or over time.

The publishing agent will respond to the most recent and most frequent biasing messages from the integrator functions. More frequent and more recent messages will reinforce an existing publish/subscribe relationship, while less-frequent messages will allow the propagator node to revert to a more generic function of simply propagating all chirps promiscuously. Over time, this means that the publish/subscribe model may shift organically in response to changing needs, seasonality, events, and so on.

One could consider the example of transit bus schedules, in which routes and timing are managed centrally. The transportation of chirp–cased small data is similarly driven by the needs of the big data centers and their subscription preferences for the published small

data flows. Setting up the schedules and routes is managed from the top down because bias and interest in some chirp streams change when and how chirps are transmitted.

In effect, the relationship of integrator functions and propagator nodes is a form of software-defined networking driven by publish/subscribe agents operating on behalf of integrator functions—completely independent of the physical network topology. This is more fully described in Chapter 5.

Degrees of Functionality

Varying end user requirements and applications will create the need for multiple classes of propagator nodes (see Figure 4-7). Specific types of propagator nodes will contain a publishing agent that may interact with one or more integrator functions. But other classes of propagator nodes will function more generically.

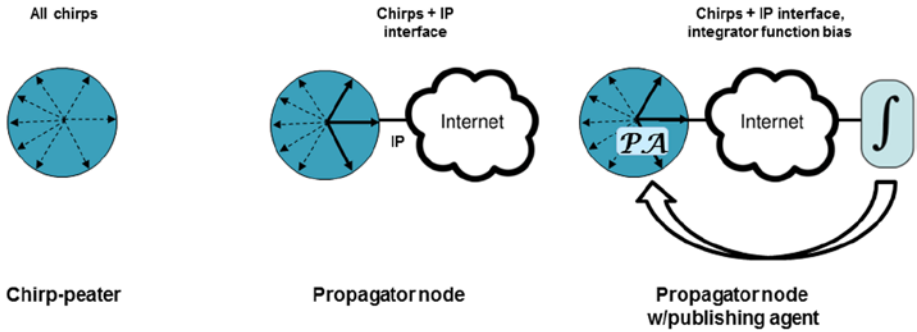

Figure 4-7. *Depending on the application, differing levels of propagator node functionality will be needed. Data forwarding is more selective moving left to right*

The most basic class of propagator node might be dubbed the "chirp-peater." Aggregated chirp messages for associated end devices are simply packaged and forwarded to one or more adjacent propagator nodes. This simplest class of propagator nodes will include only chirp interfaces, with another nearby propagator node providing TCP/IP gateways and other functions. One version of this class of propagator nodes may be designed to act as a client to an 802.11 access point for easiest integration of chirp protocol end devices into existing wireless networks in the home and office.

More powerful propagator nodes will be equipped with more sophisticated networking protocol stacks, gateways, and interfaces. Key among these will be TCP/IP gateways that permit routing through the Internet. They can be used for connections to integrator functions, for propagator-node-to-propagator-node links, and for integration of end devices that include a full TCP/IP stack. Some percentage of these fully featured devices will include the publishing agent described previously, which may often be part of a proprietary publish/subscribe overlay on the general propagator node functions.

But many propagator nodes will be deployed in a "promiscuous broadcast" mode, transporting all received traffic based on the "arrow" of transmission contained within the chirp packet markers. Although there will be little or no routing specificity in these

transmissions, there *will* typically be management of repeated transmission, broadcast trimming and pruning, and so on (described below and more fully in Chapter 6). Propagator nodes deployed in this mode will transport IoT traffic on behalf of any device and may become an important part of public and open source Internet of Things networks of unprecedented scope that have yet to be fully conceptualized.

Aggregating End Points

The other "side" of the propagator node consists of the array of interfaces facing the chirp-equipped end devices. Here too, propagator nodes will have many different physical and logical interfaces, both wired and wireless. Beyond traditional interfaces such as Ethernet, 802.11 Wi-Fi, Bluetooth, and so on, wide usage of optical interfaces such as infrared and other low-cost alternatives such as power line networking will also be found, as seen in Figure 4-8.

