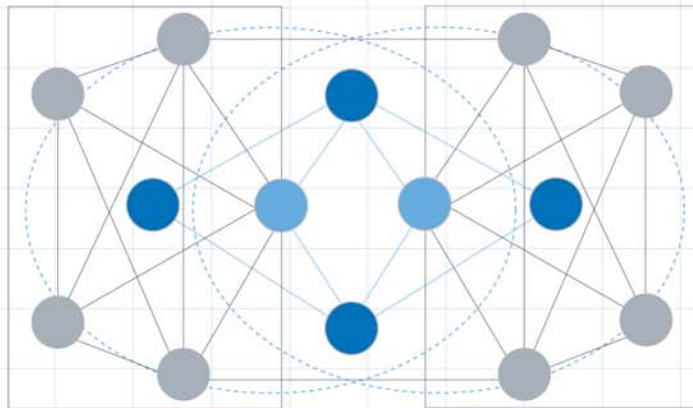




I N S T E  N
Compared



smartlabs

Table of Contents

Introduction	1
INSTEON Overview	3
Hallmarks of INSTEON	4
How INSTEON Works.....	5
INSTEON Is a Dual Mesh Network.....	5
INSTEON Simulcasts Repeated Messages.....	7
INSTEON Is a Peer-to-Peer Network.....	8
INSTEON Facilitates Product Development	9
INSTEON Specifications.....	11
INSTEON Packet Timing	13
INSTEON Powerline Data Rates	14
INSTEON Comparisons	15
Powerline Networks	16
X10.....	16
How X10 Works	16
INSTEON Compatibility with X10	18
INSTEON Compared with X10	19
UPB.....	25
How UPB Works	25
INSTEON Compatibility with UPB	28
INSTEON Compared with UPB	28
Other Powerline Networks	31
LonWorks.....	31
About LonWorks.....	31
INSTEON Compared with LonWorks	32
HomePlug	33
About HomePlug	33
INSTEON Compared with HomePlug	33
Intellon.....	34
About Intellon Technology	34
INSTEON Compared with Intellon Technology.....	34
CEBus.....	35
About CEBus.....	35
INSTEON Compared with CEBus	35
Radio Networks.....	36
ZigBee	36
How ZigBee Works	37
IEEE 802.15.4 Radios.....	37
ZigBee Software.....	38
ZigBee Interoperability.....	39

ZigBee Certification	40
INSTEON Compatibility with ZigBee	40
INSTEON Compared with ZigBee	40
Z-Wave	44
How Z-Wave Works	44
INSTEON Compatibility with Z-Wave	47
INSTEON Compared with Z-Wave	47
Other Radio Networks	50
WiFi	50
About WiFi	50
About TCP/IP	50
INSTEON Compared with WiFi	51
Bluetooth	53
About Bluetooth	53
INSTEON Compared with Bluetooth	53
WiMax	54
About WiMax	54
INSTEON Compared with WiMax	54
UWB	55
About UWB	55
INSTEON Compared with UWB	55
UHF	56
About UHF	56
INSTEON Compared with UHF	56
Conclusion	57
NOTES	58
APPENDIX I — UPB Powerline Noise Immunity Evaluation	59
APPENDIX II — Z-Wave Usability Evaluation	61
Z-Wave Network Setup	61
Z-Wave User Interfaces	63
Missing Z-Wave Modules	64
Z-Wave Lamp Dimming	65

Change Log

Date	Author	Change
10-27-05	Paul Darbee	Begun.
12-21-05	Paul Darbee	Released for comment.
1-2-06	Paul Darbee	Released for publication.

INSTEON, BiPHY, PowerLinc, ControlLinc, LampLinc, Home Network Language, SmartLabs Device Manager, and Plug-n-Tap are trademarks of SmartLabs, Inc. INSTEON networking technology is covered by pending U.S. and foreign patents.

© Copyright 2006 SmartLabs, Inc. 16542 Millikan Ave., Irvine, CA 92606-5027
 949-221-9200, www.insteon.net, www.smarthome.com

Introduction

Wouldn't it be great if...

- You arrive home and the lights come on, the door unlocks, and the thermostat adjusts.
- On a hot day, your blinds automatically close and fans cool the house before your air conditioner comes on, reducing your electric bill.
- Your smart microwave oven downloads new cooking recipes.
- You can use your cellphone to check if you left the garage door open, and close it if you did.
- Your cellphone receives a text message if there is water in the basement.
- When you watch a movie, the lights dim, the blinds close, and your TV automatically turns on the surround sound amplifier.

These are just a few of the scenarios that become possible when everything that is plugged in or battery operated in your home can share information on a network. But what kind of network does it take to link together commodity devices like light switches, door locks, remote controls, and thermostats?

We are all familiar with other kinds of networks. In the U.S., there are computers in three-quarters of our homes, and over two-thirds of them are connected to the global Internet. We use wireless WiFi networking in 17% of our homes, and at an ever-increasing number of commercial 'hotspots' around the world. More than 65% of us have cellphones. But light switches, door locks, thermostats, smoke detectors, security sensors, and remote controls cannot talk to one another, because the networks that share computer data or connect cellphones together are far too complex and expensive to be built into *infrastructure* devices that only cost a few dollars.

INSTEON™ home networking technology fills this gap, because INSTEON embodies the *optimum* combination of simplicity, affordability and reliability. Created by SmartLabs, the world's leading authority on electronic home improvement, INSTEON grew out of thirteen years of experience delivering real-world home control products to consumers and professional installers all around the world.

In 2001, when the engineers at SmartLabs laid down the architecture of INSTEON, they were well aware of efforts by others to bring about the home of the future. X10, for example, was being used by lots of 'gadgetologists' to control lights and appliances, but the aging X10 protocol was simply too limiting, with its tiny command set and unacknowledged, 'press and pray' signaling over the powerline.

SmartLabs knew that a common pitfall for new technology is overdesign—engineers often just can't resist putting in all the latest wizardry. But with added performance, cost goes up and ease-of-use goes down. A classic example was CEBus, designed by a committee of experts who wanted CEBus to be a networking panacea for everything that uses electricity. Even though CEBus did become an *official* standard (EIA-600), developers never incorporated it into mainstream real-world products.

SmartLabs carefully evaluated radio-only communication protocols, such as Z-Wave and ZigBee, but chose not to go down that path. Those protocols not only require complex routing strategies and a confusing array of different types of network

masters, slaves, and other modules, but radio alone suffers reliability problems when installed in metal switch junction boxes or other RF-blocking locations.

Other networking candidates were non-starters as well. Bluetooth radio has too short a range, WiFi radio is too expensive, and high-speed powerline protocols are far too complex to be built into high-volume, low-cost wired or battery-operated devices in the home. Overall, it seemed that everything proposed or available was too oversized and therefore would cost far too much to become a commodity in the global consumer marketplace. INSTEON would have to be created from the ground up.

SmartLabs needed a technology that would embody the ideal home networking infrastructure. INSTEON would have to be:

- Instantly Responsive
- Easy to Install
- Simple to Use
- Reliable
- Affordable
- Compatible with X10

This white paper explains how INSTEON achieves these ideals, and contrasts INSTEON with other networking technologies proposed for home control. In particular, it compares INSTEON in detail with X10 and UPB on the powerline and both Z-Wave and ZigBee over the airwaves.

Many familiar products can be improved by the ability to interact over a network, and a myriad of new products become possible when designers can routinely connect them to a simple, affordable, and reliable 'nervous system' in the home. This kind of *electronic home improvement* promises to change all of our lives for the better, but it can only happen with the right communications infrastructure. As this white paper will show, the ideal embodiment of this infrastructure is INSTEON—exactly the right home-control networking technology for the twenty-first century.

In this White Paper

[INSTEON Overview](#)

Explains the properties of INSTEON from a top-down perspective.

[INSTEON Comparisons](#)

Compares INSTEON to X10, UPB, Z-Wave, ZigBee, and other technologies.

[Conclusion](#)

Recaps the main points of this white paper.

[APPENDIX I — UPB Powerline Noise Immunity Evaluation](#)

Reports a case study of manual lamp dimmers jamming UPB signals.

