CHAPTER 1

The Evolution of Power over Ethernet

PART 1 AN OVERVIEW OF ETHERNET

Introduction

We seem to be in one of those rare moments in our history when absence has paradoxically emerged as the strongest proof of omnipresence. Since its rather humble beginnings back in 1973, Ethernet has metamorphosed so dramatically, it bears almost no resemblance today to its origins. As testimony to its ever-increasing popularity, 95 percent of all local area networks (LANs) in existence today are estimated to be *Ethernet*-based. But the irony is Ethernet may likely never have been around in its current form had it not been conceived way back then in its now-extinct form.

The basic purpose of Ethernet remains unchanged—to connect computers, printers, servers, and so on in LANs so they can exchange information and services among themselves. Although Ethernet continues to be a *packet-based* computer-networking technology, even its packets of data ("frames") are no longer exactly as originally envisaged. So much has changed in Ethernet that some very notable people have gone as far as to say Ethernet is (now just) a *business model*.

But first, to dispel a popular notion in a timely manner: Ethernet is *not* the Internet and vice versa (despite sounding similar). Internet actually came into our lives a bit *after* Ethernet did—specifically in 1989 when the first commercial Internet service provider (ISP), aptly named "The World" offered its services to the general public. Internet is, quite literally, an *inter-network*, a global *network of networks*. By now, most of the networks linked by the Internet do happen to be *Ethernet*-based, and nearly all Internet traffic starts or ends on an Ethernet connection. So it is obvious that a good part of the reason for the explosive growth of Ethernet is that it so naturally complemented the exponential growth of the Internet over the last few decades.

More recently, another very similar symbiotic growth pattern has quietly emerged from the shadows, perhaps a bit unnoticed. Riding on the remarkable growth story of Ethernet, figuratively and literally, Power over Ethernet (PoE) is now seeing a huge upswing of its own and seems to be now driving growth by itself. New families of PoE-capable network appliances and Ethernet equipment have suddenly started appearing. An ever-increasing percentage of "ports shipped" today are PoE-enabled, and PoE seems to be fast-becoming a default choice for Ethernet ports. It is therefore increasingly important for us to recognize PoE for what it is—a very fast-emerging technology. We should try and understand it much better before it gets ahead of us.

From the viewpoint of chip and systems designers, some rather unusual challenges are associated with implementing PoE. At a higher level, PoE represents the cusp of two major, hitherto parallel worlds of electronics development: *power* and *networking*. PoE is, in ways, the virtual confluence of the Applied Power Electronics Conference, (APEC) and Interop® (the annual networking expo at Vegas). It symbolizes the convergence of digital and analog hardware and software, power and data, and so on. It is exciting—and therein lie the challenges too.

If we look around, we may notice that things are not going so well on the *human* plane. Hardware personnel are still quite prone to ignoring software personnel; analog designers often shun digital designers, and so on—and vice versa of course. In the oft-repeated (though infamous) words of a famous analog legend of Silicon Valley, the late Bob Widlar: "Any idiot can count up to 1." No surprise that if we look around, we may notice several mutually suspicious "knowledge cliques" hovering around us. Unfortunately, *skill segregation* (specialization) is not commensurate with our modern world of increasing convergences. Personally, as systems and chip designers, we just cannot afford to allow our skill sets to become so increasingly finely tuned and subdivided anymore.

We need to illustrate this issue a little better, lest it sound hyperbolic to some. In 2006, one such engineer, let's just call him Bob for now (another Bob), recalled an ancient bench struggle he had faced decades ago. These are his words [italics added by the author of this book]:

> I designed this thing. It took me forever, *a good portion of a year*. I finally had it, and I got it working with my test programs. It would work, but it only worked for 15 minutes and *then it would go "bah,"* and all the lights would go in the wrong direction and then it would die. Then I would reboot, and then it would run again. I had these extensive diagnostics, random patterns, and everything, to just test the hell out of the thing, different word lengths, test every edge case. Then it would run just fine, over and over again through every—and then it would just die. *I worked on it for a month and I couldn't figure it out*…I went down to the other end…. Tom looked at

> it, and he asked me a few questions, and then he said, 'Well, I know what's wrong. . . . *you don't have any bypass capacitors on it*. Now I know you just took a bunch of courses in digital electronics but *at some point everything is analog*. So what's happening, Bob, is when certain patterns get into your registers and they all go 1, and all the transistors turn on, they take too much current, too many electrons, and then the voltages start to droop because you've sucked in all those electrons and then all the digital devices start to malfunction. So what you need to do, Bob, is *sprinkle some bypass capacitors here and there* to store up some extra charge for those cases when you have lots of 1's in your registers.' . . . . I went quickly back to the lab and got my soldering iron, and I soldered on—I probably put a bypass capacitor on every third socket to this huge board. And I plugged the sucker in and *she worked for the next 13 years*. Anyway, I learned a lesson there—the *analog digital*—which came in handy later because *Ethernet is a combination of analog and digital itself.*

Now add to the anecdote two not-so-obvious facts:

1. PoE had not even come into the picture at the time of my bench-wrestling story, and the design "gotchas" were already appearing fast on the horizon.
2. The engineer in question was none other than *Robert ("Bob") Metcalfe*, considered today to be the "inventor of Ethernet."

So, we can imagine, that if Metcalfe was confused, more so *without PoE on the scene* yet, what we may experience *today*. That thought is humbling. Therefore, one of the key objectives of this book is to try and narrow down the differences between power and networking and analog and digital, and so on. Because if we don't, we may take almost forever to get to a *successful*, commercial product down the road (*cable* in this case).

One last question before we move ahead: *Who invented Power over Ethernet (PoE)?* Was it Bob? Or John? Was it Harry? Why not Jane? Actually, *none of these*. The correct answer lies buried somewhere in the *19th* century—*J. J. Carty* was his name. And that is a true eye-opener, especially to some gung-ho "modern-day" engineers. In fact, it's actually over a hundred years ago that the basic principles underlying PoE started to take shape. So, in that sense, Ethernet actually came *after* PoE. It's funny how the world goes round and round. If not, it would probably all go "bah," in the words of Metcalfe.

And so, for just a moment longer, let's return to the future—that is, back to *Ethernet*.

A Brief History of Ethernet

Ether Network, Ether Net, EtherNet, Xerox Wire, X-wire… That's what modern Ethernet *almost* got called. And had it, very likely it could have ended up in our collective memories (and Wikipedia) today

as the "the now-defunct proprietary networking technology from Xerox Corporation." In fact, even the term "Ethernet" was originally a registered trademark of Xerox Corp. Fortunately for all of us, and perhaps even for Xerox, Xerox got talked out of all its rights on this subject by Robert Metcalfe himself and agreed to work with Digital Equipment Corporation (DEC) and Intel to spread its version of networking technology far and wide. Shortly thereafter in 1979 Metcalfe founded 3-Com Corp (last acquired by Hewlett Packard in 2010) but continued to work with the consortium, informally called DIX (for DEC, Intel, and Xerox). Together, they published the first formal (industry) Ethernet standard, DIX V1.0 in 1980. Meanwhile, the Institute of Electrical and Electronic Engineers (IEEE), in an effort to standardize LAN, had their very first meeting on this subject early the same year (1980). Because of the timing, some say that the well-known modern Ethernet standard 802.3 got it basic name—802 coming from February '80 or 2/80. Others say 802 just happened to be the next-available number in the normal sequence of IEEE standards. Either way, DIX approached IEEE to help standardize (their version of) Ethernet. But things were just as they are on any typical day at IEEE even today. In walked pushy General Motors with a rival LAN proposal called the Token Bus. Not to be outdone, IBM walked in with their Token Ring. As a result, IEEE bowed and decided to standardize *all three* proposed LAN standards. And that is how the IEEE "dot committees" got created: 802.3 was Ethernet, 802.4 was Token Bus, and 802.5 was Token Ring. These were all later blessed for international acceptance by the International Standards Organization (ISO), and became, respectively, ISO 8802-3, 8802-4, and 8802-5. Yet despite the initial boost, the latter two eventually died from "natural causes," though Token Ring does seem to have hung on rather stubbornly—for close to 15 years as per Metcalfe's estimate. Some think it is still around somewhere. But no one contests the fact that, in contrast, Ethernet grew extremely rapidly.

So, why did Token Ring in particular, lose out to Ethernet? One reason was that IBM was charging prospective vendors heavily in terms of royalties for producing Token Ring cards and medium attachment units (MAUs), or simply, the cable-driver electronics (akin to *transceiver* or PHY in modern lingo) placed between the controller card and the cable. This made Token Ring equipment too expensive overall. A Token Ring card itself could cost 5 and 6 times as much as an Ethernet card. Add to that more expensive cabling, and Token Ring literally priced itself out of the market.

Besides cost, a big advantage of Ethernet, going forward, was its inherent *flexibility*. On December 21, 1976, in an internal memo at Xerox, Metcalfe explained a key advantage of Ethernet in the following words (italics inserted here belong to the author of this book):

> The OIS [Office Information Systems] protocol is based on *distributed* many-to-many communication as required by *incrementally grown* and

increasingly interconnected office systems, rather than *hierarchical* mainframe-centered data processing systems.

The way Metcalfe originally visualized Ethernet was a *single long coaxial* cable connecting many computers (workstations) and printers. In modern terms, this is a "bus topology." A new device (computer, printer, and so on) could be simply added on, as the need arose, using a "vampire tap." This contained a needle (let's call it a Dracula tooth to be visually clear and consistent) that would get clamped down and penetrate the coaxial cable to make contact with its inner conductor, while the outer shield of the cable would connect to the outer shield of the newly added segment. So the network could be built up steadily over time, rather than needing a big central infrastructure right off the bat (preguessing future needs). This proposed type of LAN architecture was envisaged to grow along with the size of the organization.

The bus topology is somewhat akin to a giant plumbing system with a main water pipe . . .(but with data, not water) flowing down, with several feeder connections on it along the way (see the hybrid architecture example in Fig. 1.1).

It is indeed ironic that in later years, the basic framework of Ethernet (in terms of its *packet-based architecture* and supporting techniques) proved flexible enough to allow moving *away* from the original bus topology concept to a "star topology." In this new architecture, every computer (or networking device) gets connected via a dedicated cable plugged into a central switch or hub. In terms of hardware expandability, the star topology is not as flexible as the bus topology, but it can provide much higher speeds, besides other advantages. As PoE designers, we should also realize that the bus topology could never have supported PoE as it is today. The star topology is a prerequisite for power over data cables, one cable for each end-device. So clearly, things seem to have gone in the right direction, both for data and power. That's survival of the fittest.

Another reason for the continuing rise of Ethernet was that Ethernet got suddenly empowered along the way by something called a "switch." This concept seems to have originally come out of a start-up called Kalpana (Hindi for imagination or vision), cofounded by Vinod Bhardwaj, an entrepreneur of Indian origin. Kalpana was acquired by Cisco in 1994, ten years after Cisco itself was born. The switch eliminated a very basic problem of *collisions*, which was slowing down networking in general. The switch eventually helped achieve much higher data rates. It soon became unstoppable because it catered to the rapidly growing need for speed.

What are collisions? Quite similar to what you would expect would happen if hundreds of cars were let loose in both directions on a *single lane*. More formally expressed, collisions can be explained as follows: (Data) collisions occur when, for example, all the computers

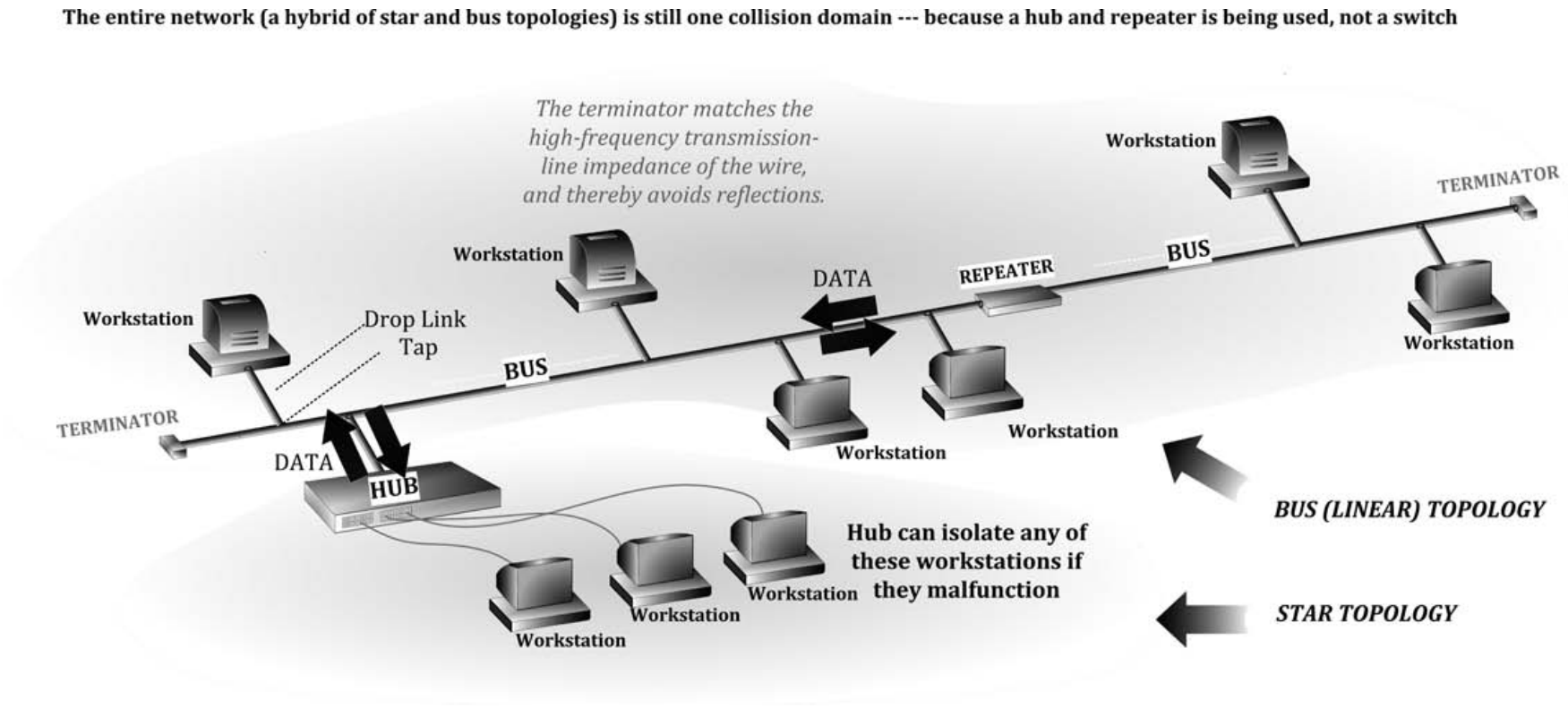


FIGURE 1.1 Older Ethernet LAN architecture showing bus/star topologies and hub/repeater.

on a shared line start "talking" (transmitting data packets) simultaneously. What results is almost noise (garbage). It is also very similar to a whole bunch of people brought into a small room, each person trying to talk over everyone else's head to someone else in the room. Pretty soon, with all the din and shouting, no one really understands a thing anymore. We have all been there and perhaps done that too. To avoid this unpleasant and *inefficient* situation, the connected computers need to detect that a collision is occurring, then *back off* and try a little later again. In doing so, they must *not* try *simultaneously* again, or a clash will recur, slowing down all communication for an even longer time. On the other hand, if they wait too long before trying once again, the communication slows down again. But if they try too fast again, they run the risk of overlapping (talking simultaneously), causing more delays. And so on. That is where Metcalfe originally came into the picture. His claim to fame is U.S. patent number 4,063,220, titled "Multipoint Data Communication System with Collision Detection." This is basically a (statistical) *algorithm to back off* and try again in an optimum manner. In contrast, and in a more *deterministic* manner, IBM's LAN architecture had a software "token" that was moved around in a circle of connected computers, and so whoever had possession of the token at a given moment, got the right to transmit. It was a little like the game of "passing the parcel" at a typical children's birthday party. Token Ring did seem to be superior to Ethernet initially, especially to engineers who felt uncomfortable with the lack of clearly defined timings characteristic of Metcalfe's software-based algorithm.

Note that in 1985, IEEE finally published a portion of the ongoing standard pertaining to Ethernet: IEEE 802.3 titled "Carrier Sense Multiple Access with Collision Detection ('CSMA/CD') Access Method and Physical Layer Specifications." We can see the title does not even mention the word "Ethernet." But Metcalfe's original term did catch on in a big way. And so the IEEE 802.3 standard was, and still is, referred to as the Ethernet standard.

CSMA/CD can be explained in informal language as follows:

1. CS: Carrier Sense (*Hey, do I hear someone talking?*)
2. MA: Multiple Access (*Careful, we can all hear what each one of us is saying!*)
3. CD: Collision Detection (*Hey, we're both talking now—stop!*)

The underlying logic of CSMA/CD is

1. If the medium is idle, transmit *anytime*.
2. If the medium is busy, wait and then transmit *right after*.
3. If a collision occurs, send 4 bytes of "jam" signal to inform everyone on the bus, then back off for a random period, and after that, go back to Step 1 above.

The last point is pretty much the social technique we use rather intuitively, in a normal, polite group or conversation, say at a dinner party in the evening, as opposed to trying to talk, or rather shout, at each other in a crowded bar (perhaps with 100 dB of rock music playing in the background).

As mentioned, with the entry of the switch, the basic problem of collisions was eliminated altogether. It was a huge boost for Ethernet, both in terms of speed and market popularity, and eventually this allowed Ethernet to move up to much higher speeds: 1 Gbps (gigabits per second), 10 Gbps (over copper), and beyond, as of today. Though in the first step, Ethernet just went from 10 Mbps (megabits per second) to 100 Mbps. That actually happened without even requiring a switch, just by using a "hub," described further below. Yet, even that was enough to start making the 16 Mbps of the IBM Token Ring architecture obsolete. The Token Ring concept, however, did seem to get a fresh lease on life in Hewlett-Packard's (HP's) 100-Mbps LAN architecture called 100VG-AnyLAN in 1995, but that was virtually extinct by 1998 too.