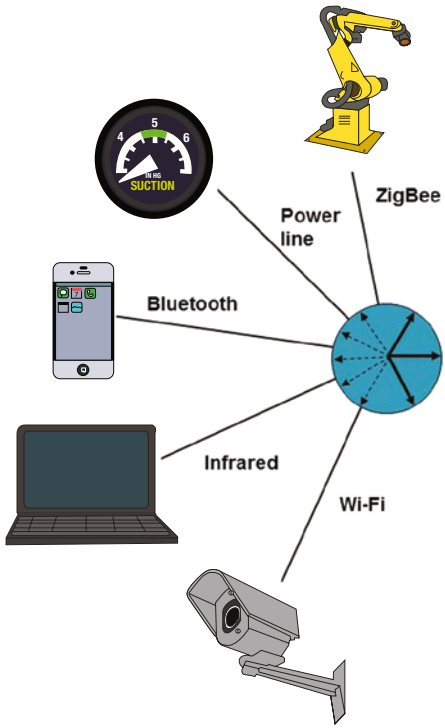

Figure 4-8. *A wide variety of end device physical interfaces may be accommodated, all communicating via chirp protocols. Propagator nodes will vary in the type and number of interfaces provided based on user requirements*

Whatever the mix of physical interfaces chosen, the chirp will be the fundamental data interface to most end devices. As noted in Chapter 3, there will be many bidirectional end devices, but the majority will likely be primarily or solely simplex, whether transmit or receive. In addition, as discussed before, many of these devices will have relatively low information rates, whatever their transmission rate. In other words, in many cases there will be a tremendous amount of repetition in the data that is sent or received by end user devices, which is the subject of the next section.

Dumping the Dupes

Devices such as pressure and moisture sensors, depending on the granularity of their measurement capabilities, will likely send the same value repeatedly for long periods of time. In the reverse direction, the valve servo in a process control application may remain in the same position for extended durations. So in these cases, there will be repeated reports or commands of identical data being sent.

More sophisticated propagator nodes will be designed with consideration of this excessive duplication of data that will likely be a hallmark of much of the Internet of Things. Data streams will be monitored and duplicate messages deleted and/or spoofed locally to avoid transmitting unneeded repetitive data to-and-from integrator functions.

Especially for those propagator nodes equipped with an internal publishing agent (described previously), the integrator functions may bias the propagator node to transmit only data indicative of readings that exceed certain thresholds in frequency and/or value.

These propagator node capabilities will limit the amount of IoT data to be transmitted. Even though individual chirps are much more compact and efficient than traditional protocols such as TCP/IP, the massive scale of the Internet of Things makes it critical to limit inconsequential repeated data wherever possible. Techniques to be used are more fully described in Chapter 6.

Loading the Bus: The Propagator Node Transit System

Another key function of propagator nodes will be managing and packaging broadcasts at all levels in the network. Lightweight chirps are ideal for the typical low-speed, low-duty cycle communication between end device and propagator node in the IoT. But if each of these chirps is then enveloped individually in a (relatively) huge TCP/IP packet before forwarding to the next propagator node, all efficiencies are lost.

Instead, propagator nodes will use their knowledge of adjacencies and routes through the network to accumulate chirps that may be efficiently forwarded *together* to the next propagator node (as noted previously, repeated chirps may be deleted or spoofed). At each successive node, these “buses” may be unloaded, some packets removed and others added, and then forwarded again, as shown in Figure 4-9.