[APPENDIX II — Z-Wave Usability Evaluation](#)

Examines the user experience with Z-Wave products from two different vendors.

INSTEON Overview

INSTEON enables simple, low-cost devices to be networked together using the powerline, radio, or both. All INSTEON devices are peers, meaning that any device can transmit, receive, or repeat¹ other messages, without requiring a master controller or complex routing software. Adding more devices makes an INSTEON network more robust, because INSTEON devices repeat each other's messages by simulcasting them at precisely the same time, so with more devices the INSTEON signal gets stronger. On the powerline, INSTEON devices are compatible² with legacy X10 devices.

This section explains why INSTEON has these properties and explains them further without going into the details. For a comprehensive introduction to INSTEON, see the white paper, *INSTEON, The Details*, available at www.insteon.net. For complete information, consult the *INSTEON Developer's Guide*.

In this section

[Hallmarks of INSTEON](#)

Gives the 'project pillars' and main properties of INSTEON.

[How INSTEON Works](#)

Shows how INSTEON devices communicate using both powerline and radio, how INSTEON devices repeat¹ INSTEON messages, how all INSTEON devices are peers, and how to develop INSTEON-enabled products.

[INSTEON Specifications](#)

Shows the main features of INSTEON in table form, explains the timing of INSTEON packets on the powerline, and gives INSTEON powerline data rates.

Hallmarks of INSTEON

These are the project pillars that SmartLabs decided upon to guide the development of INSTEON. Products networked with INSTEON had to be:

Instantly Responsive

INSTEON devices respond to commands with no perceptible delay. INSTEON's signaling speed is optimized for home control—fast enough for quick response, while still allowing reliable networking using low-cost components.

Easy to Install

Installation in existing homes does not require any new wiring, because INSTEON products communicate over powerline wires or they use the airwaves. Users never have to deal with network enrollment issues because all INSTEON devices have an ID number pre-loaded at the factory—INSTEON devices join the network as soon as they're powered up.

Simple to Use

Getting one INSTEON device to control another is very simple—just press and hold a button on each device for 10 seconds, and they're linked. Because messages are confirmed, INSTEON products can provide instant feedback to the user, making them straightforward to use and 'guest friendly.'

Reliable

An INSTEON network becomes more robust and reliable as it is expanded because INSTEON devices repeat¹ messages received from other INSTEON devices. Dual mesh communications using both the powerline and the airwaves ensures that there are multiple pathways for messages to travel.

Affordable

INSTEON software is simple and compact, because all INSTEON devices send and receive messages in exactly the same way, without requiring a special network controller or complex routing algorithms. The cost of networking products with INSTEON is held to at an absolute minimum because INSTEON is designed specifically for home control applications, and not for transporting large amounts of data.

Compatible with X10

INSTEON and X10 signals can coexist² with each other on the powerline without mutual interference. Designers are free to create hybrid INSTEON/X10 products that operate equally well in both environments, allowing current users of legacy X10 products to easily upgrade to INSTEON without making their investment in X10 obsolete.

How INSTEON Works

This section explains why [INSTEON Is a Dual Mesh Network](#), [INSTEON Simulcasts Repeated Messages](#), [INSTEON Is a Peer-to-Peer Network](#), and [INSTEON Facilitates Product Develop](#).

INSTEON Is a Dual Mesh Network

There are 109 million households in the U.S. They *all* have powerline wiring, and the ISM (industrial, scientific, medical) radio bands are freely available for unlicensed home use, so communications over the powerline or over the airwaves makes the most sense for home networking.

However, *single media* communication exclusively over powerline or exclusively over radio is fraught with problems. The FCC requires that radio devices using the ISM bands must be able to tolerate interference from other devices, and there are many ways in the home that radio signals can be attenuated and reflected. The powerline is notorious for electrical noise, and the phase bridging problem (see below) can prevent signals from half of the circuits in a house from reaching the other half.

INSTEON solves the single media signaling problems because it is a *dual mesh* network. As shown in the diagram below, INSTEON devices can communicate with each other using *both* radio and the powerline.

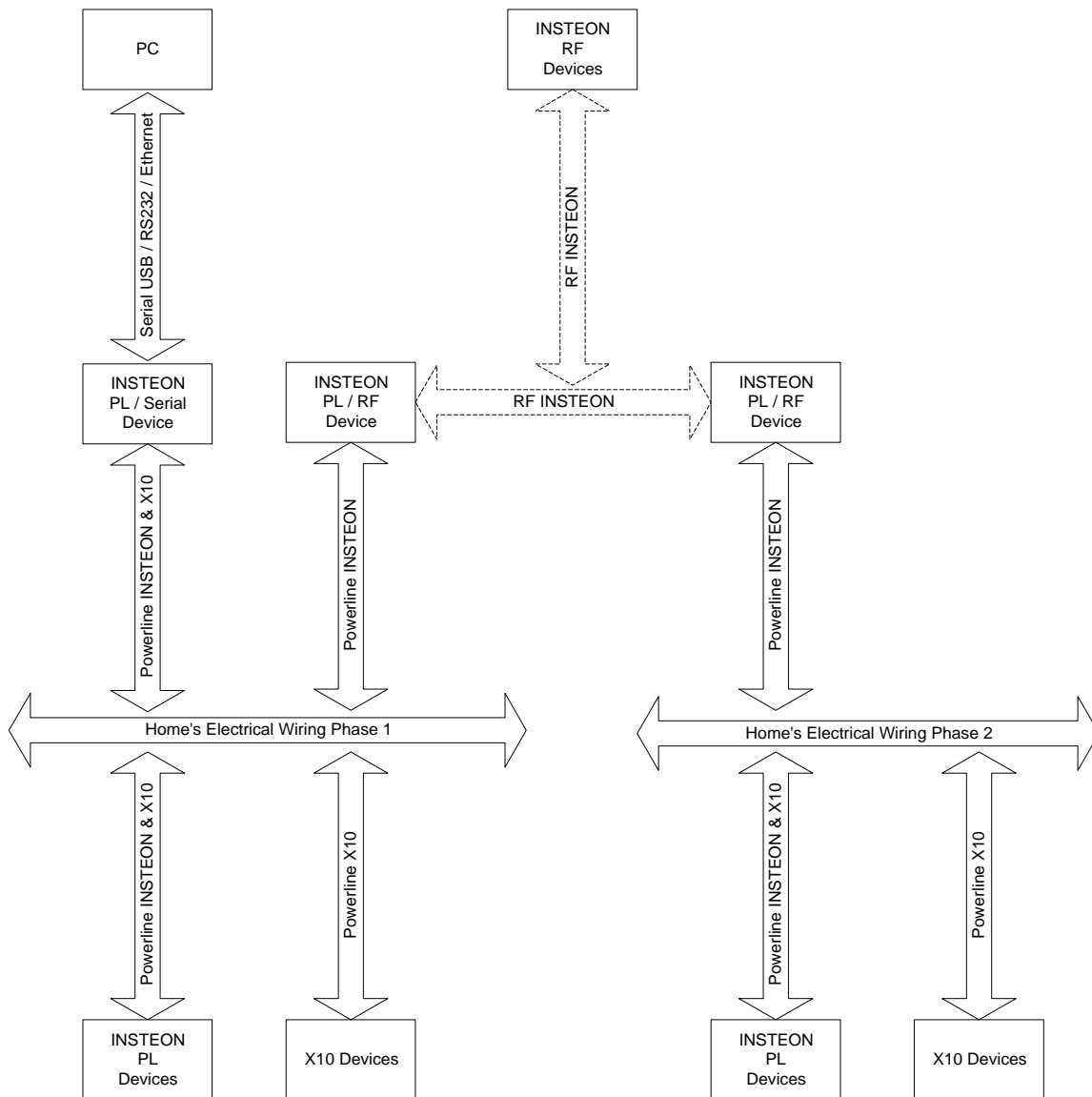
All INSTEON devices that are plugged into the powerline communicate with each other using the INSTEON Powerline protocol. INSTEON devices with radios, whether they are plugged into the powerline or handheld, use the INSTEON RF (radio frequency) protocol to communicate with other INSTEON devices that have radios. Not all INSTEON devices are required to have radios.