The advantage of "switching" in the area of Ethernet can be explained in modern terms as follows: Today, every device on a network has a unique self-assigned identifier, a hardware address, called its Media Access Controller (MAC) address. The media in this particular case is the copper wire, over which signals are sent. When a computer sends data on the network, it sends it in packets (frames). Each packet carries information describing the source and (intended) destination. Network switches (or just switches) are devices smart enough to read the MAC addresses and direct the packets from the source to the intended recipient device. In other words, switches do *not* broadcast data to all and sundry. Unlike a hub, switches do not talk loudly as over a public announcement (PA) loudspeaker, preventing others from talking when they want to. But with a little thought, it should also be clear that for switching to be successful, the shared bus topology of the original Ethernet needs to be replaced by the *star topology*, in which each computer (or node) gets its own Ethernet cable, and all Ethernet cables get plugged into the switch (into its available ports, or jacks). With such a topology, conversation can be *physically* directed from source to destination in a planned manner over dedicated cables. Note that hubs also connect to computers in star topology; the difference being that hubs are not smart enough to avoid collisions completely by inspecting and directing packets back and forth in a planned manner, as switches do. So (IEEE-compliant) hubs will just use CSMA/CD. On the other hand a switch is not even aware of CSMA/CD. It doesn't need to be.

Long before switches and hubs appeared on the scene, there were "repeaters." These devices just amplified the signal for sending across longer distances over copper, and extended the geographical reach of the LAN. No intelligence was built into repeaters. A little later, since it was

getting difficult to troubleshoot and isolate problems that were occurring on the shared wire, hubs were introduced, with the basic intention of creating a certain amount of *segmentation* (or segregation), along with security, within the LAN. Now, instead of several computers all hanging off one shared line, we could have several hubs connected to this line. Computers would then get connected to the hub in a star topology as shown in Fig. 1.1. Now if a computer malfunctioned, or perhaps just "talked too much," its hub could detect the condition, and shut off all communication to it (or to any other errant network device). This would isolate that device from the shared bus and prevent the bus from going down. Keep in mind, however, that with all hubs still using CSMA/CD, the entire LAN was still one big "collision domain."

At some stage bridges were introduced. The purpose was to create separate, *smaller* collision domains within the same LAN. Note that the entire LAN is still one, big "broadcast domain," but it now consists of not one, but several collision domains interconnected by bridges. This segmentation was found to be very helpful in a situation in which a very large number of computers were sharing the same line. Because, if they all tried to talk, even with a good collision detection/avoidance algorithm in place, they would eventually slow down the entire line considerably. So it was thought much better to have, say, two separate shared lines (or buses), with a bridge to pass data back and forth *when required*. Like two, bustling cities on either side of a river, connected by a bridge. People in either city lead separate lives, except when people from one city want to, or need to, go over to the other side. That's when they actually cross the bridge. Consider the contrasting case, with residents of both cities cramped into one city only, the traffic situation and congestion will only get much worse.

So looking back, the repeater was the dumbest of all. In between there were the bridge and the hub. The switch emerged as the smartest. Then, with the advent of the Internet, a device called a router appeared. It is even smarter or more powerful than a switch. As far as simple traffic routing *within* the LAN is concerned, a router operates exactly as a switch, learning the location of the computers on the LAN and routing traffic precisely to those computers. But the actual routing, as carried out by router, unlike as in a switch, is *not* based on MAC addresses, but on *Internet Protocol* (IP) addresses. Because ultimately, routers don't just allow different devices on a given local area network to communicate with each other, as switches do, but also allow *different networks* to communicate with each other over the Internet. To communicate between different networks, routers must have the ability to talk to other routers too (using IP addresses). In effect, a router becomes an interface between its LAN and the Internet.

When a router initially attempts to connect to the Internet, it requests an (external) IP address—one address for the entire LAN, much like a single postal address on a street for a huge building complex, though there may be several individual apartments or

single homes within that complex (that will eventually need letters delivered to and collected from). The request for this single external address is made by the router to a Dynamic Host Configuration Protocol (DHCP) server somewhere in the ISP's network. The router also distributes internal (local) IP addresses to all the devices (clients) on its LAN, to identify them (similar to assigning house/apartment numbers for residents). To accomplish communication between all these client devices and the Internet, the router uses network address translation (NAT). NAT involves modifying the source and destination IP addresses *within the packets*, so as to direct traffic appropriately between LAN clients and servers/devices on the Internet. NAT is in essence, a way to map all the devices within a network to a single external IP address. Why is it useful and/or necessary? In layperson's terms, if, for example, we want to connect just one computer in our home to the Internet, we do not even need Ethernet (no switches, hubs, routers, and so on)—just a direct connection and the use of Transmission Control Protocol/Internet Protocol. (TCP/IP) But what if we want not one, but three computers in our home to connect to the Internet (and talk among themselves too if possible)? And suppose we have just one incoming cable or digital subscriber line (DSL) connection. We do not want to pay our ISP for installing three separate lines/connections. Maybe they cannot do so either. So, as far as they are concerned, with the help of a router, all three computers can be made to appear as a single IP address (and we will of course get billed for just one connection/computer). Internally however, inside our home, the router distributes IP addresses to the three (or more) home computers. In effect, the router creates a small, switched LAN within our home, but also handles the back-and-forth exchanges from the home computers to the Internet via the IP's server, and all that in a manner that is transparent to the IP server. The IP server "thinks" it is dealing with just one computer inside our home (one IP address). In a sense, this is socially acceptable, mutually agreed-upon *deceit*. Summarizing, NAT becomes necessary when the number of IP addresses assigned by the ISP is less than the total number of computers on the LAN that need Internet access.

This natural morphing/evolving of Ethernet, combined with the growth of the Internet, contributed to the impressive rise of Ethernet. But another key reason for its success over rival architectures was that it became low-cost down the road. The star topology was the main enabler of that. Communication became possible (though over shorter distances of up to 100 m), using cheap, dedicated, "twisted-pair" copper wiring, as compared to the far more expensive coaxial cables required by rival LAN technologies (and by Ethernet itself originally). Note that *two* twisted pairs per connection were used for point-to-point communication: one for receive and one for send. So transmission and reception could now occur simultaneously too.

In other words, there were absolutely no collisions anymore: CSMA/CD could be forgotten forever, with just the flick of a switch (literally)! But note that still, data transfers were *uni*directional: Each pair worked in only one direction. Later in an attempt to achieve 1000 Mbps (1 Gbps), the electronics were made smart enough to ensure bidirectional communication over a single twisted pair by the use of a hybrid circuit as we will soon learn.

When it first started, Ethernet was just 2.94 Mbps, but that was because 2.94 Mbps happened to be the available system clock on Metcalfe's computer. Soon 10Base-2 and 10Base-5 appeared. Both these implementations ran at 10 Mbps, the first over about 200 m (hence the 2) of thin coaxial cable, the latter over 500 m (hence the 5) of *thicker* coaxial cable. But both required *coaxial wire*, and both are obsolete now. The follow-up IEEE 802.3 Ethernet standard was called 10Base-T, where "T" stands for twisted pair. This is also 10 Mbps, but, as mentioned, works over two pairs of twisted pair copper wiring (of American Wire Gauge number 26, or AWG 26). The deal breaker for Token Ring, however, was not 10Base-T, but 100Base-TX, which is 100 Mbps Ethernet over (two pairs of) twisted-pair wiring. This is called Fast Ethernet and is still popular today.

Modern Three-Layer Hierarchical Network Architecture

Having understood networking topologies, routers and switches, and so on, we take a quick look at Fig. 1.2, which represents a typical, modern three-layer Ethernet network architecture, mainly popularized by Cisco. It also includes a connection to the Internet. The basic purposes of the three layers can be summarized as follows.

Core Layer

This is the high-speed "backbone" of the "inter-network." The core layer is critical for maintaining interconnectivity between distribution-layer devices. The core must be available readily (and immediately), and also have some built-in redundancy to avoid a single failure bringing the whole network down. The core connects to Internet resources. It aggregates the traffic from all the (lower) distribution-layer devices. Core-layer switches/routers must therefore be capable of forwarding large amounts of data very quickly. And for that reason, core switches are more hardware-based than software-based. This helps reduce latencies that can arise from large number-crunching within software programs.

In small-business establishments, such as those called SMB (small and medium business), or equivalently SME (small and medium enterprise), the core and distribution layers may be one: as a single, "collapsed-core," layer.

Note that, sometimes, people call the core layer the "edge layer," and that can get confusing as indicated below.

Distribution Layer

The distribution layer aggregates the data received from the (lower) access-layer switches before it gets transmitted to the core layer for routing to the final destination(s). This layer also controls the flow of all network traffic in general, choosing the best (optimum) routes to send data between users on the LAN, also applying any relevant policies. For example, in a university we may want to separate the traffic according to faculty, students, and guests. Note that the switches used in this layer are typically high-performance devices too that have high availability and redundancy to ensure reliability.

Access Layer

The access layer interfaces with end devices, such as PCs, printers, and IP phones, to provide access to the rest of the network. This layer can include routers, switches, bridges, hubs, and wireless-access points (WAPs). The main purpose of this layer is to provide a means of connecting devices to the network, and also controlling which devices are allowed to communicate on the network at any given moment (their "access" privileges for example). Note that since end devices reside in this layer, PoE capability is most likely to be provided in switches, hubs, routers, and WAPs operating on this layer. This is also sometimes called the "access edge," or just the "edge," and it gets confusing because the core layer is also sometimes called the edge layer.

PoE is today provided even in switches meant for the upper (non-access) layers. For example, PoE capability may be present in the core-layer switches too, for powering customer-premises equipment (CPE). This would include any terminal and associated equipment, located at the subscriber's premises, connected to a carrier (or ISP's) telecommunication channel at the point of demarcation (i.e., where the line connects to the home/business wiring, and responsibility for maintenance gets handed over from provider to the customer/subscriber).

What Exactly Is "Ethernet"?

With all this evolution, what really was, or is, "Ethernet"?

In 2006, Metcalfe said: "Ethernet is [now] a *business model*." Metcalfe probably rightfully meant that Ethernet is now almost a *brand-name* of sorts, and bears almost no resemblance to what it originally was.

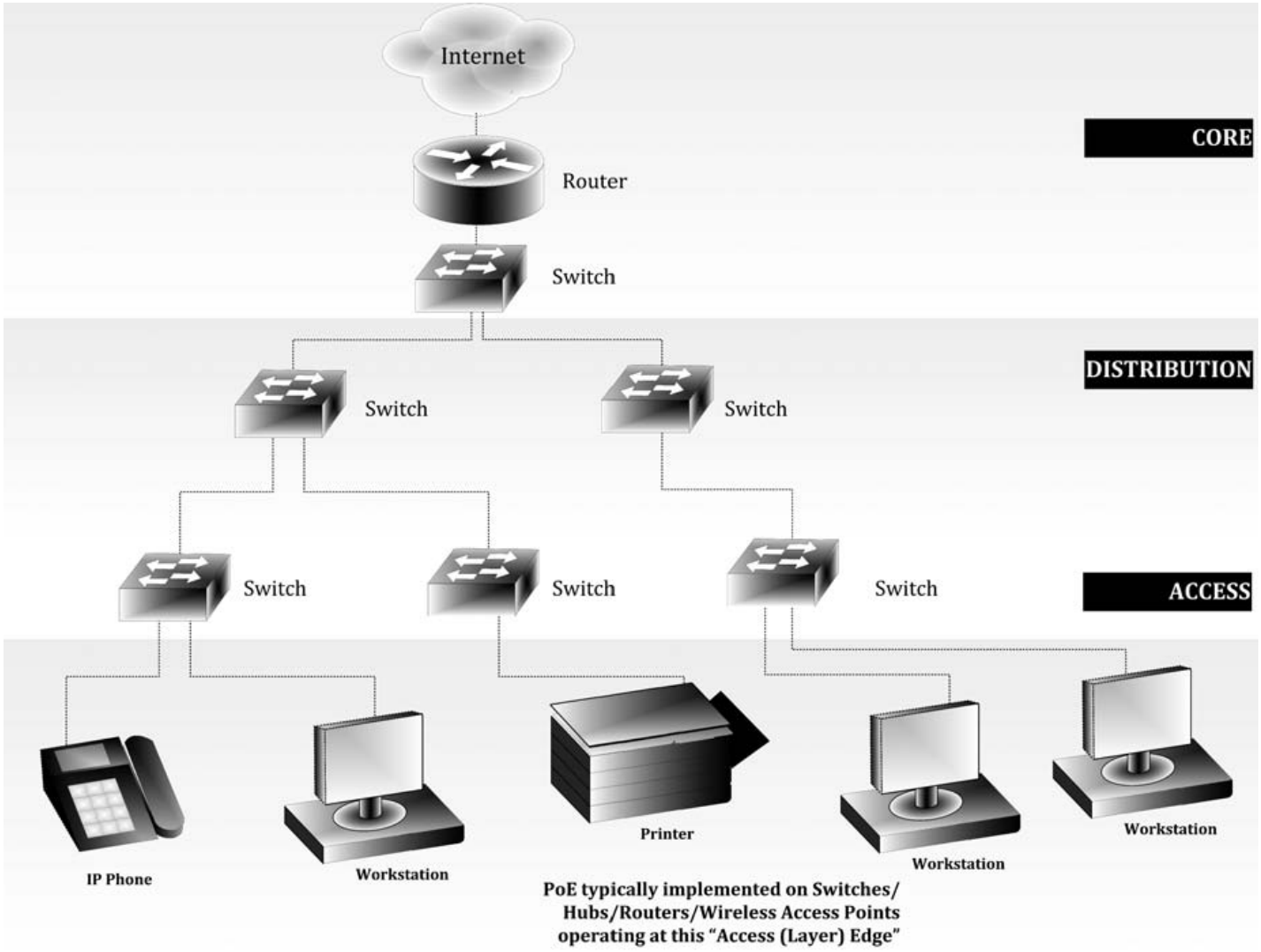


FIGURE 1.2 Hierarchical three-layer network architecture (and where PoE fits in).

But what did Metcalfe have to say about it in 1973? On May 22 of that same year, he wrote a world-changing E-mail within Xerox, Palo Alto, California—an E-mail which set the Ethernet ball rolling.

Did we just say "E-mail"? That's obviously not accurate. Because Metcalfe was *going* to enable E-mail *soon*, but till then "E-mail" (at least as we know it today) did not exist. So we just dodged a trick question.

It is more accurate to say something like this: "On May 22, 1973, Metcalfe hunched over an IBM Selectric typewriter using a spinning Orator ball, and talked about his vision of the future." In reality, his (almost) first sentence was "*I propose we stop calling this thing 'The Aloha network.'*"

The Aloha network (ALOHAnet) had been developed between the years 1968 to 1971 at the University of Hawaii. It was a radio-frequency link to connect the university facilities across different islands. Necessity is obviously the mother of invention. Metcalfe's system was an improvement over that, since it (eventually) detected and avoided collisions (his patent). But to make it clear to others that the system could support any computer, not just Alto (the Xerox workstation), Metcalfe chose to create a deliberately vague name based on the word "ether." In ancient times, people were not comfortable with the concept of a vacuum (complete nothingness). So "luminiferous-ether" was imagined to be the medium through which electromagnetic waves could propagate through space. In a similar fashion, Metcalfe envisaged a generic "physical medium" carrying bits of data to all stations (nodes in modern terminology). He explains that in the 1973 memo: "While we may end up using coaxial cable trees to carry our broadcast transmissions, it seems wise to talk in terms of an ether, rather than 'the cable,' for as long as possible. This will keep things general. And who knows what other media will prove better than cable for a broadcast network; maybe radio or telephone circuits, *or power wiring*, or frequency-multiplexed CATV, or microwave environments , or even combinations thereof." This book's author inserted the italics in the above statement.

Note very carefully that Metcalfe had already envisioned power and data sharing the same lines. But he was not the first as we will see.

Data over power cables, or power over data cables—what is the big difference?

Power-line carrier communication (PLC) has been around in a basic form since 1920s. A (modulated) wave of very low frequency was injected into high-tension power lines using coupling capacitors. It provided very basic, one-way communication/control. It was used for activating remote relays, public lighting, and so on. In the 1970s, Tokyo Electric Power Co. reported successful bidirectional operation to read and control power meters remotely. "Baby alarms" have been available as consumer products since 1940. The author too had built several pairs of "baby (monitoring) phones"

in the mid-1980s. These were small, short-distance, power-line carrier walkie-talkies, based on FM (frequency modulation using voltage-controlled oscillators) to transmit voice over home mains-AC wiring, followed by phase-locked loops inside the receivers, which were typically based on low-cost LM565/LM567 chips for decoding the modulation. Like walkie-talkies, both stations could not transmit at the same time (that would result in noise), and their best use was for monitoring purposes (one-way). In mid-1980s, research commenced into the use of existing electrical grids to support data transmissions using modulation of base frequencies up to 500 kHz. This was, however, still one-way communication. In 1997, the first tests for long-distance, bidirectional data transmissions over high-tension lines took place in Europe. Closer to our homes and times, today we have the most widely deployed power-line networking standard from HomePlug Powerline Alliance. We also have Broadband over power line (BPL), and so on. New devices from Netgear and others try to turn every AC outlet in our homes into a potential Ethernet jack. These devices comply with the IEEE draft P1901 standard and typically work up to 500 Mbps. Colloquially, this is often called Ethernet over power lines.

Thinking of other media as Metcalfe had imagined, Ethernet has now evolved into Ethernet over optical fiber too. For example, we now have the 1000 Mbps standard called 1000Base-F, where F stands for fiber. Of course we also have 1000Base-T and 1000Base-TX over twisted pair (copper). The general nomenclature being used (summarized as best as possible under the changing and evolving landscape) is presented in Fig. 1.3. In Fig. 1.4, we present an overview of communication standards, including non-Ethernet standards such as digital subscriber line (DSL), since, along with Ethernet, they remain a popular choice for data communication *over copper*.