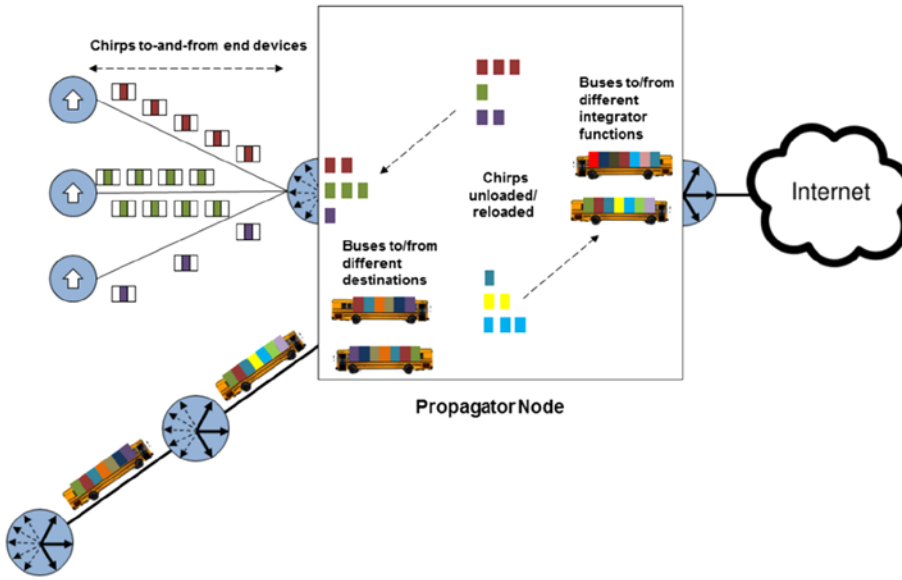

Figure 4-9. *For maximum efficiency in communication between propagator nodes, a "bus" departs periodically for adjacent propagator nodes or integrator functions via the traditional Internet or other data paths. Bus size is optimized for the particular path. At an intermediate propagator node, "buses" are reexamined, local traffic removed, and additional onward traffic added as appropriate*

This process of consolidation, pruning, and forwarding adds a delay at every intermediate point, both for processing time and a lag as the propagator node waits for a certain period to fill the "bus" as much as possible before transmitting. But in the world of the Internet of Things, these delays will have no impact on the usefulness of the data.

Bus sizes will be chosen based on the characteristics of the channel over which they will be forwarded. For TCP/IP paths, propagator nodes will attempt to fill out a packet before forwarding. For other paths, the "bus" size will also be adjusted for maximum efficiency.

Where the publishing agents in specific propagator nodes have been biased by an integrator function, these routing preferences and (typically) TCP/IP packet characteristics will take precedence over the more mechanical process defined above.

Weathering the Storms

In any network capable of broadcasts, the potential for debilitating broadcast storms is present. Propagator nodes inherently limit broadcast storm propagation through the overall tree-like structure, deletion of repeated data, and consolidation and pruning of broadcasts to and from end devices.

Detailed descriptions of the propagator node techniques used for managing traffic to-and-from end devices are found in Chapter 6.

Dodging the Collisions

As noted in Chapter 2, the simplified chirp protocols incorporate no error checking, collision detection, or collision avoidance. Instead, simple randomization schemes and variable back-offs ensure that the tiny chirps may be squeezed between other transmission in the same wireless spectrum without the risk of a "deadly embrace." Propagator nodes use these same techniques on the chirp interfaces.

What's in a Name?

A key premise of the emerging architecture for the Internet of Things is that end user devices are burdened with only the very simple chirp protocol. As described in Chapter 2, the names applied by end devices in chirps are incomplete and likely will not be unique across the network.

This limitation would be problematic if chirps were simply transmitted as-is to other devices. But propagator nodes provide the additional context and addressing specificity needed to create unique addressing, as shown in Figure 4-10. These details are developed from the routing table adjacencies and other information available to the propagator node, as described in detail in Chapter 6.