Any X10 devices that may be in the home also send signals over the powerline using the X10 protocol. The INSTEON Powerline protocol is compatible² with the X10 protocol, meaning that designers are free to create INSTEON devices that can also listen and talk to X10 devices. Legacy X10 devices, however, are insensitive to the INSTEON Powerline protocol.

INSTEON *BiPHY™* devices (those that can use *both* the INSTEON Powerline protocol *and* the INSTEON RF protocol) solve a significant problem encountered by devices that can only communicate via the powerline. As indicated in the diagram, electrical power is most commonly distributed to homes in North America as split-phase 220-volt alternating current (220 VAC). At the main electrical junction box to the home, the single three-wire 220 VAC powerline is split into a pair of two-wire 110 VAC powerlines, known as Phase 1 and Phase 2. Phase 1 wiring usually powers half the circuits in the home, and Phase 2 powers the other half.

The problem is that powerline signals originating on the opposite powerline phase from a powerline receiver are severely attenuated, because there is no direct circuit connection for them to travel over.

A traditional solution to this problem is to connect a signal coupling device between the powerline phases, either by hardwiring it in at a junction box or by plugging it into a 220 VAC outlet. INSTEON automatically solves the powerline phase coupling problem through the use of INSTEON BiPHY RF/powerline devices—INSTEON RF messaging bridges the powerline phases whenever at least one BiPHY device is installed on each powerline phase.

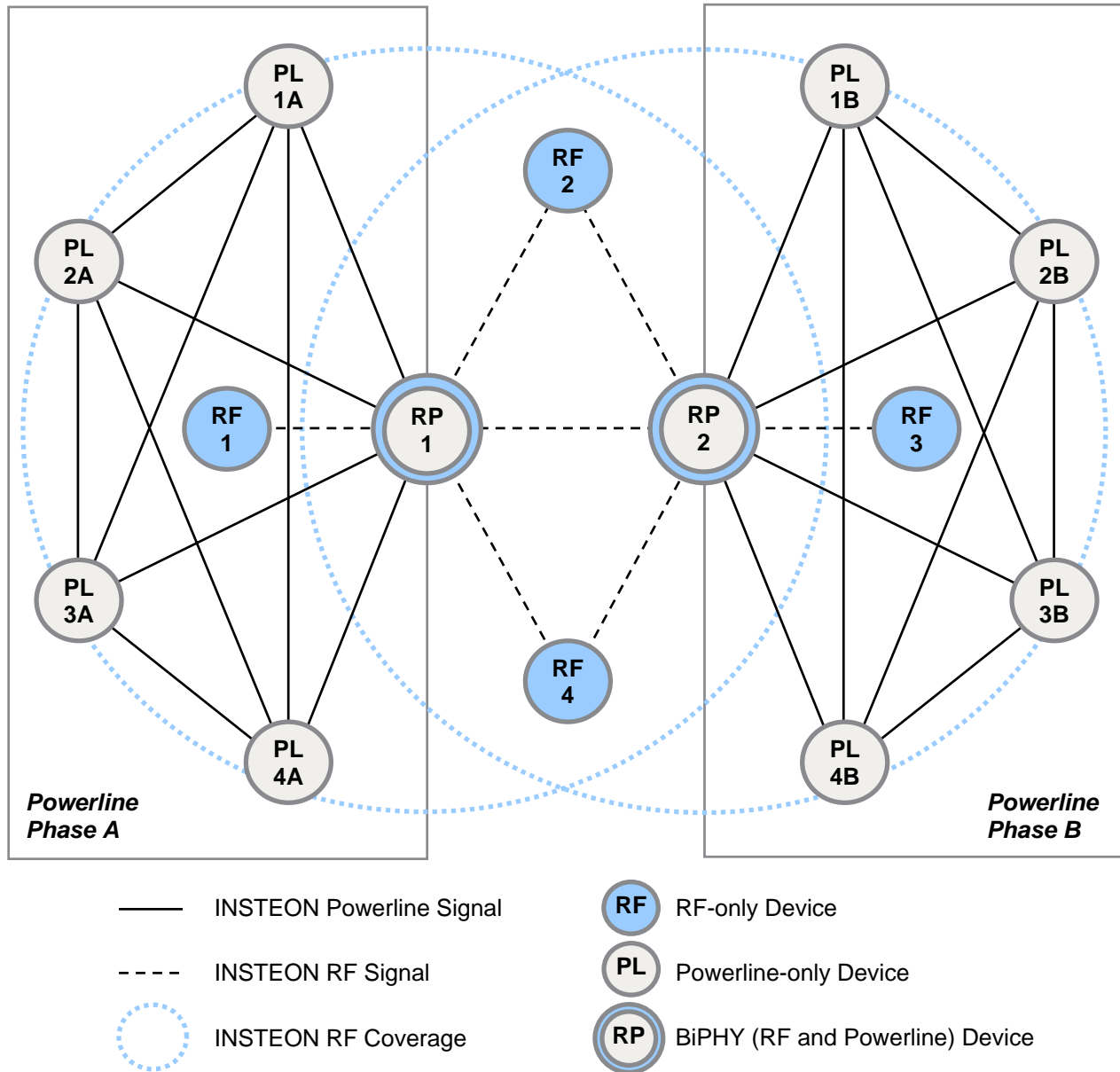


Devices on an INSTEON network can also interface with the larger world. When suitably equipped with a dedicated serial interface, such as USB, RS232, or Ethernet, INSTEON devices can also interface with computers and other digital equipment. In the figure above, an INSTEON device is shown communicating with a PC using a serial link.

Serial communications can bridge networks of INSTEON devices to otherwise incompatible networks of devices in a home, to computers, to other nodes on a local-area network (LAN), or to the global Internet. Such connections to outside resources allow networks of INSTEON devices to exhibit complex, adaptive, *people-pleasing* behaviors. INSTEON devices capable of running downloadable SALad Applications (see [INSTEON Facilitates Product Develop](#), below) can be upgraded to perform very sophisticated functions, including functions not envisioned at the time of manufacture or installation.

INSTEON Simulcasts Repeated Messages

The figure below shows how network reliability improves when additional INSTEON devices are added. The drawing shows INSTEON devices that communicate by powerline-only (PL), RF-only (RF), and both (RP).



INSTEON devices repeat¹ one another's INSTEON messages by simulcasting them in precise timeslots synchronized to the powerline zero crossing. They will do this automatically as soon as they are powered up—they do not need to be specially installed using some network setup procedure. Adding more devices not only increases the strength of the simulcast signal, but it also increases the number of available pathways for messages to travel. This *path diversity* results in highly reliable messaging, so the more devices in an INSTEON network, the better.

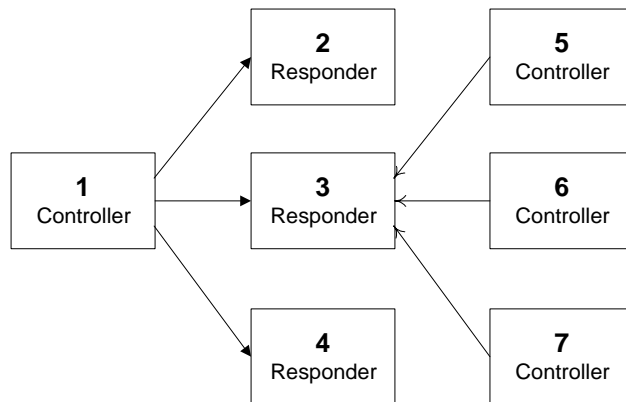
The figure also shows how messages can travel among powerline devices that are installed on different phases of a home's wiring. To accomplish phase bridging, at least one INSTEON BiPHY (hybrid RF/powerline device) must be installed on each powerline phase. In the drawing, BiPHY device RP1 is installed on phase A and BiPHY device RP2 is installed on phase B. The RF path between RP1 and RP2 allows messages to propagate between the powerline phases, even though there is no direct electrical connection.