A summary of the key Ethernet-over-twisted-pair (Base-T) standards that we will run into when designing PoE products is listed as follows (clearly PoE can't be used over fiber!):

1. 10Base-T: 10 Mbps (megabits per second) over 100 m of standard Ethernet cable consisting of four twisted pairs. Note that only *two* twisted pairs are used for data (data pairs). Two are just unused (spare pairs). Further, the data is conveyed unidirectionally on each of the two data pairs. That means one pair is dedicated to sending signals in one direction, while the other pair communicates in the opposite direction. It therefore is one-way on each pair and two-way on two pairs.
2. 100Base-TX: 100 Mbps (megabits per second) over 100 m of cable consisting of four twisted pairs. Once again, only *two* unidirectional twisted pairs are used for data. This is sometimes called Fast Ethernet and is the most prevalent Ethernet standard today.

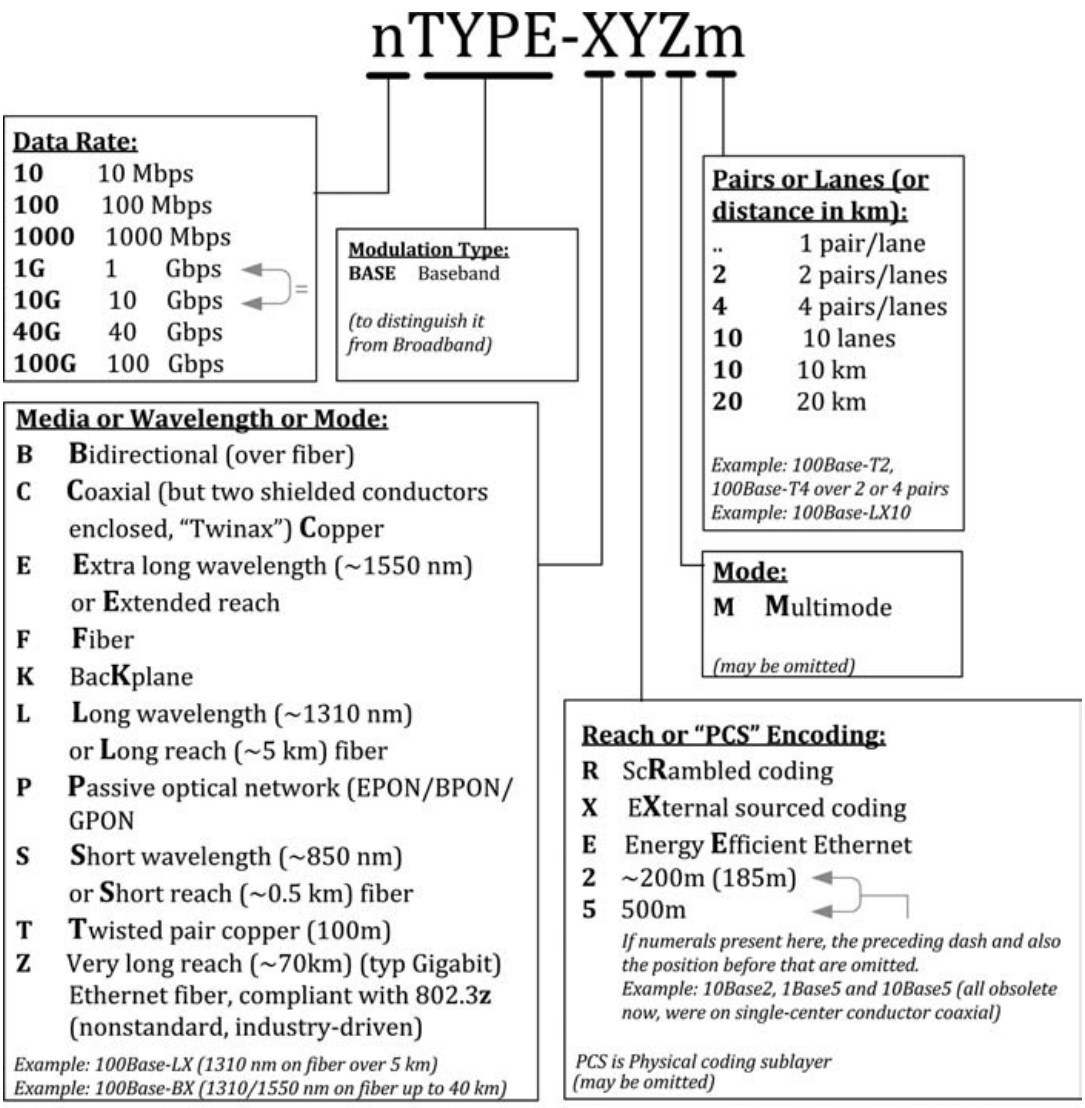
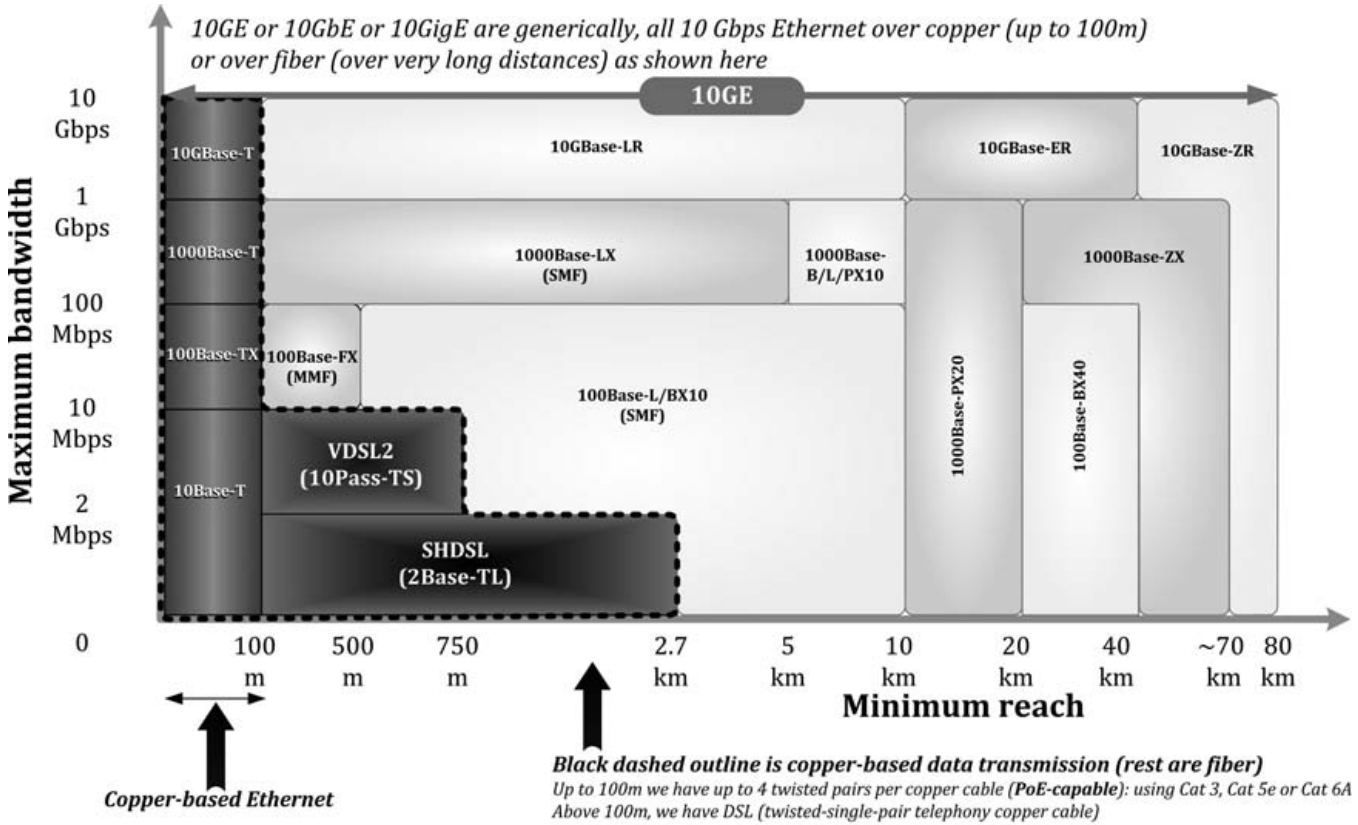


Figure 1.3 Ethernet nomenclature.

3. 1000Base-T: 1000 Mbps (megabits per second) over 100 m of cable consisting of four twisted pairs. Here all *four* twisted pairs are used for data. Further, each pair is bidirectional, which means two-way signaling on each pair, four pairs in parallel for higher speed. This is sometimes called 1 GBase-T or Gigabit Ethernet. The key advantage is that it can use the same cabling infrastructure as commonly used for 100Base-TX.
4. 1000Base-TX: 1000 Mbps (megabits per second) over 100 m of cable consisting of four twisted pairs. Theoretically, this was intended to save the cost of the electronics (the PHYs and so on), because though all *four* twisted pairs were to be used for



VDSL is Very high bit-rate digital subscriber line
SHDSL is Symmetric high-speed digital subscriber line

SMF is single mode fiber, MMF is mixed mode fiber. In the former, light follows a single path through the fiber while in the latter, it takes multiple paths resulting in differential mode delay (DMD). SMF is used for long distance communication and MMF for distances of less than 300 m. The advantage of MMF is that it can be driven by low cost lasers, whereas the advantage of SMF is that it can work over longer distances.

"-LX" are over a pair of SMFs
"-BX" are over an individual SMF
"-PX" are over PONs (passive optical networks)

FIGURE 1.4 Overview of Ethernet and related data transmission technologies.

data as in 1000Base-T, *each pair was to remain unidirectional*, as in 10Base-T and 100Base-TX. However, to compensate for the lowering of electronics capability, the cable data capability had to be correspondingly raised. In other words, 1000Base-TX requires more expensive cabling than 1000Base-T or 100Base-TX, which is just not easily available. Therefore, for all practical purposes, 1000Base-TX is now considered a commercial failure and effectively obsolete.

NOTE *The "T" at the end, as in 10Base-T, comes from twisted pair.*

NOTE *"Base" as in 10Base-T stands for* ***Base****band. A Baseband network is one that provides a single channel for communications across the* physical medium *(the common Ethernet cable in the case of copper transmissions) so only one device can transmit at a given time. Devices on a Baseband network are permitted to use all the available bandwidth for transmission (no sharing of bandwidth is necessary). The opposite of Baseband is Broadband. Broadband implements multiple channels, typically using frequency- or time-division multiplexing techniques. A typical example of a Broadband network is cable or satellite TV. Here bandwidth is shared.*

A list of key acronyms is provided in Table 1.1. One key acronym we run into all the time in PoE is PHY. In Ethernet, on one end of the cable we can have a digital-line driver, on the other end a digital receiver. In general, we could have digital transceivers (a combination of transmitter and receiver) at both ends. Generally speaking, these cable drivers/transceivers are referred to as PHYs, which literally stands for *phy*sical-layer drivers/transceivers. The physical layer in the case of Base-T Ethernet is simply the twisted-pair copper cable. In Base-F applications, the physical medium is the fiber-optic cable. In either case, the driver/transceiver is called the PHY.

What Is Interoperability?

Continuing Metcalfe's 2006 interview, he went on to say: "What the word Ethernet actually means today is six things… (1) It begins with a *de jure* standard made by a legitimate standards body, in this case the IEEE 802. (2) The implementations of that standard, painfully arrived at over years, are *owned by private companies*… (3) *Fierce competition* among the purveyors of the standard with their various implementations… (4) *Evolution* of the standard based on how things look after it

IEEE	Institute of Electrical and Electronics Engineers	EFMC	Ethernet in the First Mile over Copper
Cu	Copper	EFMF	Ethernet in the First Mile over Fiber
CO	Central Office	PoE	Power over Ethernet
LRE	Long-Range Ethernet (Cisco)	PoE+/PoEP	Power over Ethernet Plus
FTTH	Fiber to the Home	UPOE	Universal Power over Ethernet (Cisco)
FTTB/C	Fiber to the Building/Curb	PSE	Power-Sourcing Equipment
MDI	Medium-Dependent Interface	PD	Powered Device
PHY	Physical-Layer Device ("PHYceiver")		
PI	Physical-Interface (e.g., Cu)		
MII	Medium-Independent Interface		

Table 1.1 Key Acronyms to Keep in Mind

gets shipped, that is in the marketplace… . (5) Maximization of backward compatibility… . (6) an ethic in the competitive marketplace, *where it is not allowed to be incompatible.*

The last sentence points us to what we call interoperability today. The IEEE glossary defines this term as: *the ability of two or more systems or components to exchange information and to use the information that has been exchanged*. Wikipedia says it is the *ability of diverse systems and organizations to work together* (to "interoperate"). [Italics added by the author of this book].

For us, this basically means that equipment from Manufacturer A should "play well" with corresponding equipment from Manufacturer B, and also with Manufacturer C, and so on, because all this various equipment supposedly complies with the same governing standard. So provided the standard itself was carefully debated and formulated to start with, especially in terms of what is really crucial or important to overall performance, and hopefully is unambiguous to help reduce the possibility of mismatch (where it matters), then no interoperability issues should arise *in principle*. But the truth is there are lingering ambiguities in all standards. Also there are some subtle interpretation issues we need to consider very carefully. In this book we will attempt to show not only how to design good and reliable PoE equipment but also ensure they work and play well together. Hence the title of this book too.

NOTE *To put things in perspective, Metcalfe is also well-known for his prediction that the Internet would suffer a catastrophic collapse in 1996. He also promised to eat his words if it did not, and indeed he tried to when, in 1997, he took a printed copy of his column that had predicted the collapse, put it in a blender with some liquid and then consumed the pulpy mass. Metcalfe is also known for his harsh criticism of open-source software. In particular he had predicted that Linux would be finished after Microsoft released Windows 2000. He had said it was "utopian balderdash," and likened it to communism. He also predicted the end of wireless networking in the mid-1990s: "after the wireless mobile bubble bursts this year, we will get back to stringing fibers . . . bathrooms are still predominantly plumbed. For more or less the same reason, computers will stay wired." This is all available on Wikipedia.*

PART 2 THE HISTORICAL EVOLUTION OF PoE

Introduction

In Power over Ethernet (PoE), power and data are sent together down a standard Ethernet cable. The first formal PoE standard, IEEE 802.3af, was ratified in 2003, applicable to devices requiring up to 13 W. IEEE 802.3at followed in 2009, bringing into its fold higher-power devices, up to 25.5 W. The IEEE 802.3at standard actually contains two clear application categories. The first 13 W is as measured at the end of a 100-m cable, called Type-1, or "low-power." This was the same as in IEEE 802.3af. But it also introduced a new category for 25.5 W at the end of 100 m and called it Type-2 or "medium power." So, the "AT standard," as it is often colloquially called, is supposed to be just an "enhancement" of the previous AF standard, but it actually encompasses the previous standard and, in effect, *supersedes* it.

As we look back at the development of Ethernet in Part 1 of the chapter, and the advent of PoE, we can't but help feel all these events seem very recent, the underlying technology very modern. But as mentioned, the basic idea of sending information and power simultaneously over copper didn't even start with Metcalfe's 1973 memo, *it is actually almost two centuries old*.

It turns out that a surprising amount of ideas, tricks, and techniques that are in use today, not only in PoE, but in the general area of networking, can be traced back to a small group of incredibly resourceful engineers, scientists, innovators, and entrepreneurs, working against immense odds in what we perhaps consider a rather obscure moment in history. It is to this motley group that we owe many of our much-vaunted successes of today, and perhaps more to come. In contrast, the much-touted achievements of modern-day pioneers, many claiming a huge impact on mankind and society, pales into insignificance and borders self-promotion if not ignorance.

History can be not only entertaining and enlightening as a conversation topic over coffee, but very useful too. For example, not too long ago, two digital subscriber line (DSL) world-speed records were set in quick succession. DSL is a digital-transmission technology over existing telephone copper wiring, but it is different from Ethernet since raw data is not sent down the line; instead, *modulated* data is sent on a high-frequency carrier-sine wave (quite like a radio).

These DSL breakthroughs occurred just when the world seemed poised to conduct a perfunctory "let's get-it-over-with" funeral ceremony for DSL. The soothsayers were already starting to say: FTTH (optical *fiber to the home*), with speeds up to 100 megabits per second (Mbps), is the future, whereas DSL is a relic of the past. But all that changed suddenly over just a few months in 2010. In April, a DSL speed record of 300 Mbps was set (over 400 m of standard telephone wire) by the legendary Bell Labs (which is now part of Alcatel Lucent). That compared very well to the maximum prevailing DSL speeds of just around 10 Mbps typically (maximum 40 Mbps). Then in October of that year, equally unexpectedly, Nokia Siemens Networks announced a staggering 825 Mbps (over 400 m of telephone wire), bringing DSL close to the threshold of gigabit (1000 Mbps) over copper. In the process, something else also happened: copper had just become the "cockroach of telecom"—do what you like, you just can't make it go away.

Perceptive observers noticed something else in the twin DSL breakthrough announcements—a common underlying feature. Both companies had declared they had used something called *phantom DSL*. What exactly is that? Bell (Alcatel Lucent) elucidated further by admitting that they had exploited a 100-year-old networking trick. At first it seemed a little unusual to see such forthright candor and self-abnegation in our modern times. But it turns out they had every reason to be both candid and proud because that particular networking trick had also originated from an ex-Bell employee: named John Joseph ("J. J.") Carty, way back in 1886. In fact, we will soon learn that PoE is also based on the same phantom circuit principle. It is interesting to realize that this is really does make the very basis of PoE an offshoot of J. J. Carty's mind from way back in the 19th century.

We start to discover that nothing is as completely modern as we were hitherto inclined to believe. Also, both networking and PoE share a common heritage. Knowing that fact, we can hardly argue that a deeper knowledge of "past tricks" won't serve us well going into the future. That is why we too have chosen to take the historical path toward explaining PoE in this book.