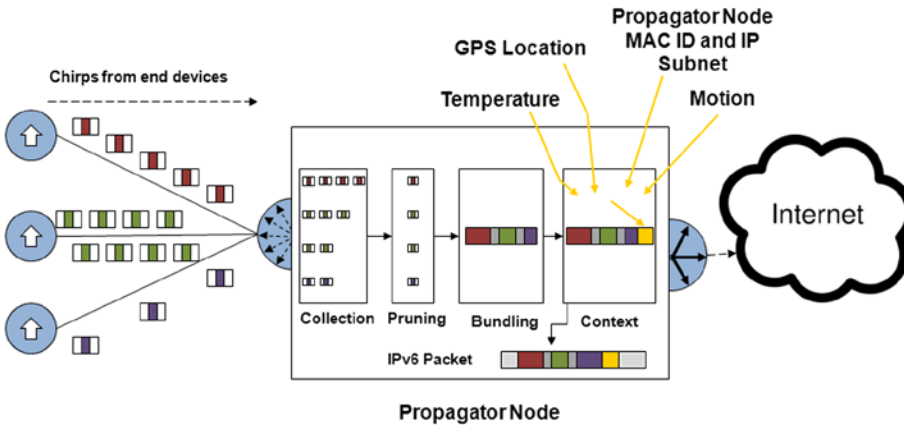

Figure 4-10. *As chirps are bundled within propagator nodes, additional location, addressing, protocol, and other information is added*

Propagator nodes may then "publish" these small data streams onward toward the appropriate integrator function via the propagator node network or, with addition of the appropriate IPv6 encapsulation, directly via the traditional Internet.

For data whose "arrow" points toward the end device, the procedure is reversed: headers and formatting needed for routing to the target propagator node are stripped by that device, and only a lightweight chirp is transmitted to the end device using that device's simple non-unique address.

Packaging Options

There will be many packaging combinations of end devices, propagator nodes, integrator functions, and so on. A particularly interesting combination may be a propagator node with an on-board specialized integrator function. An example of this combination might be designed for local analysis of video surveillance and alarm data, with only exceptions and unusual combinations of data being propagated up to a central site.

Propagator nodes will certainly be packaged with existing types of networking and home entertainment equipment, including routers, Wi-Fi access points, LAN switches, set-top boxes, and so on. There will also be packaging options with nontraditional devices such as smart meters, vehicles, televisions, air conditioning and lighting equipment, and various household appliances as shown in Figure 4-11. Propagator nodes may require little or no human intervention and may be unobtrusively packaged as a wall wart or in other inconspicuous form factors.

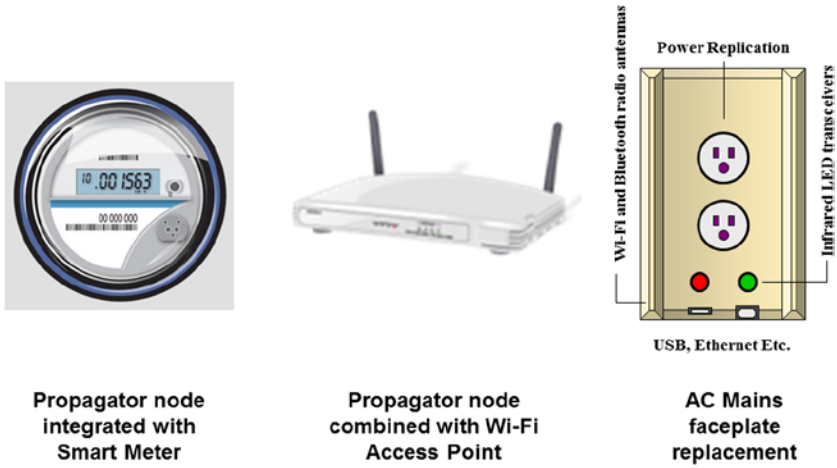

Figure 4-11. *Propagator nodes will be available in many form factors and in combination with other devices from the IoT, including end devices and integrator functions*

Commercial environments will find propagator nodes (often in combination with end devices) in manufacturing equipment, process control devices, vehicles, and many more.

Although it is likely that some instantiations of the propagator node will be software-only on platforms such as smartphones, tablets, or PCs, these devices typically will have two limitations: insufficient number and variety of interfaces for connecting to end user devices and the transient nature of their location.

Packaging options and example network configurations are further discussed in Chapter 7.

Building Blocks of the IoT

Propagator nodes truly are the fundamental components of the tree-like structure of the emerging architecture for the Internet of Things. Propagator nodes create reasonably efficient networks for the transport of IoT data while controlling broadcasts and eliminating unnecessary repetitive data. They make possible the conversion of the lightweight protocols at the edge of the network to the more robust protocols demanded in the traditional Internet and elsewhere.

The next chapter will explore the "business end" of all these data flows: the human-facing integrator functions.