With all devices repeating messages, there must be some mechanism for limiting the number of times that a message may be retransmitted, or else messages might propagate forever within the network. Network saturation by repeating messages is known as a *data storm*. The INSTEON protocol avoids this problem by limiting the maximum number of times an individual message may be retransmitted to three.

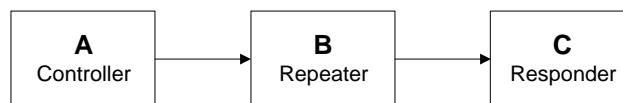
INSTEON Is a Peer-to-Peer Network

All INSTEON devices are peers, meaning that any device can act as a Controller (sending messages), Responder (receiving messages), or Repeater¹ (relaying messages).

This relationship is illustrated in the figure below, where INSTEON device **1**, acting as a Controller, sends messages to multiple INSTEON devices **2**, **3**, and **4** acting as Responders. Multiple INSTEON devices **5**, **6**, and **7** acting as Controllers can also send messages to a single INSTEON device **3** acting as a Responder.



Any AC-powered INSTEON device can repeat¹ messages, as with device **B**, below, which is shown relaying a message from device **A** acting as a Controller to device **C** acting as a Responder.



INSTEON Facilitates Product Development

INSTEON, with its emphasis on simplicity, reliability, and low cost, is optimized as an *infrastructure* network for home integration and control. Common devices in the home, such as light switches, door locks, thermostats, clocks, and entertainment systems currently do not communicate with one another. INSTEON can change all that.

When devices are networked together, there is a potential for coordinated, adaptive behavior that can bring a new, higher level of comfort, safety, and convenience to living. But networking devices together cannot by itself change the behavior of the devices. It is application-level software, created by product developers, that transforms a network of previously unrelated devices into a coordinated, adaptive, lifestyle-enhancing system.

The INSTEON Product Developer Community

SmartLabs is exceptionally supportive of outside developers. A complete INSTEON development kit costs only \$99, yet it includes a PowerLinc™ Controller, A LampLinc™ dimmer module, cables, full documentation, comprehensive software, support certificates, and unlimited access to the *INSTEON Developer Forums*.

There are two basic kinds of applications that developers can create for INSTEON-networked devices: *External Applications* and *Internal Applications*.

External Applications run on a computing device such as a PC, PDA, or custom controller. A special type of INSTEON module called an *INSTEON Bridge* connects the computing device to an INSTEON network. SmartLabs offers a bridge called the *PowerLinc™ Controller* (PLC) that connects a PC to an INSTEON network via a USB or RS232 serial port. Application developers can use a simple scripting interface called *Home Network Language™* along with *SmartLabs Device Manager™* (SDM) software to interface remotely with everything on the INSTEON network. SDM exposes a DirectX interface that lets you embed INSTEON methods, properties, and events directly into other applications, or you can use SDM as an HTTP server to interface with INSTEON via the web or a LAN.

Internal Applications run on INSTEON devices themselves. SmartLabs has developed an embedded language interpreter, called *SALad*, which resides in the firmware of SALad-enabled INSTEON devices. A *SmartLabs Integrated Development Environment* (SIDE) makes it easy for programmers to create and debug *SALad Apps* using a PC. From within the SIDE, you can interface to an external PLC and a network of INSTEON target devices, or if you prefer, you can develop SALad applications offline using a software-simulated PLC and virtual INSTEON devices.

Not only can devices running SALad Apps exhibit very sophisticated behavior, but new SALad Apps can be downloaded into SALad-enabled devices that have already been installed in the home. With both upgradeability and outside-world connectivity, INSTEON can dynamically adapt to people's expectations and needs as the electronic home improvement marketplace evolves.

The INSTEON Alliance

SmartLabs cordially invites companies interested in influencing the development and marketing of INSTEON solutions within the electronic home improvement industry to join over 400 other leading product developers as members of the *INSTEON Alliance*.

Select Membership in the INSTEON Alliance is \$250 per year, and *Elite Membership* is \$2,000—visit www.insteon.net for complete details.

Membership in the INSTEON Alliance provides exclusive insight into the INSTEON community, along with maximum industry exposure for member companies and products. Alliance members are invited to participate in members-only forums and events throughout the year. INSTEON websites, press releases and trade show pavilions further promote Alliance members and their products. And, depending on the level of membership, SmartLabs will place Alliance member products on INSTEON.com, Smarthome.com, or in the Smarthome catalog.

INSTEON Specifications

INSTEON is a true peer-to-peer dual mesh network. Its most important property is its simplicity.

INSTEON messages are fixed in length and synchronized to the AC powerline zero crossings. Because messages propagate by synchronous simulcasting, no network controllers or routing tables are necessary—a three-byte source and destination address in each message suffices.

Optimized for home command and control, INSTEON allows infrastructure devices like light switches, thermostats, security sensors, and remote controls to be networked together at low cost. In turn, these devices can appear as nodes on larger networks, such as WiFi LANs, the Internet, telephony, and broadband entertainment distribution systems, because INSTEON can connect to them using bridge devices.

The following table shows the main features of INSTEON at a glance.

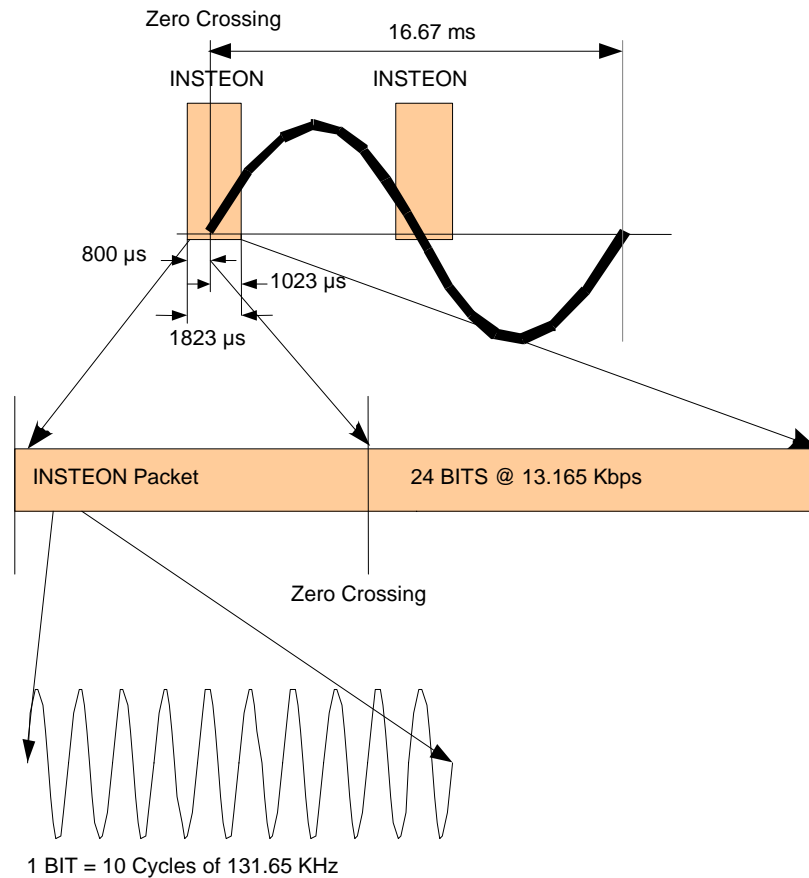
INSTEON Property	Specification	
Network	Dual Mesh (RF and powerline) Peer-to-Peer Mesh Topology Unsupervised No routing tables	
Protocol	All AC-powered devices are two-way Repeaters ¹ Messages synchronized to powerline Repeated messages are simulcast Message Error Detection Messages acknowledged Retry if not acknowledged	
Data Rate	Instantaneous Powerline	13,165 bits/sec
	Sustained Powerline	2,880 bits/sec
	Instantaneous RF	38,400 bits/sec
Message Types	Standard	10 Bytes
	Extended	24 Bytes
Message Format	From Address	3 Bytes
	To Address	3 Bytes
	Flags	1 Byte
	Command	2 Bytes
	User Data	14 Bytes (Extended Messages)
	Message Integrity	1 Byte
Devices Supported	Unique IDs	16,777,216
	Commands	65,536
	Members within a Group	Limited only by memory
INSTEON Engine Memory Requirements	RAM	80 Bytes
	ROM	3K Bytes