Blasts from the Past

To many of us today, the 19th century swirls with names we've never heard of, and perhaps don't care to either. Emile Baudot, Claude Chappe, Cyrus West Field, William Thomson, John Joseph McCarty,

Oliver Heaviside, and so on, to name a few. Wait a minute: Doesn't the term "baud rate" sound very similar to the first name? Indeed, Baud rate did come from Baudot's work. Similarly, "modern" transmission line equations came from William Thomson and Oliver Heaviside over the period 1855 to 1885. William Thomson and Cyrus West Field were the pioneers behind the early transatlantic cables. Modern transmission-line equations were a direct result of their efforts to understand long-distance propagation of (telegraph) signals across these new "submarine" (underwater) cables. Incidentally, Thomson is also responsible for our "modern" temperature scale because of his discovery of absolute zero in 1848. And much more, in fact. It is therefore indeed surprising that most of us don't even have an inkling who Thomson was! But perhaps this rings a bell: William Thomson was subsequently knighted and became Sir William Thomson. A little later he took on the title *Lord Kelvin*. And *that* we may have heard of!

Not to forget Thomas Edison (1847–1931), with 1093 U.S. patents under his belt, considered the fourth-highest inventor in history. Especially in Edison's case, it was never a case of quantity over quality, or claiming innumerable "inventions" just to rake in the money from "incentive" corporate restricted stock units (RSUs). Incentive to cheat? Edison has to his credit the incandescent electric bulb, the phonograph, a motion picture camera, the first public, power-generation company, the electrical stock ticker, a quadruplex telegraph, and so on. We should, however, not forget that the key to Edison's fortunes was actually *telegraphy*. He learned the basics of electricity during years of working as a telegraph operator, and later he applied that knowledge to the telephone too. For example, the famed carbon transmitter (telephone mouthpiece) found in telephones, even until a few decades ago in many parts of the world, came from Edison. This author too remembers taking apart a standard Delhi City landline phone in mid-1970s to study its carbon microphone, complete with tightly packed carbon granules and all. That was Edison immortalized.

Engineers in that bygone era achieved a lot with almost nothing in their hands. Certainly they had no Internet to scour for information, much less to communicate with—no Wikipedia, Google, Facetime, Skype, Twitter, E-mails, IMs, nothing at all… horrifying as it may seem. They probably had to undertake long journeys by horse-drawn carriages or small boats just to arrive at the door of some eccentric, perhaps even suspicious, visionary or financier, hoping to generate fleeting interest in working together toward some vaguely defined mutual advantage. But these were still only relatively minor communication issues compared to the fact that both their hands were, technically speaking, tied firmly behind their backs. Think about it: What were their available resources at the time? In the 19th century, electricity had barely been harnessed, much less fully understood. Ohm's law

arrived in 1827, Kirchhoff's circuit laws in 1845. There were no vacuum tubes lying on rough-hewn work-tables, certainly no semiconductors, let alone 40-nm (nanometer) monolithic integrated circuits (ICs). There was barely an incandescent lamp in sight: Even the carbon-filament lamp (from Edison) came in 1879, the vacuum tube much later in 1906. Plastics had yet to be invented: The first plastic from a synthetic polymer was Bakelite, in 1907. Centralized electricity generation and distribution had just gotten off the ground—in 1881. What could modern-day greats like Henry Samueli (founder of an organization that proudly claims it is "connecting everything") have achieved under these circumstances? The truth is: probably zilch. No wonder we too instinctively start to think what could these poor 19th-century guys have done other than bow their heads and pray?

To our complete astonishment, on August 16, 1858, 18 years before even the invention of the most basic telephone, the first transcontinental message was being sent, not by horses or ships, but *telegraphically*, using electricity coursing through 2500 miles of copper lying deep under the Atlantic ocean. That momentous event, akin to landing a man on the moon in its time, set the stage for scenes of unprecedented jubilation and rejoicing across America and Europe. Alas, all for just a fleeting moment in time because this brand-new, very-first, transatlantic submarine cable failed in barely a month, but for reasons that can hardly be considered related to any fundamental design infirmity, technical oversight, or even ignorance. At least not ignorance on *both* sides of the cable. Many historians have concluded that the untimely demise of the cable was the handiwork of one person: Doctor Edward Wildman Whitehouse. A medical doctor by profession, he was the assigned engineer at one end of the long cable, with a theory of electrical propagation that can be best summarized in a few words (his) as follows: "the further that electricity has to travel, the larger the kick it needs to send it on its way." Banking on this little tidbit of knowledge, reportedly impervious to others around him, Whitehouse started zapping the cable using induction coils, with voltages of up to 2000 V—about four times larger than the cable was meant to carry. Thousands of miles away, stationed on the other side of the cable, *not* linked by a 3G network, Skype, telephone, or even a *telegraph* (the latter is what they were trying to barely get working at this point), Lord Kelvin reportedly had a chance to get through to Whitehouse, literally and figuratively—right untill the moment Whitehouse seems to have concluded with an impressive demonstration of a phenomenon we call "dielectric breakdown" today. Admittedly, there is no smoking-gun evidence in the form of a viral YouTube video, but Whitehouse is largely believed to have been the one to firmly kick the month-old transatlantic cable into the annals of history (or whatever constituted history way back then). However, to be fair to him, the person with the ultimate responsibility for the

debacle was arguably Cyrus West Field, the famed entrepreneur, who recruited Whitehouse in the first place. But as often happens today, Whitehouse was the (only) one who got fired. Cyrus got himself a second and third chance to succeed. And he did, rather spectacularly.

After a few years' delay on account of the Civil War, completely undaunted and undeterred by the previous cable failure, and despite having been thoroughly ostracized by neighbors and generally labeled a charlatan across the globe, Cyrus West Field, working with Thomson again (not Whitehouse this time), succeeded in laying not one, but *two* brand-new transatlantic cables, in 1866—in a procedure that is mind-boggling to read about even today. These new cables served their purpose for around a decade thereafter, and in doing so, they spurred a revolution in lifestyle that had hitherto never been seen before. That was truly a societal change. Wikipedia has a page dedicated to a 1998 book called *The Victorian Internet: The Remarkable Story of the Telegraph and the Nineteenth Century's On-Line Pioneers* by Tom Standage. The book reveals some of the astonishing similarities in the rise of the 19th-century telegraph and the rise of the Internet in the late-20th century. The central idea of the book, Wikipedia points out, is that of these two technologies, it is the telegraph that is the more significant, because the ability to communicate globally in real-time was a *qualitative* shift at the time, while the change brought on by modern Internet is merely a *quantitative* shift. Roll over Samueli.

Whether we agree with that viewpoint or not, a historical perspective invariably creates a very interesting entry point into the heart of what we instinctively consider to be modern technologies.

Don't SWER No More

A very, very long time ago, telegraph systems were based on just *one* copper wire laid down over several miles. Metal poles buried in the ground on both sides completed the return path of the current (through moist subsoil, water, sea, or even ocean). This is called for *single-wire earth return* (SWER). This single-conductor principle was used extensively in power distribution systems even later, and is still considered an effective and economical choice for rural electrification in remote and backward locations. The same single-conductor principle is also often used today for modern light-rail systems, remote water pumps, and so on.

Unfortunately, completing a return path through (earth) ground creates a current loop with a huge arbitrary, almost *undefined*, and possibly varying enclosed area. This makes the entire system susceptible to picking up extraneous disturbance and noise (of the electromagnetic variety)—it is a big antenna, courtesy of Ampere's circuital law combining forces with Lenz's/Faraday's law of induction. In modern

parlance, we always need to ensure our systems have adequate *electromagnetic immunity*.

The terms *immunity* and *susceptibility* are used equivalently to describe the ability of equipment to function acceptably in a typical electromagnetic environment. But why did the vague boundaries of SWER *not* pose much of a problem with the telegraph? Because telegraphy is essentially *digital* in nature! Yes, digital was there long before analog. The dots and dashes can be thought of as a string of ones and zeros. We know all too well today that digital systems are inherently more noise-resistant than analog systems—same as in the 19th century. Therefore, telegraph systems worked quite well within the rather vague physical boundaries of SWER. Unfortunately, the inadequacies of this single-ended architecture were thoroughly exposed when analog (voice) signals were attempted to be transmitted over the existing telegraph-wiring infrastructure following the invention of the telephone in 1876. Plagued with strange noises, the solution emerged in quite quickly too, in the form of the return copper wire, proposed in 1881 by the very same J. J. Carty mentioned previously. And that same year, Alexander Graham Bell, the man behind the telephone (or the talking telegraph as it was initially called), filed a patent for the *twisted* return wire—which is basically what we call *unshielded twisted pair* (UTP) today. From Part 1 of this chapter we remember that is what drove Ethernet into all-time popularity. In one word: *cost*. UTP still happens to be the most cost-effective, most common type of Ethernet (and telephone) cable in use today.

Some may argue that Ethernet is *digital* too, so why can't we still use SWER? There are several reasons for that:

1. In our modern world of ever-decreasing voltage sources, we now have digital thresholds that are very close together compared to the higher-telegraph voltages, so noise immunity is not so good either.
2. At the high data-transfer speeds we are talking about, we can no longer afford too many errors caused by noise.
3. By using the ground for high-frequency data, we would create a huge amount of electromagnetic radiation that would impact neighboring (sensitive) equipment. This is discussed in Chap. 2.

Historically, compared to SWER, the return-wire concept (metallic circuit), when proposed, seemed to imply *double the cost* of copper, so the idea was obviously met with some high initial (human) resistance. But it was also quickly apparent that a copper return wire was simply unavoidable for ensuring acceptable performance in telephony. However, there was another major breakthrough toward the end of the 19th century, in the form of Pupin (loading) coils that helped greatly. These are coils of very large inductance inserted every

few thousand feet (typically 88 mH every 6000 feet) over the entire length of a long telephone cable. The discrete inductors couple electrically with the existing distributed cable capacitance, creating LC-type transmission-line effects, similar to what we rely on in modern high-frequency data transmissions, but now effective at very low (audio) frequencies. This "pupinization" of telephone wiring, as it was called, allowed voice frequencies to travel much greater distances—an alternative to blindly increasing the thickness of copper just to lower the DC resistance for achieving comparable propagation distances. Pupinization is said to have saved up to 75 percent of the projected copper costs associated with telephone cabling.

All this is just a fascinating example of the prolific ideas swirling and accruing rapidly in the 19th century. Riding on such clever breakthroughs, by the close of the century, there was an almost complete conversion from grounded circuits (SWER) to metallic circuits (those with copper returns). And with that, the *twisted pair* rose to supremacy. As indicated previously, not only does Ethernet uses it today—so does DSL.

The Twisted Pair and the Principle of Immunity

The basic principle behind the twisted pair and its various implementations is shown in Fig. 1.5. At the very top of the figure is an analog signal being transmitted from the microphone of a traditional telephone to the loudspeaker on the other side.

NOTE *We are ignoring another clever technique here for the time being, by which we combine, and then later separate, the loudspeaker signal from the microphone (transmitter) signal, over a single twisted pair—this involves an innovation called the "hybrid transformer." It is discussed in more detail in Chap. 13.*

In the cases that follow in Fig. 1.5, the signals are digital, but the underlying principle is the same. We see the noise spikes (small triangles) riding *equally* on both constituent wires of the twisted pair (same amplitude and same direction/polarity). The rationale behind that is that since the wires are twisted uniformly, they get exposed *equally* to disturbances—without any preference to either wire. Otherwise, we could well ask why the noise pickup is *different* on one of the two wires if nothing distinguishes one from the other. In other words, just by plain symmetry, the two wires of the twisted pair must have identical noise pickups. In modern terminology, the disturbance/pickup is called *common-mode*. We can, however, ask in a common-mode case: Is the noise voltage identical on both wires *relative* to what exact potential? The answer to that is the noise spikes are identical with respect to (earth) ground. And that is what we mean by

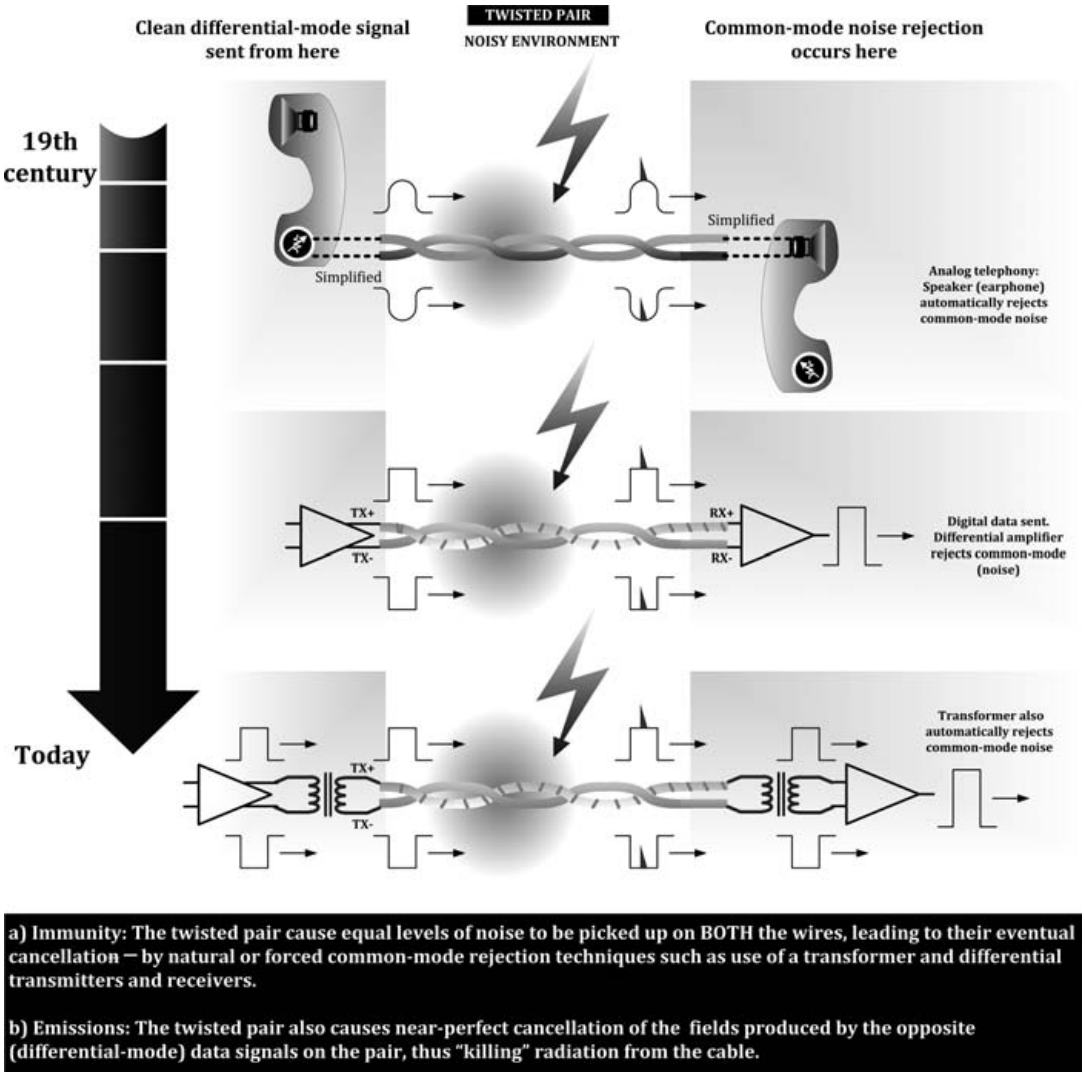


FIGURE 1.5 How the twisted pair, along with a differential amplifier and a transformer, helps reduce electromagnetic noise pickup.

common-mode. We can alternatively say that there is no voltage difference between the two wires because of noise spikes, and so the noise pickup is *not differential-mode*.

Differential-mode implies the opposite of common-mode: At any instant, common-mode consists of two *equal* signals with the *same* polarity, whereas differential-mode consists of two *equal* signals of *opposite* polarity, and in both cases the referred-to voltages are with reference to earth-ground potential. In Fig. 1.6 we see more clearly what exactly are differential-mode (DM) and common-mode (CM) currents. We have marked the noise source generically as N inside a circle. We also see how in a general case of mixed-mode (MM) currents, we can split the currents into their DM and CM components. Keep in mind the sign logic we are using in the numerical

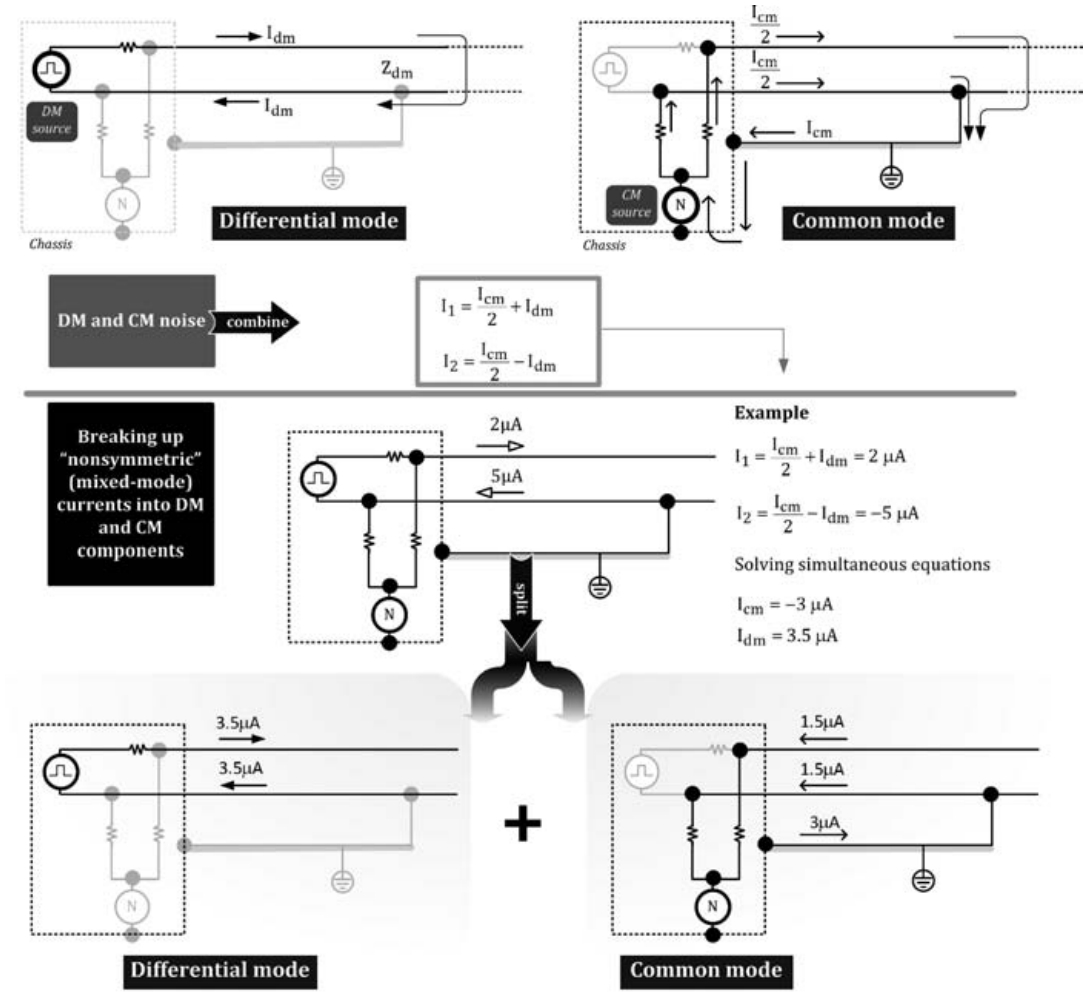


Figure 1.6 DM and CM currents (top), and splitting mixed-mode (MM) currents into constituent DM and CM components.

example: Any current from left to right is positive, and from right to left is negative.