INSTEON Property	Specification	
Typical Application (Light Switch, Lamp Dimmer) Memory Requirements	RAM	256 Bytes
	EEPROM	256 Bytes
	Flash	7K Bytes
Device Installation	Plug-in Wire-in Battery Operated	
Device Setup	Plug-n-Tap™ manual linking Optional PC or Controller	
Security	Physical device possession Address masking Encrypted message payloads	
Application Development	sIDE (SmartLabs Integrated Development Environment) SALad interpreted language Software and Hardware Development Kits	
X10 Compatibility ²	INSTEON devices can send and receive X10 commands INSTEON devices do not repeat or amplify X10	
Powerline Physical Layer	Frequency	131.65 KHz
	Modulation	BPSK
	Min Transmit Level	3.16 Vpp into 5 Ohms
	Min Receive Level	10 mV
	Phase Bridging	INSTEON RF or hardware
RF Physical Layer	Frequency	904 MHz
	Modulation	FSK
	Sensitivity	-103 dbm
	Range	150 ft unobstructed line-of-sight

INSTEON Packet Timing

All INSTEON powerline packets contain 24 bits. Since a bit takes 10 cycles of 131.65 KHz carrier, there are 240 cycles of carrier in an INSTEON packet. An INSTEON powerline packet therefore lasts 1.823 milliseconds.

The powerline environment is notorious for uncontrolled noise, especially high-amplitude spikes caused by motors, dimmers and compact fluorescent lighting. This noise is minimal during the time that the current on the powerline reverses direction, a time known as the powerline zero crossing. Therefore, INSTEON packets are transmitted during the zero crossing quiet time, as shown in the figure below.



The top of the figure shows a single powerline cycle, which possesses two zero crossings. An INSTEON packet is shown at each zero crossing. INSTEON packets begin 800 microseconds before a zero crossing and last until 1023 microseconds after the zero crossing.

INSTEON Powerline Data Rates

As explained at greater length in *INSTEON, the Details* (available at www.insteon.net), INSTEON Standard messages contain 120 raw data bits and require 5 zero crossings, or 41.667 milliseconds to send. Extended messages contain 264 raw data bits and require 11 zero crossings, or 91.667 milliseconds to send. The *raw sustained* INSTEON bitrate is therefore 2880 bps (bits per second) for either kind of message.

However, the INSTEON protocol waits for one additional zero crossing after each Standard message, and for two additional zero crossings after each Extended message to allow for transmitter 'politeness' and possible RF message transmission. Therefore, the *actual sustained* bitrate is 2400 bps for Standard messages, or 2437 bps for Extended messages, instead of the 2880 bps second it would be without waiting for the extra zero crossings.

INSTEON Standard messages contain 9 bytes (72 bits) of usable data, not counting packet sync and start code bits, nor the message integrity byte. Extended messages contain 23 bytes (184 bits) of usable data using the same criteria. Therefore, the *usable data* bitrates are further reduced to 1440 bps for Standard messages and 1698 bps for Extended messages. Counting only the 14 bytes (112 bits) of User Data in Extended messages, the *User Data* bitrate is 1034 bps.

These data rates assume that messages are sent with Max Hops set to zero and that there are no message retries. They also do not take into account the time it takes for a message to be acknowledged. The table below shows net data rates when multiple hops and message acknowledgement are taken into account. To account for retries, divide the given data rates by one plus the number of retries (up to a maximum of 5 possible retries).

Condition			Bits per Second		
Max Hops	ACK	Retries	Standard Message Usable Data	Extended Message Usable Data	Extended Message User Data Only
0	No	0	1440	1698	1034
1	No	0	720	849	517
2	No	0	480	566	345
3	No	0	360	425	259
0	Yes	0	720	849	517
1	Yes	0	360	425	259
2	Yes	0	240	283	173
3	Yes	0	180	213	130

INSTEON Comparisons

This section compares INSTEON to existing [Powerline Networks](#) and [Radio Networks](#). On the powerline, both [X10](#) and [UPB](#) are examined in detail, with a number of [Other Powerline Networks](#) also discussed. Of the networks using radio, [ZigBee](#) and [Z-Wave](#) are looked at in particular, with various [Other Radio Networks](#) considered as well.

There are two fundamental differences between INSTEON and *any* of the other powerline or radio networks—INSTEON is a dual mesh network, and INSTEON propagates messages by simulcasting.

INSTEON Is a Dual Mesh Network

INSTEON is the only home-control networking standard that combines powerline and radio communications. Powerline by itself is a demanding environment due to electrical noise, variable attenuation, and phase bridging issues. And anyone who uses a cellphone knows that radio, even sophisticated state-of-the-art digital radio, is far from perfect.

INSTEON's dual mesh approach is a radical breakthrough that uses the strengths of one medium to overcome the weaknesses of the other. Networks that depend on powerline alone or radio alone require complex remedies in an attempt to achieve reliability in the face of inevitable problems with a single medium. Complexity, though engineers may view it as necessary or even elegant, significantly raises prices to mass-market consumers, and it makes products harder to use. In contrast, INSTEON uses low-cost powerline communications backed up by low-cost radio to achieve a price-performance ratio far superior to what can be achieved with either medium alone. When it comes to command and control communications in the home, *INSTEON finds a way*.

INSTEON Simulcasts Messages

There is another fundamental difference between INSTEON and the single-media networking protocols described below—INSTEON simulcasts messages. The other protocols, whether on the powerline or the airwaves, go to great lengths to ensure that messages do not interfere with each other (except for X10, which is too primitive for such concerns). The upshot is that messages must be *routed* through the network, with all of the software complexity and different kinds of hardware devices that such routing entails. Routing is expensive, so routed protocols usually define reduced-function devices that cannot route messages in order to keep costs down for commodity products like light switches. But this kind of cost reduction forces consumers to buy, install and maintain different kinds of devices with different capabilities in order to use a routed network. The tradeoff is difficult for manufacturers and consumers alike—a confusing array of different kinds of devices, versus high-cost devices that all work the same way.

INSTEON's simulcasting solves the dilemma. Whenever an INSTEON device hears a message intended for another INSTEON device, it synchronously repeats¹ the message along with other INSTEON devices that heard the same message. Since the messages are identical, having several devices sending the message at precisely the same time *strengthens* the signal, whether it is on the powerline or on the radio. (To keep messages from traveling forever and causing a *data storm*, there is a counter in each INSTEON message that limits the number of repetitions.)

Powerline Networks

This section compares INSTEON to [X10](#), [UPB](#), and a number of [Other Powerline Networks](#), including [LonWorks](#), [HomePlug](#), [Intellon](#), and [CEBus](#).

X10

At the low end, X10 powerline signaling technology (www.x10.com) has been around since the 1970s, but its early adoption is its limiting factor—it is too unreliable and inflexible to be useful today as an infrastructure home-control network.

How X10 Works

Invented in Scotland by Pico Electronics in 1975, X10 was a pioneering breakthrough for its time, but even though it employed one of the world's first custom integrated circuits, the technology of the 1970s put severe constraints on its design.

For example, there can only be 256 different X10 devices on a single powerline, because each X10 device can only be assigned one of 16 possible House Codes (A through P) and one of 16 possible Unit Codes (1 through 16).

Furthermore, X10 defines only 16 different Command Codes, but not all devices can respond to all X10 commands. The six most common X10 commands are *On*, *Off*, *Dim*, *Bright*, *All Units Off*, and *All Lights On*.

X10 transmits one bit of information at each powerline zero crossing. A one-millisecond burst of 120 KHz carrier signifies a *one* bit, and the absence of a carrier signifies a *zero* bit.

An X10 message consists of a 4-bit **Start Code** followed by an 8-bit **House Code** followed by a 10-bit **Key Code**. Each message is sent twice, followed by 6 zero-crossings of silence before starting another message. (The silence interval can be omitted if certain X10 Commands, such as *Bright* or *Dim*, are being repeated.)

The **Start Code** is always 1110, but the rest of the message consists of complementary pairs of bits (either 01 or 10, but never 00 or 11). If the 10-bit **Key Code** ends in 01, the first 8 bits are interpreted as a **Unit Code**. If the Key Code ends in 10, the first 8 bits are interpreted as a **Command Code**.

The following table shows the meaning of the 16 possible 8-complementary-bit fields when interpreted as a **House Code**, a **Unit Code**, or a **Command Code**.

8-bit Field	House	Unit	Command Code	Command Description
0101 0101	M	13	All Units Off	Turn off all modules on this House Code
0101 0110	E	5	All Lights On	Turn on all lamp modules on this House Code
0101 1001	C	3	On	Turn a module on
0101 1010	K	11	Off	Turn a module off
0110 0101	O	15	Dim	Dim a lamp module by one step
0110 0110	G	7	Bright	Brighten a lamp module by one step
0110 1001	A	1	All Lights Off	Turn off all lamp modules on this House Code
0110 1010	I	9	Extended Code 1	Designated for data and control
1001 0101	N	14	Hail Request	Request a module to send Hail Acknowledge
1001 0110	F	6	Hail Acknowledge	Response to Hail Request
1001 1001	D	4	Extended Code 3	Designated for security messages
1001 1010	L	12	Unused	
1010 0101	P	16	Extended Code 2	Designated for meter reading
1010 0110	H	8	Status On	Reports that a module is on
1010 1001	B	2	Status Off	Reports that a module is off
1010 1010	J	10	Status Request	Request module to send Status On or Off

As an example, the following X10 bitstream will turn on module A1:

```

1110      0110 1001  0110 1001 01
Start   House A  Unit 1
1110      0110 1001  0110 1001 01
Start   House A  Unit 1
00 00 00
Silence
1110      0110 1001  0101 1001 10
Start   House A  Command On
1110      0110 1001  0101 1001 10
Start   House A  Command On
00 00 00
Silence

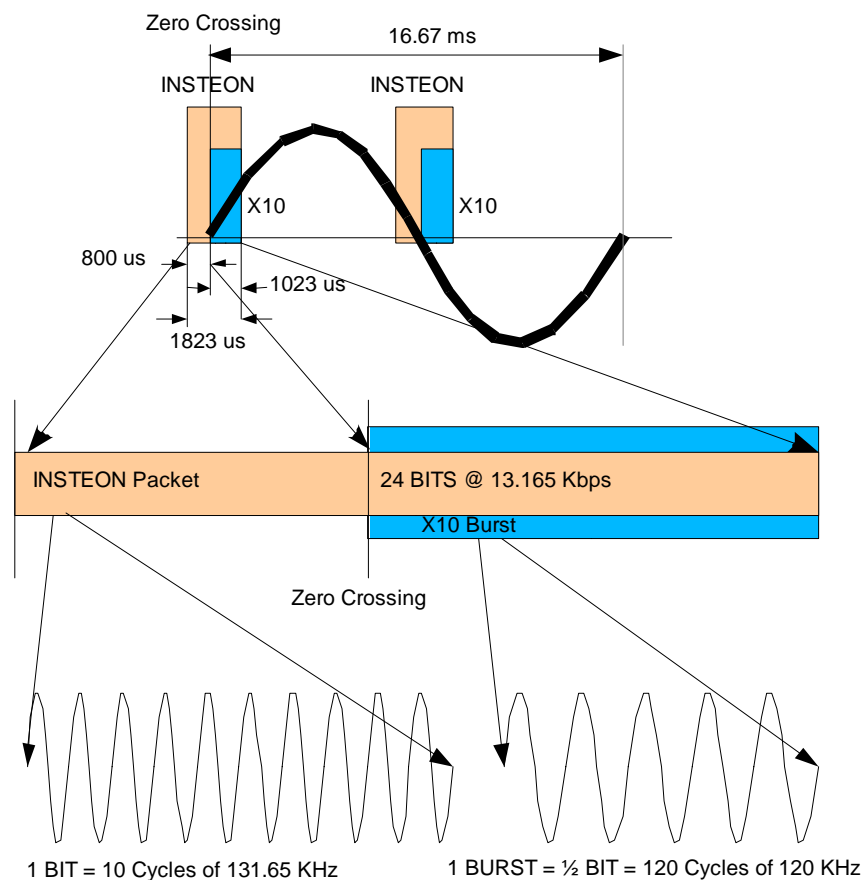
```

One X10 bit is sent each time the 60 Hz powerline voltage crosses zero, which occurs every 8.33 milliseconds. There are 100 bits, so the entire sequence takes 833 milliseconds to send.

INSTEON Compatibility with X10

X10 is the signaling method used by many devices already deployed on powerlines around the world. Compatibility² with this existing population of legacy X10 devices is an important feature of INSTEON. At a minimum, X10 compatibility means that INSTEON and X10 signals can coexist with each other, but compatibility also allows designers to create hybrid devices that can send and receive both INSTEON and X10 signals.

The top part of the following figure shows a full 60 Hz powerline cycle, which possesses two zero crossings. Both an INSTEON packet and an X10 burst are shown at each zero crossing. INSTEON packets begin 800 microseconds before a zero crossing and last until 1023 microseconds after the zero crossing.



The X10 signal uses a burst of approximately 120 cycles of 120 KHz carrier beginning at the powerline zero crossing and lasting about 1000 microseconds. A burst followed by no burst signifies an X10 one symbol and no burst followed by a burst signifies an X10 zero symbol. An X10 message begins with two bursts in a row (a start symbol), followed by a one symbol, followed by nine data symbols. The figure shows an X10 burst at each of the two zero crossings.

The X10 specification also allows for two copies of the zero crossing burst located one-third and two-thirds of the way through a half-cycle of power. These points

correspond to the zero crossings of the other two phases of three-phase power. INSTEON is insensitive to those additional X10 bursts and does not transmit them when sending X10.

The middle of the figure shows an expanded view of an INSTEON packet with an X10 burst superimposed. The X10 signal begins at the zero crossing, 800 microseconds after the beginning of the INSTEON packet. Both signals end at approximately the same time, 1023 microseconds after the zero crossing.

INSTEON devices achieve compatibility with X10 by listening for an INSTEON signal beginning 800 microseconds before the zero crossing. INSTEON receivers implemented in software can be very sensitive, but at the cost of having to receive a substantial portion of a packet before being able to validate that a true INSTEON packet is being received. Reliable validation may not occur until as much as 450 microseconds after the zero crossing, although an INSTEON device will still begin listening for a possible X10 burst right at the zero crossing. If at the 450-microsecond mark the INSTEON receiver validates that it is not receiving an INSTEON packet, but that there *is* an X10 burst present, the INSTEON receiver will switch to X10 mode and listen for a complete X10 message over the next 11 powerline cycles. If instead the INSTEON device detects that it is receiving an INSTEON packet, it will remain in INSTEON mode and not listen for X10 until it receives the rest of the complete INSTEON message.

The bottom of the figure shows that the data bitrate for INSTEON is much faster for INSTEON than for X10. An INSTEON bit is coded in ten cycles of 131.65 KHz carrier, which gives an *instantaneous* INSTEON bitrate of 13,165 bps (bits per second). INSTEON packets consist of 24 bits, and one INSTEON packet can be sent during each 8,333 microsecond zero crossing, so the nominal *sustained* bitrate for INSTEON is 2,880 bps.

On the other hand, X10's *instantaneous* data rate is one X10 *signal* bit per zero crossing, or 120 bps. But two complementary X10 signal bits are needed to code one X10 *data* bit, called a *symbol* above, so X10's *sustained* data bitrate is 60 bps. INSTEON's *sustained* data bitrate of 2,880 bps is therefore 48 times faster than X10's.

Note, however, that these sustained data bitrates must be derated to arrive at actual *payload* data rates. The section [INSTEON Powerline Data Rates](#), above, accounts for INSTEON packet and message overhead, as well as message retransmissions. Similarly, X10's payload data rate must account for X10 message repetition, trailing gaps, and repeated House Codes. Given that sending 20 X10 payload data bits (one House Code, one Unit Code, and one Command Code) requires sending 50 X10 sustained data bits, X10's *payload* data rate is 40% of its sustained data rate, or just 24 bps.