Noise pickup is common-mode (with a negligible differential-mode component), only provided the wires are twisted tightly together. If not, we can certainly get unintentional asymmetry, which will lead to a small unintentional differential-mode noise component (as indicated by the numerical example in Fig. 1.6). But why is that so scary anyway? The problem with that scenario is the actual signal is (by design) that transmitted down the wire in a *differential* fashion (explained further below). If the noise has a differential component, it will end up interfering with the actual (useful) signal. In that case, we could well ask how any circuit would "know" what constitutes signal and what constitutes noise? In a good setup, signal and noise are distinguishable (and separable) only

because one is purely differential-mode (the signal), while the other is purely common-mode (the noise). In other words, noise and signal are made to reside in separate and distinguishable domains using special techniques. Then, with appropriate circuitry, we accept one of them (signal) and reject the other (noise).

Common-Mode Rejection by Coils/Transformers and Other Techniques

Besides the most basic requirement of a twisted pair, what special techniques were we discussing previously ? Let's list some of them here.

1. We start by revealing the simplest technique to separate signal from noise, applicable to both analog (telephony) and digital (Ethernet). Visualize the following situation: If both ends of a magnetic coil, such as in the loudspeaker of a telephone (the first schematic in Fig. 1.5), or one of the windings of say, a data transformer (the last schematic in Fig. 1.5), are raised or lowered in unison by *exactly the same amount* (and that by definition is common-mode), no corresponding current will be produced in the coil. Why? Because current only flows if there is a voltage *difference* (delta) present, and in this case we have no voltage difference *across* the ends of the coil (the voltages at its two ends are changing in unison by equal amounts and with the same polarity). In other words, only a *differential*-mode signal applied across the ends of a coil/winding will produce a delta V with a resultant current flow. Common-mode will do nothing here. That is common-mode rejection, by definition.

   Alternatively, if the noise picked up is purely common-mode, as is true for the twisted pair in Fig. 1.5, the loudspeaker will not emit any sound corresponding to the noise. Only the voice signal, which is applied differentially across the twisted pair by the microphone on the other side of the cable, will be heard through the loudspeaker.

   A very similar situation arises in a transformer. If no current flows through a winding on one side, no voltage or current can appear on its other side—because transformer action requires that a (time-varying) current flow through one winding, creating an induced voltage across the isolation barrier on the other winding. In other words, both coils and transformers have inherent common-mode rejection properties. This property is commonly used in Ethernet today, as it was in analog telephony over a hundred years ago. In telephony, voice-frequency transformers were typically placed at certain key positions, such as in telephone exchanges.

They were called repeating coils at the time because their main application was to inductively transfer (or repeat) the signal from one telephone circuit (branch) into another. But the actual signal came through clearly to the other side *minus* (a good deal of) noise. So they were also used for common-mode noise rejection. Some people express this property of a transformer in a slightly different manner by saying "repeating coils (isolation transformers) break up ground (earth) loops," (and that leads to cleaner signal transmissions, with no funny, buzzing sounds in telephony, and so on).

2. Ground loops are nothing but a path for common-mode noise/signals to flow. So breaking up a ground loop, however we do it, is tantamount to enforcing common-mode rejection. One way to do that is by using data transformers as explained previously. But it is also obvious we should *avoid making any direct galvanic connection* to earth (ground) on the line (cable) side. We will learn that PoE stages, which are always located on the line side, are for that reason *never* connected directly to earth (ground). Blocking capacitors of very small capacitance, called Y-capacitors, are the only link from PoE (line) side to earth (ground) (some amount of capacitance to ground is deemed necessary for overall EMI-suppression purposes). On the other side of the data transformer (the driver side), there is however almost invariably a direct physical connection to earth (ground) (via the AC wiring) for safety reasons (to protect the user from electric shocks). We will discuss the safety and isolation aspects in greater detail in Chap. 10.

3. Let's briefly summarize here: SWER systems *depend* on ground loops to work. We tried to eliminate that ground loop by the use of the twisted pair (metallic circuit). The data transformer helped further in that mission because it enforced a break in any inadvertent ground loop. In addition, we learned that we should avoid physical connections to earth (ground) and instead use Y-capacitors to connect anything on the line (cable) side to earth, (ground). In other words, we need to isolate the line and line-side circuitry from earth (ground).

   On top of this, capacitor injection is another technique that can be used to inject a signal into the twisted pair (instead of using drive transformers). In that position, the capacitors will block DC and thereby help break up any ground loops that may form through the line driver side. Unfortunately, a capacitor is a high-frequency bypass, because the impedance of a capacitor is $1/(2\pi f \times C)$ and if f and/or C is large, the impedance is very low. In other words, capacitors do block DC, and

break up DC-ground loops, but they also permit high-frequency or AC-ground loops to continue to exist. In contrast, a drive transformer is much better at common-mode rejection and breaking up of ground loops. So a drive transformer is the preferred choice in Ethernet. Capacitor coupling is not as effective.

Note that, theoretically, the twisted pair can be directly driven as shown in the middle of Fig. 1.5. But it is now obvious that is not a good idea for breaking up ground loops, so that is certainly one reason it is never used. But we should note that there is another subtle reason to avoid direct drive too: Under a fault condition, a direct-drive line driver (transceiver) can be easily damaged. Capacitor coupling or transformer coupling, on the other hand, are relatively fail-safe since they both end up blocking any DC, which will likely result in the case of most common types of fault conditions.

4. All these techniques—twisted pairs, data transformers/coils, and so on—are part of our growing war-chest of tricks for separating noise and signal, thereby ensuring "signal integrity" over long distances. What other techniques can we use? As indicated in the schematics of Fig. 1.5, the most obvious way of rejecting common-mode noise is to use *differential* stages, both for transmitting the signal at one end (differential driver), and for receiving it at the other (differential amplifier).

However, just so we do not lose track of the bigger picture here, we need to emphasize once again that in all the schematics of Fig. 1.5, actually depend on noise being picked up identically (common-mode) on both wires. That is the key advantage of a twisted pair. So, the twisted pair is a basic requirement. Combined with the repertoire of related techniques, such as described previously, we then continue to restrict noise to the common-mode domain and the (useful) signal to the differential-mode domain. Eventually, that is what makes noise and signal distinguishable, ultimately filterable and separable.

Immunity and Emissions

We have been mentioning "immunity" in previous sections without having spelled it out very clearly so far. We also referred to "a typical electromagnetic environment." What do these terms mean and relate to?

Electromagnetic immunity/susceptibility (EMS) is one side of the total coin called *electromagnetic compatibility* (EMC). See the left side of Fig. 1.7. On the other side of the EMC coin lies electromagnetic emission (EME). For example, an intentional/unintentional electromagnetic emitter (Device A) sends out electromagnetic interference (EMI)

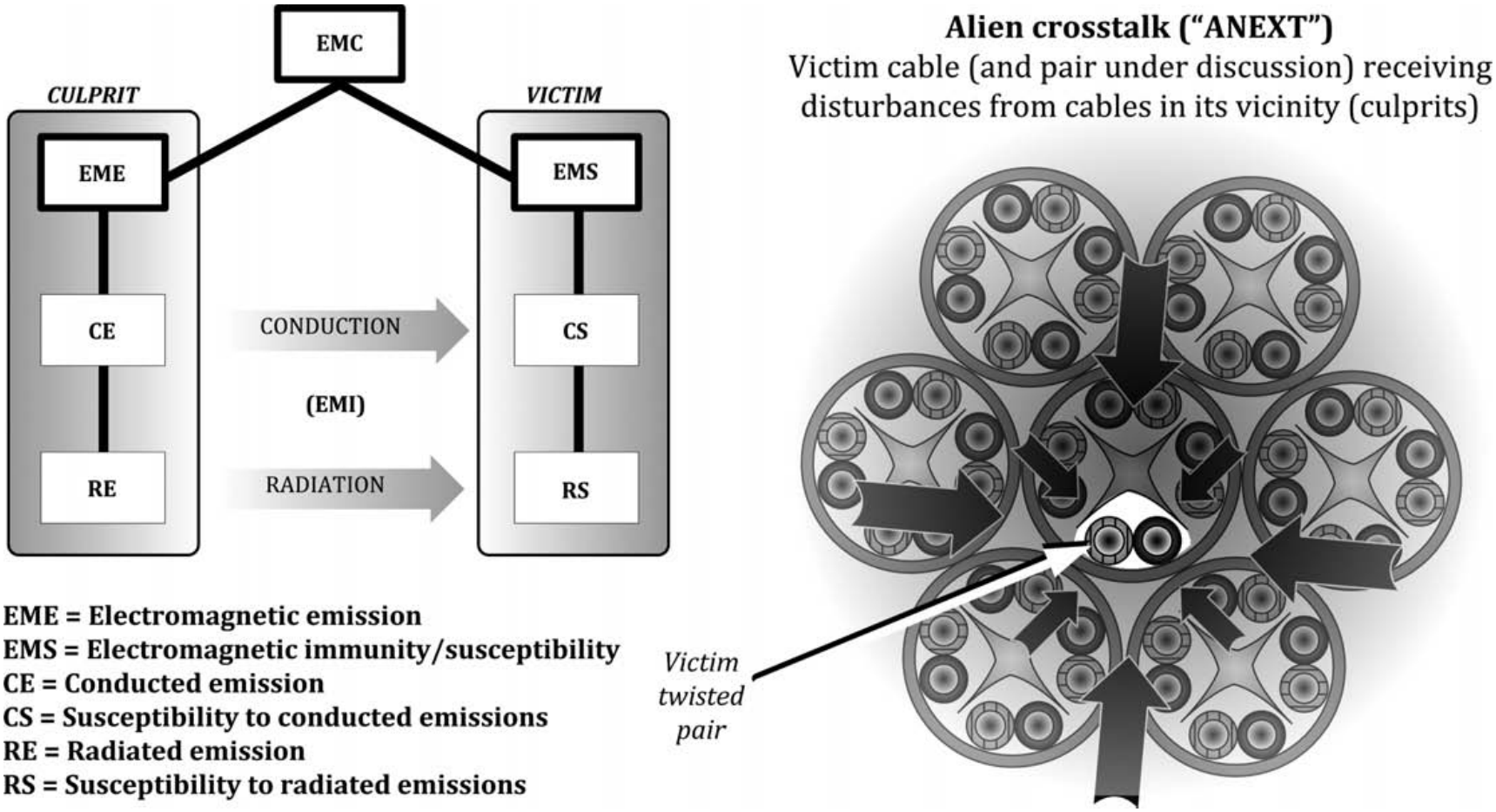


FIGURE 1.7 Concept of EMC and cross talk in cables.

all around it. Another device (Device B) in its immediate vicinity should not only continue to work well when faced with this impinging EMI (provided, of course, the levels of that are not excessive), but must itself not emit significant amounts of EMI, so as to allow other devices in its vicinity, such as Device A, to function acceptably too. These are just basic good-neighbor principles at work—within the EM environment.

To clearly define and regulate both aspects of EMC, there are well-known European Norms (EN) in Europe and Federal Communication Commission (FCC) standards in United States, but detailed discussions about regulatory EMC regulations are out of our scope here because that hardly concerns PoE, which can be considered largely *DC-based* (and we know DC does not radiate). The important thing to remember for our humble purpose here is that, in general, a good RF antenna is not only a good *receiver* of EMI, but a good *transmitter* too. Similarly, a bad receiver of EMI is a bad emitter of EMI too. So, for example, we know that a long, single-conductor wire is a good antenna, both for transmission and reception. However, perhaps rather nonintuitively to us at first, a long twisted-pair cable is a relatively bad antenna, or at least not as good an antenna as we may have expected, based on its good length. But at least one thing remains true and consistent (though it is almost coincidental in this case): The twisted pair cable, as used in an Ethernet environment, happens to be a bad antenna for both reception and transmission (of EMI). We will explain the reasons for all this below.

When used in Ethernet applications, the twisted pair not only rejects incoming EMI (in effect, it provides system immunity), but does not radiate too much itself (so it doesn't test the immunity of neighboring devices too severely either). The reason for the low emissions comes from the fact that the magnetic fields produced by each wire of the twisted pair, when driven with purely *differential* signaling, are in opposite directions with equal magnitudes at any given instant. So they mutually cancel each other out—there is no *net* (resultant) field, at least not in theory. But yes, if the differential-mode signal has an inadvertent *common-mode* component to it, not due to noise this time, but from design-related issues, (such as inherent imperfections in the differential nonideal driver), the cable will end up radiating somewhat. Similarly, if the noise pickup is purely common-mode, it does get rejected very well as discussed previously, and that gives us immunity. But if the noise has a small differential-mode component (e.g., caused by poor twisting in a certain area of the cable, such as at a very sharp bend), the system *will* see some noise getting mixed in with the signal, and that will, in effect, lower, the overall system immunity. In other words, *signal integrity* will be compromised.

On the right side of Fig. 1.7, we see at the cross section of four twisted pairs of a typical Ethernet cable (viewed from the top). With four twisted pairs in every cable, and several cables in a bundle (as the cables go out from their hub/switch to the workstations), there can be

significant "pickup" via radiation between *adjacent* twisted pairs. This is called cross talk. In effect, it degrades signal integrity, affects data transmission capability, and eventually reduces its reach (distance). So for all the reasons described above, the twisted pair will help significantly reduce cross talk too, since interference from adjacent pairs, in the same cable or from adjoining cables, is basically noise pickup from the viewpoint of any given twisted pair under study.

NOTE *When the disturbance is from pairs in surrounding cables, the word for that is alien cross talk (ANEXT or AXT). We also have near-end cross talk (NEXT), and far-end cross talk (FEXT), which refer to the cross talk from the other three pairs of the same cable as the victim. NEXT occurs when a receiver overhears a signal being sent by a transmitter positioned at the same end of the cable as the receiver, whereas FEXT occurs when the overhead transmitter is located at the opposite end of the cable, away from the receiver.*

NOTE *In Fig. 1.7, the victim cable has been shown with exactly six disturber cables surrounding it. We realize though that in a typical Ethernet cable bundle there may be many more cables surrounding any given cable. However, as we can see, by sheer geometry, six cables will completely surround a given cable with no intervening gaps, so they effectively shield the victim from the effects of outer cables. Therefore, for studying ANEXT, the standard setup is as displayed—with exactly six cables surrounding the cable containing the victim pair. This is often called the 6-around-1 configuration.*

We thus see that the magical unshielded twisted-pair cable (UTP) helps achieve *both* immunity and low EMI (with the help of all the supporting techniques as explained previously). It turns out that UTP is not only low-cost, but high-performance too. It's like a free lunch, in effect. And that's why it contributed significantly to the explosion of Ethernet, compared to rival LAN proposals. Note that there is no need for any separate external shielding either. A shield may only complicate matters by providing an alternative and ambiguous return path for the CM and DM currents. So it is no surprise that both the coaxial cable and the shielded twisted-pair cable (STP) are almost dead (for this purpose). In contrast, UTP abounds all around us.

Twist Rate and Wire Diameter

An extremely important characteristic of a twisted pair is related to the basic question: How twisted is it? Cable categories have been defined in the standards, and these eventually relate directly to a certain twist rate, or twists per unit distance (distance is measured in inches or feet, for example). We do not need to go into too much

networking detail here, but it is good to keep in mind that the greater the twisting, the better the performance of the cable in general.

For PoE, the twist rate is not of any *direct* concern, except perhaps in some esoteric system-design matters as discussed later. Because, in PoE, we are essentially concerned only with the DC resistance of the cables, not its reactive parasitic elements, like inductance and capacitance per unit length. However, in what may be considered a fortuitous coincidence, "good" cables from the viewpoint of *data*, are usually good for PoE too. Not because of the higher twist rate (rather, *despite it,* as discussed below), but because "good" cables are typically made of *thicker* wire. Greater wire thicknesses imply not only lower-DC resistance, but lower-AC resistance too. That helps both data and PoE. A thicker wire increases the useful signal received at the end of a 100–m cable, because it reduces Insertion Loss (reduces attenuation). For almost the same reason, in the case of PoE, a thicker wire allows for more power to be delivered at the end of the cable, because we have lower I^2R losses in the cable. In other words, in most cases, power and data capabilities of cables seem to go hand in hand: They end up dovetailing, much to our design satisfaction and ease.

We may notice, while playing around in the lab, that some of the twisted pairs of a typical Ethernet cable are easier to unravel. There is a good reason for that. If adjacent pairs have an exactly identical twist rate (pitch), we could end up with a situation in which wires of different pairs fall coincidentally almost adjacent to each other for the *entire* cable run, affecting differential signaling negatively, and increasing cross talk. To prevent this, Ethernet cable manufacturers use different twist rates for different pairs in a good cable, though all this is not usually declared or apparent to the user.