INSTEON Compared with X10

When first introduced in the 1970s, X10 seemed like magic. For a few dollars, you could remotely control your lights and appliances just by plugging X10 controllers and X10 control modules into existing wall outlets. Soon you could buy wire-in wall switches and handheld radio remote controls. There are now X10 devices in several million homes. But if it's so good, why isn't X10 in *every* home?

The main reason is that X10 doesn't work reliably enough. Developed as it was in the 1970s, it is too limited to function as a command and control infrastructure

network in the mass market. Not only is X10 powerline signaling dreadfully slow, but X10 communication has no built-in mechanism to verify that X10 messages got through—X10 is open loop. Although X10 did eventually define Status Request and Status Response Commands, very few X10 devices actually employ them, and those that do take even more time to send and receive the extra messages.

Tweaking a home's wiring so it can support more than a few X10 devices is not for the faint of heart. X10 signals usually do not cross powerline phases, so you will need some kind of active or passive phase coupler wired into your main electrical panel or plugged into a 220-volt outlet (if there is one). You will probably need X10 signal boosters if you want to control devices that are some distance away, and you are likely to find that you will have to buy X10 blocking filters to keep certain electrical devices, like computers and entertainment gear, from swallowing the X10 signal. Running common appliances like electric drills or food processors generates electrical noise on the house wiring that can confuse X10 receivers by preventing legitimate X10 commands from getting through, or worse, by masquerading as false X10 commands.

Even with your home's electrical system all tuned up for X10, you are likely to find that a few times out of a hundred, a controlled light will not go on when you first press the button that you use to control it. Of course, if you can see the light, you could just press the button again, but what if the light is out of sight in the backyard? Or what if you're trying to use X10 to turn off a pool heater, or your lawn sprinklers, and it sometimes doesn't work? Without feedback, you wouldn't even be alerted that you have to manually retry the X10 command.

When you try to do something and it doesn't happen, that is called a false negative. But X10 is also susceptible to false positives, such as lights or appliances going on or off all by themselves, because X10 devices sometimes interpret powerline noise as valid X10 commands. Having your dining room lights go off while you're eating, or waking up suddenly when your bedroom lights go on in the middle of the night does not have to happen very many times before you wonder if the convenience of X10 remote control is worth the aggravation.

Even if X10 were 100% reliable, it doesn't have the flexibility to support true networking. There are only 256 combinations of X10 House Codes and Unit Codes, so there can only be 256 logical devices on a powerline. Furthermore, odds are that you may share that powerline with your neighbors if they are on the same utility transformer as you.

If both you and your neighbor happen to choose the same House and Unit Codes for some of your X10 devices, you might end up turning each other's lights on and off. Worse, if your neighbors are not your friends, there is nothing to prevent them from maliciously controlling X10 modules in your home just by trying out the 256 possible addresses at random.

You can buy X10 remote control handsets and keyfobs that send radio signals to special receivers that translate the radio commands into X10 commands on the powerline. Although both radio and powerline signaling are involved, the two media do not augment each other. X10 radio commands are entirely separate from X10 powerline commands, and the radio is one-way from the handset to the receiver. So, the unreliability of open-loop X10 radio *adds* to the unreliability of X10 powerline signaling, making the overall system even more fragile.

Despite X10's drawbacks, since 1992 SmartLabs has been a major distributor of devices that use X10 signaling. In 1997, aware of the reliability problems its

customers were having with X10 devices available at the time, SmartLabs independently developed and began manufacturing its own *Linc* series of improved X10 devices, including controllers, dimmers, switches, computer interfaces and signal boosters. Despite the enhanced performance enjoyed by Linc products, it was still mostly do-it-yourselfers and hobbyists who were buying and installing them.

Through direct experience with customers around the world, SmartLabs concluded that X10 signaling would have to be *replaced* by a better protocol in order for home control to meet the needs and expectations of the mass consumer market. X10's low cost and simplicity were wonderful features, but the new standard, INSTEON, would have to be much more reliable and flexible.

To overcome the pitfalls of X10, SmartLabs designed INSTEON with five major differences:

1. INSTEON devices are simulcasting repeaters¹,
2. All INSTEON devices are two-way, with verified communications,
3. INSTEON powerline and radio signaling back each other up.
4. INSTEON is much faster than X10, and
5. INSTEON has a vastly expanded address and command space.

INSTEON Devices Are Simulcasting Repeaters

Because INSTEON devices simultaneously repeat¹ each other's messages during the same zero crossing intervals, adding more devices to an INSTEON network adds more energy to the INSTEON signal. Thus, as an INSTEON network grows, it becomes more robust and reliable. INSTEON is a mesh network, with multiple pathways for any given message to travel, but thanks to simulcasting, INSTEON messages travel over *all possible* paths without requiring a network controller, routers, or complicated routing software.

INSTEON Is Closed Loop

All INSTEON devices can both listen and talk, and the INSTEON protocol requires that all INSTEON messages that are not broadcast be acknowledged. So, INSTEON devices know when the messages they send to another device got through, and in the rare cases that they don't the message will automatically be retried. Likewise, INSTEON controllers can verify that controlled devices actually did what they were instructed to do, thus providing feedback to users that what they wanted to happen, happened.

INSTEON Uses Both Powerline and Radio Communications

No communication medium is perfect. If devices can only communicate over the powerline, or if they can only communicate by radio, system designers must either tolerate poor performance or compensate for the single point of failure by using more complex designs. Because INSTEON powerline messages are repeated using radio, and radio messages are repeated on the powerline, INSTEON messaging is highly reliable while still using simple signaling methods on both the powerline and the radio.

Like X10, INSTEON uses simple, low-cost narrowband signaling on the powerline. In fact INSTEON's carrier frequency, 131.65 KHz, is very close to X10's 120 KHz. INSTEON does not require more sophisticated and costly technologies like DSSS (direct sequence spread spectrum) or OFDM (orthogonal frequency division

multiplexing) to achieve vastly greater reliability than X10, because it uses dual mesh simulcasting instead. Mass-market devices like light switches can therefore be manufactured and sold at commodity prices.

Another significant advantage that comes from using both radio and powerline communication is that radio solves the powerline phase coupling problem. By installing at least one INSTEON device that can talk over both radio and the powerline on each powerline phase, the INSTEON radio signal will automatically bridge the powerline phase gap without any modifications to the house wiring.

INSTEON Is Faster than X10

INSTEON uses BPSK (binary phase shift keying) to continuously modulate the narrowband powerline carrier, instead of X10's OOK (on/off keying), where a carrier burst is either present or not present. At each powerline zero crossing (every 8.33 milliseconds), INSTEON sends an entire 24-bit packet of information, while X10 sends only $\frac{1}{2}$ bit of information. Thus INSTEON's raw signaling rate is faster than X10's by a factor of 48.

BPSK is faster and more reliable than OOK for several reasons. First, INSTEON devices use a special kind of BPSK receiver, called a Costas loop, which is very effective at digging faint signals out of noisy channels. Severe attenuation and noise spikes that are much larger than the INSTEON signal do not appreciably affect the Costas loop's ability to reliably demodulate the data in an INSTEON packet. In contrast, X10 receivers often consist of a broadly tuned filter followed by a threshold detector. Not only is the sensitivity of such a receiver low, but noise spikes easily pass through, so it is not surprising that X10 is susceptible to both false negatives and false positives.

In an attempt to increase reliability, X10 uses Manchester coding, where a one symbol consists of a carrier burst followed by no burst, and a zero symbol consists of no burst followed by a carrier burst. Manchester coding is the reason that X10 only sends $\frac{1}{2}$ bit of information per zero crossing. However, X10 also specifies that whole commands be sent twice, thus halving the throughput again.