Unfortunately, a high-twist rate leads to a *longer* (unraveled) length of copper wire. This not only increases the AC and DC resistance somewhat, but also increases the propagation delay, which is the time taken by the signal to travel across the cable. Luckily, propagation delay by itself is usually of less concern than the *differences* in the propagation delays of adjacent pairs of a cable, which is called delay skew. By using differing twist rates on different pairs of a cable to reduce cross talk, we end up with larger delay skews. And that can become of serious concern, especially in high-definition video applications and/or very high-speed data transmissions. But twisting in general, despite this relatively minor disadvantage, has overwhelming advantages.

Categories of Ethernet Cable

In principle, we can implement Ethernet technology not only over unshielded twisted pair (UTP), but also shielded twisted pair (STP), coaxial wire, or even optical fiber. However, in this book we are going to focus only on the ubiquitous UTP, since for most

applications, UTP happens to be the most cost-effective, popular, and prevalent choice. In this section we will list the key cable categories that make them either suitable or unsuitable for Ethernet applications. We have also summarized key applications versus cable categories in Table 1.2.

The Telecommunications Industry Association (TIA) categorizes cables depending on their data transmission capabilities (over 100 m of cable). The TIA is largely North American. In Europe, the corresponding standard is from the International Standards Organization (ISO). In Europe, the cable categories/components are called by different names, as we can see in Table 1.2. But they are U.S.-equivalent categories. Keep in mind that in Europe, telephone/Ethernet/AC color coding can all be quite different from the United States too.

The older U.S. Ethernet cable standard was TIA-568A, the more modern one is TIA-568B. These standards are the origin of the prefix "CAT" or "Category" that we will find on a typical (North American) Ethernet cable, expressing its rating and capability. The most common cable category in use until a few years ago was Category 3 (CAT3), which is considered good for 10Base-T Ethernet, or basic telephony applications. For 100Base-TX, the most common cable around is Category 5e (or CAT5e, in which "e" stands for *enhanced*).

What are the associated wire diameters? We note that TIA-cable standards are inherently pre-PoE, or datacentric. The good news is that since data and power capabilities do seem to dovetail, we can deduce the worst-case wire thickness (AWG) required for PoE power calculations. It eventually leads to

1. CAT3 (Class B) (16 MHz/16 Mbps): Typically AWG26 (worst-case) to AWG24. Used primarily for 10Base-T. The first PoE standard, IEEE 802.3af-2003, was written with this category in mind.
2. CAT5 (100 MHz): Typically AWG26 to AWG22. Rare or obsolete. Ignore.
3. CAT5e (Class D) (guaranteed 100MHz; typically up to 350 MHz): Typically AWG24 (worst-case) to AWG22. "e" stands for enhanced, which implies a higher twist rate and lower cross talk than CAT5. Used primarily for 100Base-TX, but can usually also support 1000 Mbps over 100 m by using all four pairs. The second (most recent) PoE standard, IEEE 802.3at-2009, was written with this category in mind.
4. CAT6 (Class E) (250 MHz): Typically AWG23 (worst-case) to AWG22. It is rarely used, since it was intended for 1000Base-TX, which is dead as discussed earlier. It also falls short of supporting 10 G (10,000 Mbps applications) over the full 100 m as required.

TIA and ISO EQUIVALENTS								
Frequency Bandwidth (MHz)	**TIA Components**	**TIA Cabling**	**ISO Components**	**ISO Cabling**				
16	Cat 3	Cat 3	Cat 3	Class B				
100	Cat 5	Cat 5	N.A.	N.A.				
100 (Typ 350)	Cat 5e	Cat 5e	Cat 5e	Class D				
250	Cat 6	Cat 6	Cat 6	Class E				
500	Cat 6A	Cat 6A	Cat 6A	Class E_A				
600	N.A.	N.A.	Cat 7	Class F				
1000	N.A.	N.A.	Cat 7_A	Class F_A				
APPLICATION CHART								
	Direction	**Cat 3**	**Cat 5**	**Cat 5e/ Class D**	**Cat 6 / Class E**	**Cat 6A / Class E_A**	**Class F**	**Class F_A**
Telephony (separate analog signals on each pair)	↔ ↔ ↔ ↔	•	•	•	•	•	•	•
10Base-T	→ ← — —	•	•	•	•	•	•	•

TABLE 1.2 Summary of Cable Categories and Applications

	Direction	Cat 3	Cat 5	Cat 5e/ Class D	Cat 6 / Class E	Cat 6A / Class EA	Class F	Class FA
100Base-T4	→ ← ↔ ↔	•	•	•	•	•	•	•
100Base-TX	→ ← — —		•	•	•	•	•	•
1000Base-T	↔ ↔ ↔ ↔			•	•	•	•	•
1000Base-TX	→ ← → ←				•	•	•	•
10GBase-T	→ ← → ←					•	•	•
Broadband CATV (with Ethernet on same cable)	→ — — —						•	•

TABLE 1.2 Summary of Cable Categories and Applications (*Continued*)

5. CAT6A (Class E_A) (500 MHz): Typically AWG23 (worst-case) to AWG22. This is a future specification, intended for 10 G applications. It is becoming increasingly popular in an attempt to "future-proof" new installations.

Why is CAT5/CAT5e so much better than CAT3 anyway? The minimum wire gauge is better for one. We can also intuitively understand that another key reason is the twist rate. Typically, CAT3 has three twists per foot, whereas CAT5/5e has about 2 to 3 twists per inch (10 to 12 times more than CAT3). In CAT5e has lower cross talk than CAT5. One way to reduce cross talk significantly is to use dissimilar twist rates in the pairs of a given cable.

PoE Cable Categories

From a PoE perspective we need to remember this:

1. IEEE 802.3af-2003 assumes a worst-case of AWG26 (CAT3).
2. IEEE 802.3at-2009 assumes a worst-case of AWG24 for higher-power applications (CAT5e).

(Keep in mind that AWG24 is thicker than AWG26.)

In terms of resistances:

1. IEEE 802.3af assumes that a 100 m CAT3 cable that has a worst-case (DC) loop resistance of 20 Ω. This is the cable resistance assumed for low-power applications (13 W at the end of the cable).
2. IEEE 802.3at assumes that a 100 m CAT5e cable that has a worst-case loop resistance of 12.5 Ω (for Type 2 medium power applications). This is the cable resistance assumed for medium-power (Type 2) applications (25.5 W at the end of the cable).

We will do some calculations later, based on the resistivity of copper. At this point the above information is enough, but we may also want to keep in mind that these resistance numbers are actually for 90 m of Ethernet cable plus a total length of 10 m of patch cables at either end. It also includes estimated contact resistances of connectors on both sides. Temperature variations are also included in these resistance numbers.

Bandwidth and Information Capacity of Cables

We may have noticed from Table 1.2 that CAT5e can support 1000Base-T (1 Gbps), even though it is only rated 100 MHz. We realize that we are using all four pairs of the 100 MHz cable for doing 1 Gbps, but we still

can't seem to explain this rather big jump to 1 Gbps. There seems to be no obvious math here. And does that mean 100 MHz is really *not* equivalent to 100 Mbps as often assumed? Yes, *there is really no obvious relationship between bandwidth and maximum data rates.*

Historically, especially when used for RF purposes, the usable bandwidth (maximum frequency range) of a cable was supposedly related to the relative attenuation of different sine-wave frequencies as they passed through the cable. For example, we have for years used coaxial cable (RG-6) for cable TV (CATV), in which many stations are carried simultaneously up to very high frequencies (~1000 MHz). Also, the length of the cable really does not seem to profoundly affect its frequency characteristics; the length actually seems related more to the attenuation of the entire signal over very long cable lengths, and the sensitivity/design of the RF preamplifier/receiver to extract a "clean" signal from the noise. Yes, we do know today that the diameter of the cable is a key factor in determining its frequency characteristics (cutoff frequency).

What Metcalfe proposed in 1973 was a very different application of coaxial cable. First, it was now being used *not* for analog sine waves but for *digital* signals, with sharp "edges" containing a lot of high-frequency harmonic content. Second, it was being *shared* for data. So the final point-to-point data rates would be affected by the number of computers hanging off the bus. Clearly the concept of bandwidth and "information capacity" was evolving and developing.

Let us fast-forward to modern times where we have a star topology (no shared bus), and we are using twisted pairs, not coaxial, because that is what is relevant to us today. One of the most important and basic parameters that defines the final performance of telecommunications cabling is its *channel bandwidth*. This is the key differentiator between what we call CAT3 and what we call CAT5e, for example. The channel bandwidth is the frequency range over which the signal-to-noise ratio (SNR) is a positive quantity when expressed in decibels (dB); which basically just means the signal level is greater than the noise level. SNR is basically the same as the (power sum) attenuation-to-cross talk ratio (called PSACR or just ACR). For example, for a CAT5/5e channel, the objective is to have a PSACR greater than zero (a positive number in decibels) over a frequency range up to 100 MHz. That is, by definition, bandwidth. Note that in all cases, we are assuming 100 m cable length in Ethernet applications.

Coming to the information capacity of cables, people often equate 10 MHz bandwidth to 10 Mbps, 16 MHz to 16 Mbps, and so on. This, in fact may be true, but only *coincidentally* so. For example, a cable of 100-MHz bandwidth is *not* limited to 100 Mbps. It can usually go to much higher bit rates. We know that using all four pairs of CAT5e, rates up to 1 Gbps can be achieved. Many factors come into the picture in determining maximum bit rate. The upper megabits per second (data rate) achievable is very hardware-dependent for

one. In addition, modern Ethernet PHYs (transceivers) use many novel techniques to extend data rates. These are out of the scope of this book, but if the reader is interested, he or she can refer to "Manchester coding" on the Internet, and branch out from there. Underlying all this, there is in fact a fundamental relationship between the bandwidth of a channel expressed in megahertz (MHz) and the maximum information capacity (or data rate) expressed in megabits per second.

A good analogy is the traffic flow on a major highway. Bandwidth is similar to the number of lanes of traffic on a highway. The data rate is very similar to the traffic flow (the number of vehicle crossing over per hour). So one obvious way to increase the traffic flow (data rates) is to widen the highway (increase bandwidth). But another way is to, say, improve the road surface, eliminate bottlenecks, use better signage, and so on (lower the cross talk, use special encoding schemes, and so on). It is therefore possible to pack more bits of information per Hertz of available bandwidth; but that requires a higher SNR.

NOTE *The mathematical relationship between bandwidth and information capacity was discovered in the 1940s by Claude Shannon, an engineer with Bell Telephone Laboratories. This is called the Shannon limit or the Shannon-Hartley theorem. It determines the maximum information rate for a noisy channel as a function of the available bandwidth and the SNR. DSL is also credited to Shannon. As per Wikipedia, "the theory behind DSL, like many other forms of communication, can be traced back to Claude Shannon's seminal 1948 paper: A Mathematical Theory of Communication… He is also credited with founding both digital computer and digital circuit design theory in 1937, when, as a 21-year-old master's student at MIT, he wrote a thesis demonstrating that electrical application of Boolean algebra could construct and resolve any logical, numerical relationship. It has been claimed that this was the most important master's thesis of all time." For such contributions, Shannon is often called "the father of information theory."*

Effect of Temperature on Cable Performance

Ideally, we want the network to be unaffected by our decision whether to run power down the cable (use PoE) or not. We want power and data to be separate and as transparent from each other as possible. Otherwise, for one, troubleshooting can become very challenging. Many techniques and tricks are employed to make the separation of power and data over Ethernet cables a reality, and we will discuss some of these later in this chapter. But there is an obvious manner in which they can interfere, and we will discuss that here.

Signal strength is a critical factor in overall network performance. A lower Insertion Loss is the functional equivalent of a strong signal

at the receiver end. We prefer thicker conductors because that lowers Insertion Loss and thus helps improve the SNR, thereby increasing immunity to external and internal noise sources. We also realize that cables with a lower Insertion Loss will be able to support longer distances. What we may not immediately recognize is that good cables also support a higher-operating temperature range. Cables are often installed in ceiling spaces, air plenums, and riser shafts, where the ambient temperature is much higher than in a typical air-conditioned environment. A study performed by the Lawrence Berkeley National Laboratory at the University of California revealed that temperatures in plenum spaces of medical buildings could reach as high as 49°C on a hot day in the middle of summer. We can expect that in tropical countries and/or in warehouses and factory environments, even higher cable temperatures will be encountered. Add to that possible self-heating if we are also sending PoE down the cables.

Keep in mind that Ethernet cables are typically rated *only* up to 60°C. In the long term, high temperatures can adversely affect the life expectancy of the cabling. In the short term, performance can be severely affected because the resistivity of copper increases significantly with temperature.

Let us do the math here. The resistance of copper goes up 4 percent every 10°C. For example, if a certain cable has a resistance of 10 Ω at 20°C, then at 30°C the resistance is 10 × 1.04 = 10.4 Ω. What is the resistance at, say 60°C? Note that some wrongly say that since 60 – 20 = 40, the resistance has gone up by 4 + 4 + 4 + 4 = 16%, which gives 10 Ω × 1.16 = 11.7 Ω. That is not quite correct! The actual increase needs to be calculated based on the cumulative factor: 1.04 × 1.04 × 1.04 × 1.04 = 1.17, which leads to an increase of 17 percent, which in turn leads to 10 Ω × 1.17 = 11.9 Ω. Agreed, it doesn't seem to be much different from the 11.7 Ω calculated by the previous (incorrect) method, but in general, the first method is inaccurate and can produce noticeable error.

Knowing that the resistance of copper goes up 17 percent from 20°C to 60°C (a rise of 40°C), and since DC losses depend on I^2R and are clearly proportional to R, we expect cable losses related to PoE to also go up 17 percent for the same temperature rise (for a given maximum current, I).

From the viewpoint of data/signal transmissions, the Insertion Loss also goes up proportionately. But note that Insertion Loss is usually expressed in decibels (dB). So raising the temperature by 10°C, leads to an increase in Insertion Loss by the amount 20 × log (1.04) = 0.34 dB. Similarly, going all the way from 20°C to 60°C, the Insertion Loss increases by 20 × log (1.17) = 1.36 dB. In decibels we can just add up numbers. So we could have written the increase in Insertion Loss from 20°C to 60°C as 0.34 dB + 0.34 dB + 0.34 dB + 0.34 dB = 1.36 dB. The "wrong math" would have given us 20 × log (1.16) = 1.29 dB, noticeably different from the correct answer of 1.36 dB.

What do these numbers really imply? Consider a cable of 90-m length at 20°C. If we raise its temperature up to 40°C (a rise of 20°C), the resistance goes by a factor 1.04 × 1.04 = 1.082. That is just 8.2 percent higher. But the Insertion Loss also goes up by the same factor. So to have the same transmission performance at 40°C as a 90-m cable at 20°C, we need to reduce the length of the cable by the very same factor too: that is down to 90/1.082 = 83 m. So just a 20°C rise has impacted the data reach by 7 m. In other words, if the 90-m wire was just acceptable ("marginal") for a given application at 20°C, it will certainly have serious trouble in the form of data bit-errors as the cable heats up, unless we started off with a smaller cable (83 m in this case) than was just *adequate* at 20°C (90 m).

As mentioned, the increase in temperature of the cable may be caused by rising ambient temperatures, but also due to self-heating from PoE losses. Since this will also cause an increase in Insertion Loss, to truly keep data and power separate (transparent from each other), we need to account for PoE-induced temperature rise upfront: if necessary by using a *better-quality* (nonmarginally-compliant) cable.

Cable Temperature Rise Caused by PoE

We need to know the expected temperature rise caused by PoE self-heating so we can estimate more accurately the maximum temperature of the cable, and thus prevent deterioration in signal-transmission capabilities (increase in Insertion Loss).

We will keep this simple. Temperature rise and the maximum allowable PoE current was the subject of several committees, reports, and intense discussions, especially during the creation of the IEEE 802.3at standard. But the dust has settled, so it is enough to just quote the results that matter to us going forward.

Initially, during the creation of the older (IEEE 802.3af) standard, the logic was very simple. The TIA liaison reported that existing infrastructure was rated for an absolute maximum of 500 mA on any one conductor. That was the starting point. Keep in mind that at this stage, the assumption was CAT3 cabling with AWG26. Now, as we will soon learn, although a normal PoE connection uses both conductors of a twisted pair in parallel, the committee decided *not* to allow twice the current per pair (2 × 500 mA = 1 A). The reason is a) active current balancing is not present, so we can't say for sure how the PoE current will actually distribute on the two wires of the center-tapped pair, b) in addition, we may also have a defective connector, with continuity on only one conductor. And so if 1-A wire were to be allowed on a twisted pair, we could, under faulty connector conditions, get 1 A flowing through only one conductor. That would be unsafe. So the absolute maximum current was fixed at 500 mA

per twisted pair. To comply with this absolute maximum, a fairly fast-acting current limit with a typical ± 50 mA tolerance (± 10% of 500 mA) needs to be set. Its nominal (center) value must be at 450 mA. Because then we get a practical current limit lying anywhere in the range 450 ± 50 mA, or 400 to 500 mA. In other words, with tolerances considered, the lowest level of the current limit could be worst-case 400 mA. Now coming to normal operation, we typically also want to include an overload region just above the normal continuous current rating. This will allow a typical device running off PoE power to draw momentary surges of power if necessary (as per the normal operating profile of most devices), without the port being shut down by the activation of the current limit. So if we plan on a 50-mA overload region (below the lowest value of current limit), we get the normal continuous current rating of the cable as 400 mA – 50 mA = 350 mA. And that's how 350 mA was fixed as the maximum continuous PoE (DC) current in IEEE 802.3af. It corresponds to 13 W at the end of 100-m of CAT3 cable as we will soon see.

When the AT standard (IEEE 802.3at) was being drafted, the cable category under discussion was CAT5e (for medium-power/Type-2 applications). TIA guidance recommended a maximum temperature increase of 10°C because of PoE self-heating in a typical cable bundle, up to an absolute maximum cable temperature of 60°C, which is the maximum temperature rating of most Ethernet cables. But that implies that the maximum ambient temperature is restricted to 60 - 10 = 50°C (for Type-2 applications). We then have the desired headroom of 10°C for PoE self-heating, without exceeding the rating of the cable. With several tests on cable bundles, the committee found that 600 mA is a good value, since it gives about a 7.2°C rise. Yes, there is some *additional* built-in headroom here, since the temperature rise is less than 10° C, but that is certainly nice to have and can only help in extending the life of the cabling. And that's how, in a nutshell, 600 mA was fixed as the maximum continuous PoE (DC) current in IEEE 802.3at for Type-2 (medium-power) applications. It corresponds to 25.5 W at the end of 100 m of CAT5e cable, as we will soon see.

NOTE *The temperature rise of 7.2°C is actually for the case of power applied through only two pairs of the four available pairs of an Ethernet cable. Specifically, we have 600 mA flowing in a forward direction through one pair, and the same current returning through the other pair. Two pairs are always unused in normal IEEE-compliant PoE, whether Type-1 or Type-2. But as an experiment, if all four pairs are energized with 600 mA (1.2 A going forward through two pairs, and 1.2 A returning through the other two pairs), the observed temperature rise is 10°C. We see that this temperature rise is also acceptable as per broader TIA guidelines. And that is the reasoning driving some new industry standards for four-pair PoE. For example we have recently seen, Universal Power over Ethernet,*

(UPOE) from Cisco. This corresponds to twice the output wattage, that is 25.5 × 2 = 51 W at the end of 100 m (of CAT5e cable). But keep in mind that still, IEEE PoE standards apply only to 2-pair PoE. four-pair PoE is not covered by the standard, nor is it ruled out. For example, Section 33.1.4.1 of the AT standard deliberately kept the door open for that future possibility.

Some caution needs to be applied in interpreting the listed results and recommendations. First, we are not allowed to increase the max current (above 600 mA) if the ambient is somehow known to be lower than 50° C. This, in effect, means we are not just concerned about the actual average temperature of the cable bundle, but its temperature gradient or *rise* (δT) above ambient too. A higher temperature rise will create hot spots inside the cable bundle, possibly degrading the life of the cabling infrastructure. For that reason, a cable temperature rise greater than 10°C above ambient is not allowed under any circumstances. Second, nor is there some simple formula to allow us to lower the max current judiciously, allowing us to raise the ambient above 50°C, though still staying less than 60°C (by the use of some derating curve). There is simply no derating curve presented in the IEEE standard. The rules were created to keep things simple as far as possible, and also ensure life expectancy of the cabling infrastructure.

The bottom line is we are not allowed to go above 50°C ambient for both Type-1 and Type-2 applications, nor above 350 mA and 600 mA for Type-1 and Type-2, respectively.

With that background and understanding of how the twisted cable was adopted and used in modern Ethernet, we get back to another key reinvention from the past, the center-tapped transformer. It is an implementation of the "phantom" circuit principle we had mentioned previously in connection with the 2010 DSL breakthroughs.

The Center-Tapped (Hybrid) Transformer and the Phantom Circuit

In the top schematic of Fig. 1.5, we see the signal from the microphone being transmitted down a twisted pair to a loudspeaker inside a remotely located telephone. That ensures Person A can talk to Person B. But what about *reverse* communication? We need Person B to talk to Person A too, and *simultaneously*. We could use a second twisted pair for that. See the uppermost schematic of Fig. 1.8. That works, but it is neither smart nor cheap. Can we save one twisted pair? The historical answer to that was the hybrid transformer. It is presented in a very simplified form in the second schematic in Fig. 1.8 (shown in more detail in Chap. 13). We are not going to do any math here, but we should notice the separation of the microphone

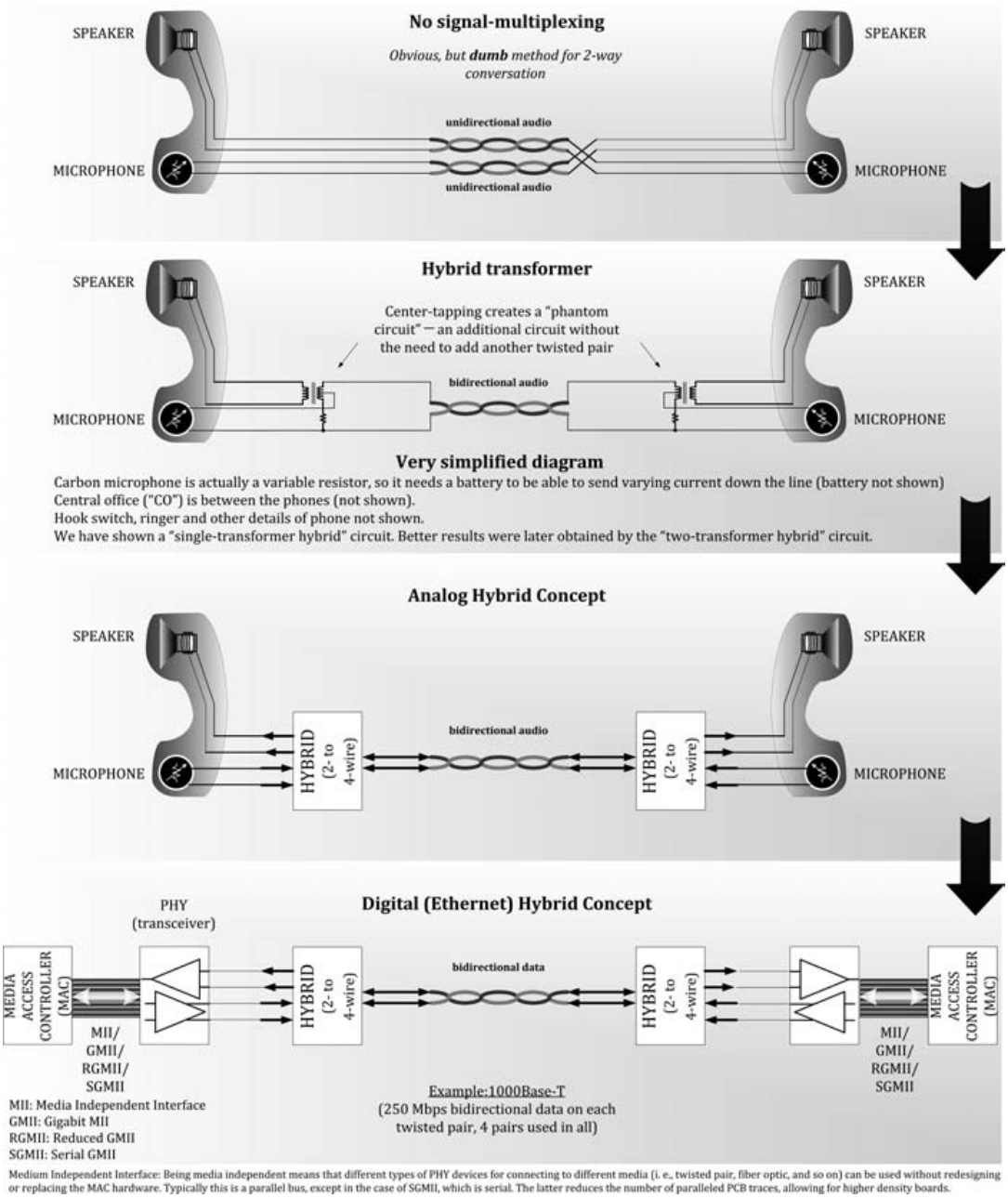


Figure 1.8 Development of the hybrid (2- to 4-wire) concept from telephony to Ethernet.

from the speaker by center-tapping. This is a clue to the overall concept used here. Subsequently, with the advent of electronics, the hybrid transformer disappeared and was replaced by active circuitry, though the circuit block was still aptly called a "hybrid" circuit. It is basically a 2- to 4-wire (or in the reverse direction a 4- to 2-wire) multiplexer of sorts. The same concept is used in 1000Base-T, in which all four pairs of the Ethernet cable are used for data transmission, and each pair is *bidirectional* as shown. Of course, we cannot connect the

output of a differential transmitter directly to the differential receiver located next to it, for that will lead to the digital equivalent of "audio feedback"—the familiar howling we often hear during stage shows when the microphone happens to catch (and amplify) its own sound coming from the speakers. Clearly, we need to insert a separator/multiplexer, as shown. As indicated, for historical reasons, this multiplexer is also called a "hybrid" in Ethernet terminology.

Returning to the second schematic from the top in Fig. 1.8, we see that we have multiplexed two audio signals on the same twisted pair by using a center-tapped transformer. This could be one of the earliest such circuits discovered and used. In effect we are creating an additional circuit (we can call it a phantom or ghost circuit) that rides on top of the existing twisted-pair circuit. It is somewhat like two people sharing the same seat on a bus, unaware of each other. In the process, we are saving a seat (a twisted pair in this case).

To really understand the principle behind the phantom circuit, we need to open our old high-school physics book to a page we have likely forgotten long ago: the *Wheatstone Bridge*. Historically, that's exactly how the principle of phantom circuits was first discovered and analyzed. In Fig. 1.9, we have selected a special case of the Wheatstone Bridge with all its bridge resistances *exactly equal*. By simple voltage-divider principles, we realize that the voltage at the common node between R1 and R2 is going to be $V/2$, where V is the battery voltage. Similarly, the voltage at the common node between R3 and R4 is also $V/2$. And the voltage *across* the load resistor (which is the resistor shown inside the gray box of the other cross-branch) is therefore $V/2 - V/2 = 0$. In other words, *no* current will flow through this load resistor.

Then we do a little "morphing" to show that, in fact, both the load resistor and the battery are in *equivalent* cross-branches of the bridge, even though they may have been sketched in a seemingly different way (with one of them appearing to be inside the bridge, the other outside). In reality, their positions are, fully interchangeable—using *identical* bridge resistors, they are, in effect, *identical positions*. In other words, we could replace the load resistor with another battery too (or a general voltage source), and the two voltage sources will never drive currents *into* each other. We could mix and match and have, for example, a DC source in one cross branch and an AC source in the other. One could even be a voltage source, the other a current source, and so on. We will discover that the two sources in the cross-branches of this "equalized" Wheatstone Bridge *never interact with each other*. In effect they are mutually independent. Each is a "phantom" to the other. So neither sees the other. Of course the resistors "know better." Each source will drive its corresponding current contributions through the four resistors of the bridge. To calculate the total currents in the resistors, we need to calculate the current contributions corresponding to one source being present, *assuming the other source is not even connected*. Then we add their individual contributions to get the net current in

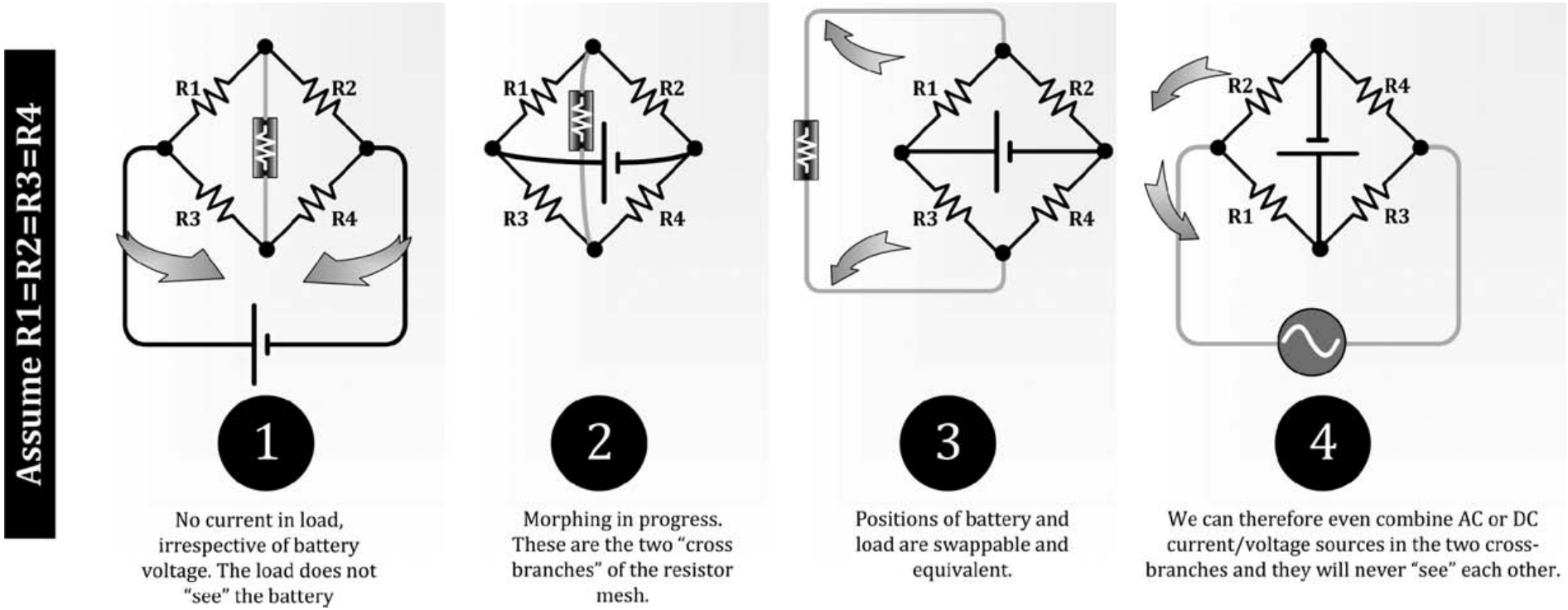


Figure 1.9 The two cross-branches of a Wheatstone Bridge with equal resistances have equivalent positions and are independent of each other.

each resistor. This process is shown with a numerical example in Fig. 1.10. We thus see that the two voltage sources (persons) can share the same bridge resistors (seats), but remain independent (unaware) of each other. Right out of the pages of a Harry Potter novel!

In Fig. 1.11 we take this bridge morphing further in a few simple steps. In the first schematic (marked 1), we create two independent circuits: one involving an AC signal (in this case a telephone signal actually, symbolically indicated by the circle with a "T" inside), the other with a battery and load resistor in series with each other. The two circuits are independent, as mentioned previously. We now also clearly see that the current flow produced by either source does not go through the other source. In the next schematic (marked 2, we show that we could do the same thing, not by using resistors but by using identical *inductors*)

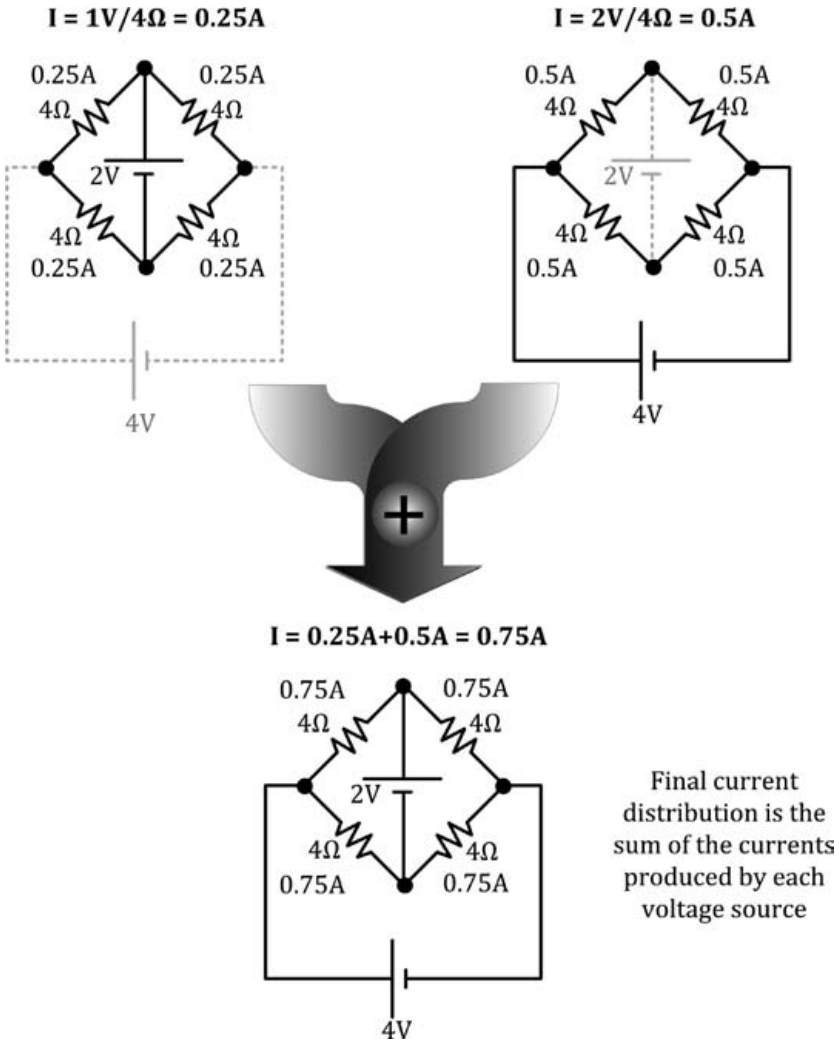


Figure 1.10 Numerical example of how two voltage sources in the cross-branches produce currents independent of each other, which can then be added up to get the net current in each resistor.

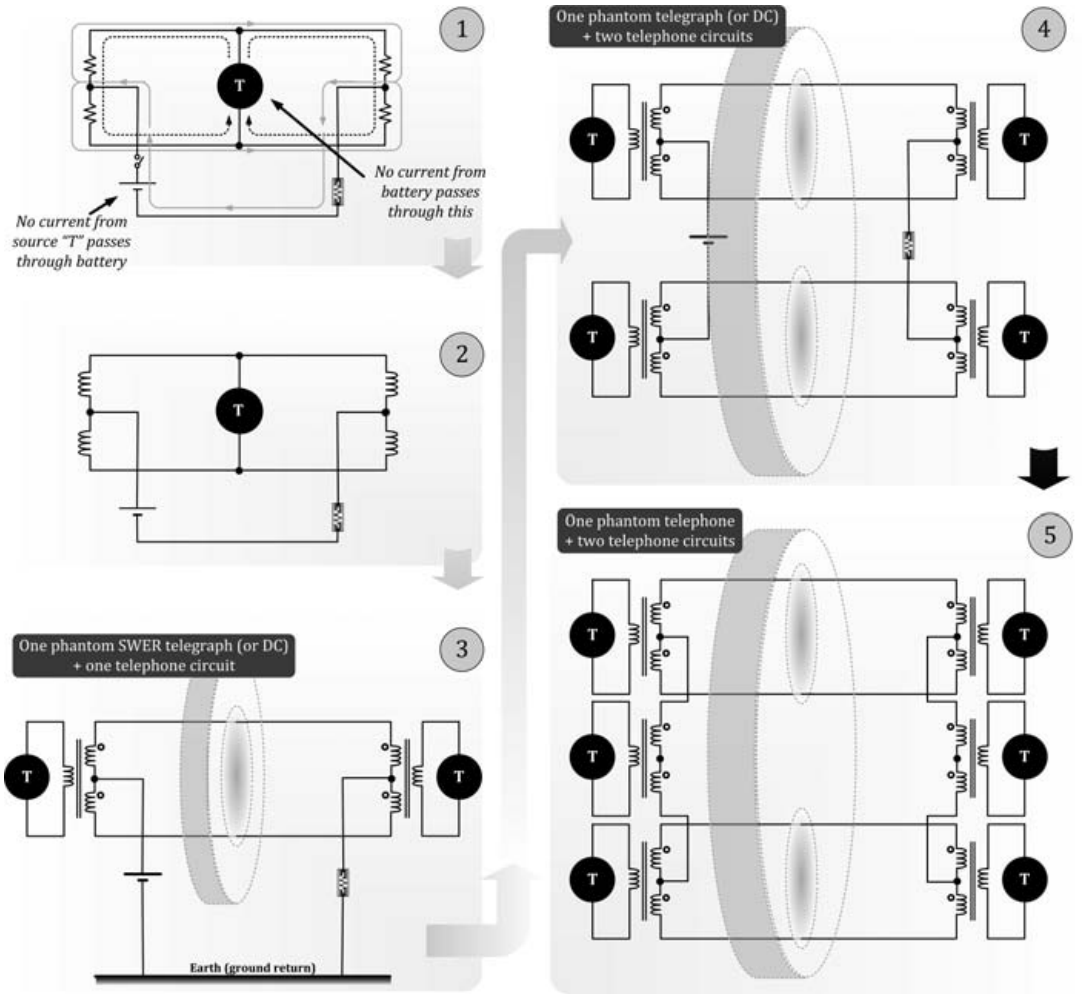


Figure 1.11 Morphing the Wheatstone Bridge to produce several exemplary phantom circuits.

(though in this case we are assuming there is some real-world winding resistance present for ensuring that sufficient impedance is presented to the DC source; otherwise it will get shorted out). In the next schematic (marked 3), we replace the inductors by transformers. The long connecting wires are now considered part of a (single-pair) cable. In this case, if the cable is long enough, and assuming the resistances of its two wires are equal, we do *not* need the transformer windings to have any resistance at all. This particular schematic can serve as a telegraph (or switched-DC) circuit based on SWER architecture, in combination with a normal telephone circuit.

To remove the SWER architecture, in the next schematic (marked 4), we introduce *two* identical Wheatsone Bridges, corresponding to the case of *two* twisted pairs. We see that the two twisted pairs carry not only two telephone circuits, but a phantom telegraph/DC circuit with a proper return wire (not requiring a ground return). That is, in fact, almost exactly what we do in PoE today. (See top of Fig. 1.12.)

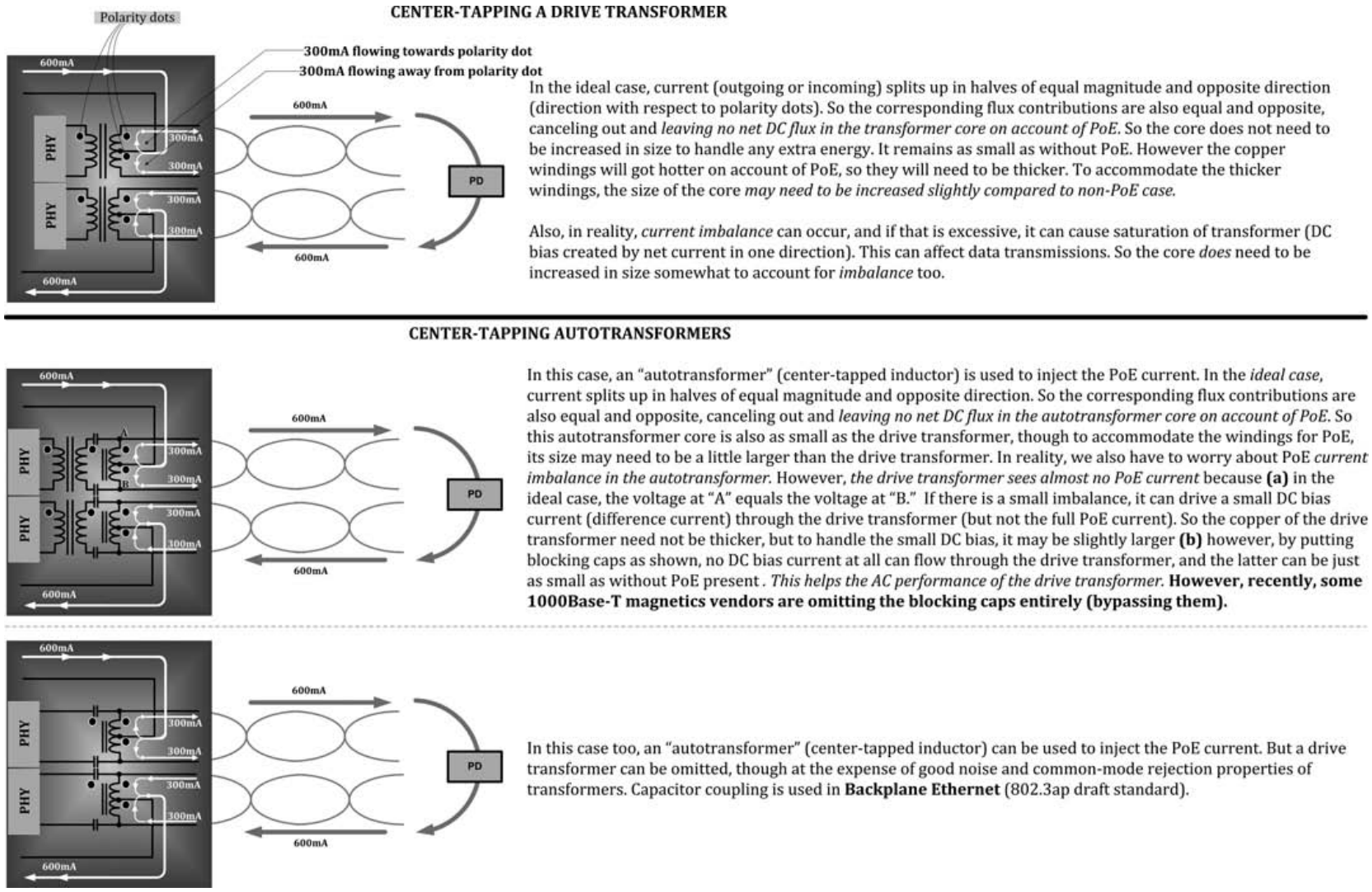


FIGURE 1.12 Flux cancellation principle for ensuring small-sized magnetics in phantom-powered PoE center-tapped drive transformers and autotransformers.

We will explain Fig. 1.12 in more detail shortly. For now, coming back to Fig. 1.11, note that in the last schematic (marked 5), we have replaced the DC source by a new (phantom) telephone circuit. This is in fact the historical way in which *three* telephone circuits were created out of just *two* existing twisted pairs—in effect, providing a free (phantom) telephone circuit without the added cost and complexity of actually laying out a new twisted pair across several miles.

We have discovered the sheer resourcefulness and ingenuity of engineers working in that obscure period from the late-19th and early-20th centuries. Keep in mind that the phantom DSL breakthroughs of 2010 we talked about earlier are based on the phantom circuit principle, and in fact, perhaps both the DSL breakthroughs are very similar to the last circuit discussed (with some proprietary enhancements to reduce "cross talk" and so on). The underlying idea of using transformers instead of resistors in the Wheatstone Bridge came from J. J. Carty in 1886. That was the 100-year-old networking trick Alcatel Lucent talked about.

Methods of Injecting PoE via Phantom Power

Schematic 4 in Fig. 1.11 is the basic way of inserting PoE by phantom power. The key difference, or rather addition, is a pass-FET in series with the battery on the left side and another pass-FET in series with the load on the other. The purpose of these pass-FETs and related circuitry is basically to impart some intelligence to the power-delivery scheme, and that is discussed in great detail in the following chapters.

In Fig. 1.12, in the upper portion, we have drawn this popular way of injecting (and extracting) PoE (DC power) on an Ethernet cable via the center-taps of the data transformers. We have learned that this is the way telegraph and telephone circuits have been historically combined for over a century. Nothing new here. We should have known it would work from way back in *1886*. Metcalfe also talked about the possibility of combining data and power lines in his 1973 memo but did not mention phantom power specifically.

The question is how did this evolve in PoE more recently?

One of the earliest references to center-tapping of drive transformers for combining power and data (digital) is U.S. patent number 5,065,133 filed in 1989. This one is assigned to The Siemon Company, and the inventor is Gary Howard. Its basic intent is simply to increase the reach of digital signals over unshielded twisted pairs by creating an "enhanced analog signal," by suitably mixing the digital signals with AC power. With some tuned-impedance matching, this creates transmission-line effects and extends the range of the digital signals, which would otherwise get severely attenuated, if not distorted, in UTP cable. The method of combining data and AC power as per the

patent, is basically schematic number 5 in Fig. 1.11, except that the inventor has used the phantom circuit (in the middle), not for a third telephone line (voice), but for transmitting AC power. In other words, the center circular source shown marked "T" should now be marked "AC" instead. The AC frequency could be derived from household mains wiring, but it is better to use higher frequencies for reducing the size of filters, and so on. The idea of extracting the AC power and using it for remote powering is also mentioned in this patent. This was indeed clever and ground-breaking.

The above patent was later cited by a key U.S. patent number 5,148,144, filed in 1991. This one was assigned to Echelon Systems Corp., Palo Alto, and the inventors are Philip H. Sutterlin et al. This seems to be one of the first showing center-tapped data transformers for DC remote powering. It also contains a very good discussion on the advantages of phantom powering, and for the first time perhaps, it talks of how center-tapping avoids "core saturation." Keep in mind that when J. J. Carty introduced his idea in 1886, they were using audio-frequency transformers that were big and bulky to start with. These also had many turns on them, and so there was plenty of DC resistance to limit the currents. These transformers were also wound on iron-cores, and we know today that iron-cores can support very high flux densities without core saturation. So the whole idea of reducing transformer core size and also avoiding core saturation, was of little concern back then. No saturation was likely ever observed. So it seems plausible that the 19th-century inventors were themselves unaware of the biggest advantage that center-tapping brought to the table: avoiding core saturation. But with ever-decreasing sizes of components today, it is something we are very cognizant of today. Especially when using the tiny ferrite/powdered iron/Kool Mu cores found in typical data transformers today.

We now realize that center-tapping has a big advantage in avoiding core saturation when injecting power—AC, DC, or PoE. If not for center-tapping, we would need to significantly increase the size of the drive-transformer core for the sake of adding PoE capability. That would be hardly desirable. For one, the AC characteristics of such a bulky drive transformer will get severely compromised as pointed out in the patent number 5,148,144 too. Besides, a typical switch/hub will have 4, 8, 24, 48, or 96 ports. Each port has two drive transformers at least, and so in the interest of keeping overall size of equipment manageable, we need to keep the magnetics very tiny, despite introducing PoE. And center-tapping is the way to do that.

What determines the size of a transformer? Every core has a physical limit as to the amount of energy it can store. Larger cores can store more energy. Flux ($\Phi = B \times \text{Area}$) corresponds to stored energy. It is proportional to ampere-turns. So from ampere-turns, we can estimate core size. A complete treatment of magnetics can be found in this author's *Switching Power Supplies A–Z* book.

For calculating ampere-turns at any given moment, we need to (algebraically) sum up the product of the current in every winding placed across a core and the number of turns of that winding (ΣNI). The sign of the current is determined with respect to the polarity dots of the winding, as shown in Fig. 1.12. In center-tapping, *in an ideal case*, current splits up exactly equally in the two halves, and these current components have opposite polarities (away from the dot, toward the dot). So they cancel out completely in terms of the flux in the core. For all practical purposes, the core does not "see" the PoE current, unless there is an *imbalance*, and then it would just see the "difference current" (difference of the magnitudes of the currents in the two halves). We also realize, the transformer core would need to be a little larger to handle real-world imbalances as discussed in more detail in Chap. 9. However the copper windings *do* see the full PoE current, and there is no "cancellation" at work here, because heating depends on I^2R, and the squaring of current masks its sign anyway. So the copper windings *will* need to be somewhat thicker for a "PoE-capable" drive transformer. To accommodate these thicker windings, on rare occasions, the core size may need to be increased just a little, to provide a larger "window." But in general, center-tapping for the purpose of introducing phantom power causes almost no increase in the size of the magnetics. This is what was so explicitly pointed out, apparently for the first time, in U.S. patent number 5,148,144. To quote from that patent (italics inserted by this author):

> …..prior art systems are not without their disadvantages. One major disadvantage of this type of prior art system lies in the fact that *the transformer must be sized to handle the DC current without saturating.* In general, a transformer which can accommodate DC currents without saturating *has much poorer AC characteristics than one in which does not have to handle any DC current.* These degraded AC characteristics are manifested by poor communications signal quality and by a limited bandwidth… . To overcome the difficulties associated with providing power and communications along the same cable, some practitioners have chosen to provide separate conductors for power and message delivery… . A further problem of conventional power distribution approaches is that they tend to make inefficient use of cable… .Therefore, what is needed is a means of providing power and communications over the *same* cable network… . the present invention provides a wire-based communications network in which power and message information is delivered over the same cable network with improved AC characteristics. The enhanced communication capabilities of the present invention permit greater communication speeds and transmission over greater distances.

Do we need to always center-tap the drive transformers for injecting PoE? No, we can also use center-tapped *inductors* instead. These are also called "autotransformers." See the lower portion of Fig. 1.12

for a breakdown of their pros and cons. The first patent that seems to have talked about this alternative method for phantom powering via center-taps is called "Power transfer apparatus for concurrently transmitting data and power over data lines," filed on May 29, 1997. It bears the U.S. patent number 5,994,998, naming David Fisher et al., from 3Com. There were several continuation patents of this initial patent, extending to U.S. patent number 6,710,704 filed in 2002 and more recently U.S. patent number 6,989,735 filed in 2004.

The first mention of the possibility of using phantom power for PoE at the IEEE 802.3af meetings seems to have been on March 10, 1999, by Nick Stapleton of 3Com. On July 6 of the same year, Amir Lehr from PowerDsine (now part of Microsemi) mentioned the possibility. In fact 3Com and PowerDsine were the key companies at the time, urging IEEE to standardize PoE. Later, Yair Darshan of Power Dsine (Microsemi) became one of the key technical persons involved in the development of the PoE standards, along with Fred Schindler of Cisco.

In this manner, PoE got built from the ground up—the "ground" in this case being J. J. Carty's idea from 1886. That idea had worked spectacularly for telephony, later for power over data in Ethernet, and finally data over data in phantom DSL too. It is the old "networking-trick" in its various incarnations, but the same basic principle.

NOTE *The key concern in phantom circuits is that they truly remain "phantom" to each other. In other words, in our case, data should be completely unaffected by power, and vice versa. Unfortunately, the latter is almost a fait accompli,* the former is typically *not: Power can easily affect data. We saw in this chapter when we learned that an increase in temperature of the cable caused by to PoE self-heating will cause an increase in the Insertion Loss, which can affect the reach/quality of data transmissions. We also made initial assumptions about how well-matched/equal the resistances of the Wheatsone Bridge were. Because if they are not, we will get current through the cross-branches, and so, in effect, the two cross-branches will interfere with each other. That means the two subcircuits are no longer very good "phantoms" to each other. We can thus understand that any asymmetry in, say, the center-taps, or even in the wire resistances of the twisted pairs of the cable, can lead to PoE severely affecting data transmissions. These nonidealities will be discussed later in more detail.*

PoE Chip Vendors: The Emerging Landscape of PoE

This happens to be the first book on the subject. We can ask how did the general technical community, more specifically, PoE engineers, survive and learn so far? The answer is with the help of some very useful technical information on PoE and the related IEEE standards

available on several major chip vendors' Web sites. We end this chapter by listing such vendors. Most of them have been a huge part of the technical community at large and the growth of PoE as a field. This book, too, has relied heavily on their technical information in an effort to disseminate and "put it all in one place." The acronyms PSE and PD (see Table 1.1) are further explained in the next chapter.

1. Microsemi (PowerDsine): The pioneers of PoE currently have PSE chips with both internal and external pass-FETs. They also have PD chips, both with only the front-end pass-FET, and also with integrated PWM (DC-DC converter) controller stages.
2. Texas Instruments (TI, along with recently acquired National Semiconductor): They currently have PSE chips with both internal and external pass-FETs. They also have PD chips, both with only the front-end pass-FET, and also with integrated PWM (DC-DC converter) controller stages.
3. Linear Technology: They currently have PSE chips with both internal and external pass-FETs. They also have PD chips, both with only the front-end pass-FET, and also with integrated PWM (DC-DC converter) controller stages.
4. Silicon Labs: They currently have PSE chips with both internal and external pass-FETs. They also have a highly integrated PD chip with on-board bridge rectifiers, front-end section (with pass FET), and a complete DC-DC switcher (including the switching FET).
5. ST Microelectronics: They currently only have PD chips, both with only the front-end pass-FET, and also with integrated PWM (DC-DC converter) controller stages.
6. Broadcom Corp.: Integrated-FET PSE-chip vendor. No further details. Extremely secretive. "Protects" datasheets and App Notes in an electronic documents safe ("docsafe") under heavy surveillance. Known to have unsuccessfully tried to convict departing employees for "espionage"—those who "suspiciously" downloaded files from docsafe (see Tien Shiah case on Google).
7. Akros Silicon: The most highly integrated PD chips with on-board bridge rectifiers, front-end and integrated PWM (DC-DC converter) controller stages. Also includes on-board isolation barrier and secondary-side buck switchers.

And that completes our discussion on the evolution of PoE. In the next chapter, we move on to more specific implementation details.