As explained above (see [How X10 Works](#)), it takes almost one second (833 milliseconds, to be exact) for X10 to send one of 16 possible commands, to turn on a light, say. In contrast, INSTEON can send one of 65,536 commands in under 50 milliseconds, or about 16 times faster. Note, however, that the INSTEON message sender could specify up to 3 additional retransmissions for the message, for a worst-case allowance of 200 milliseconds for the command plus another 200 milliseconds for the acknowledgement from the message recipient. Even in the worst case, the INSTEON light will come on with no noticeable delay.

INSTEON Has a Large Address and Command Space

INSTEON devices come from the factory with a 3-byte (24-bit) ID number burned into nonvolatile memory. This ID number serves as a unique one of 16,777,216 possible INSTEON network addresses, but because the number is preassigned, users never have to deal with code wheels or network enrollment issues. To link an INSTEON controller device to an INSTEON responder device, users merely hold down buttons on each device for ten seconds, or else they can optionally use software running on a PC or the like. With over 16 million possible devices that users can link to, there is virtually no limit to the sophistication of possible control scenarios.

Furthermore, INSTEON allows 65,536 basic commands, with a mechanism for increasing the command set by using extended messages. The same extended messages allow for encryption in order to support devices such as door locks and security panels.

With room for more than 16 million addresses and 65 thousand basic commands, INSTEON is clearly in a different league compared to X10 with only 256 addresses and 16 commands.

INSTEON versus X10 Comparison Table

The following table summarizes the differences between INSTEON and X10.

Property	INSTEON	X10
Media	Powerline and Radio	Powerline Only
Module Types	All are peers	Controllers Controlled Devices Controlled Devices with Status
Message Propagation	Simulcasting Repeaters ¹	Single Transmitter Optional Signal Boosters
RF Data Rate, bps	38.4K Instantaneous at 904 MHz	Custom UHF Handsets
RF PHY	FSK in ISM Band	OOK in UHF band
Powerline Data Rate, bps	13,165 Instantaneous 2,880 Sustained 1,440 Standard Msg Payload 1,698 Extended Msg Payload 1,034 User Data	120 Instantaneous 60 Sustained 24 Payload (20 bits in 833 ms)
Powerline PHY	131.65 KHz Carrier BPSK	120 KHz Carrier OOK
X10 Compatible ²	Yes	Yes
Powerline Phase Bridging	RF or Hardware	Hardware
Acknowledged	Yes	No
Addressing	24-bit pre-assigned module ID (16,777,216 Devices)	16 House Codes 16 Unit Codes (256 devices per powerline)
Network Enrollment	Automatic	Code wheels, some EEPROMs
Groups	Databases in modules	Scenes in some controllers
Commands	2-byte (65,536)	16 Total
Device Types	3-byte (16,777,216)	None
Global Clock	Powerline Zero Crossing	Powerline Zero Crossing
Retries	Yes	No
Message Length	15, 33 bytes	100 bits (8.5 bytes)
Collision Avoidance	Yes	No
Triac Control	Yes	Yes
SRAM	256 Bytes	?
Flash	4K x 14 bits (7K Bytes)	?
EEPROM External	4K Bytes	None
EEPROM Internal	256 Bytes	None

Property	INSTEON	X10
Watchdog	Yes	No
Application Development	SALad Language in EEPROM SALad sIDE SmartLabs Device Manager, (DirectX and HTTP)	Devices not programmable

UPB

UPB (Universal Powerline Bus) is a low-speed powerline-only network designed by PCS (Powerline Control Systems, Inc., www.pcslighting.com), of Northridge, California.

UPB devices send messages over the powerline by coding data in the timing of high amplitude pulses. Normal UPB devices like light switches do not repeat or amplify messages sent by other UPB devices, so all of the signal power must come from the UPB transmitting device. Because UPB only communicates via the powerline, if for any reason powerline messaging fails, UPB devices can only retry and hope for the best. Unfortunately, electrical noise from many types of lamp dimmers and other electrical equipment can easily masquerade as a UPB signal pulse, causing interference that can last indefinitely. For a case study see [APPENDIX I — UPB Powerline Noise Immunity Evaluation](#).

INSTEON, on the other hand, is a *dual* mesh network, using both radio and powerline to back each other up. Moreover, INSTEON devices repeat¹ each other's messages by *simulcasting* them in precise timeslots synchronized to the powerline zero crossings. With each new device adding to the signal strength, an INSTEON network grows in reliability as more devices are added.

There is no integrated UPB chip available. Instead, the UPB protocol is implemented on general-purpose microcontrollers with discrete external components. Product development entails writing application code that runs on the microcontroller.

PCS has two U.S. patents issued. Number 6,734,784, granted May 11, 2004, is titled *Zero Crossing Based Powerline Pulse Position Modulated Communication System*. Number 6,784,790, granted August 31, 2004, is titled *Synchronization/Reference Pulse-Based Powerline Pulse Position Modulated Communication System*.

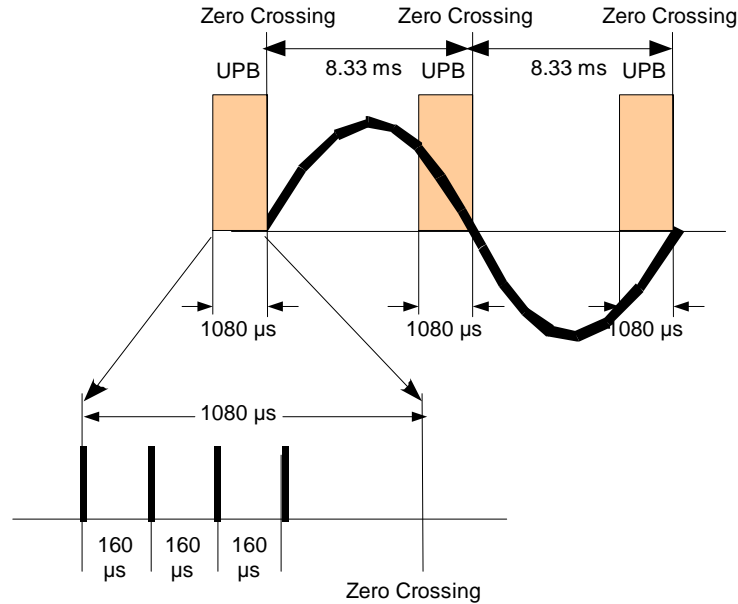
How UPB Works

UPB sends two bits of information in the millisecond before each powerline zero crossing, which gives a fixed data rate of two bits every 8.33 milliseconds, or 480 bits per second. The value of the two bits is coded by sending a large signal pulse at one of four possible time positions. The signal pulse, nominally 40 volts in amplitude, looks somewhat like a noise spike on the powerline, similar to the spikes created by triac lamp dimmers or electric motors. Because such noise spikes are indistinguishable from a UPB signal pulse, it is possible for UPB to be jammed by electrical noise that occurs in its time window.

UPB Physical Layer

UPB transmitters typically generate the 40-volt signal pulse by discharging a capacitor into the powerline, which creates a broadband ringing waveform. With filtering, the spectrum of the pulse is kept between 4 and 40 KHz. Because of the large amount of energy in the pulses, some devices emit an audible buzzing noise when they send UPB pulse trains.

The following diagram shows three powerline zero crossings with UPB signaling windows occurring in the 1080 microseconds preceding each zero crossing.



UPB uses PPM (pulse position modulation) to encode a dibit (two bits) in each UPB signal window. At the bottom of the figure the four possible positions of the UPB signal pulse are shown. If the pulse occurs in the leftmost position, the dibit value is 00. The next position to the right encodes 01, then 10, and finally 11. UPB specifies a tolerance of ± 40 microseconds for the position of any one pulse.

Unfortunately for this kind of signaling, triac lamp dimmers create powerline noise spikes when they switch on. These spikes look a lot like UPB pulses. Triacs dim lamps by switching on later and later during a powerline half cycle and then turning off at the next zero crossing—the later the triac switches on, the dimmer the lamp. To keep the triac switching transient well outside the UPB signal window, UPB dimming devices do not allow their own internal triacs to dim past 1440 microseconds prior to the zero crossing, which means that the dimmest setting of UPB-controlled lamps is still fairly bright. Most other manual and automated dimmers on the market, however, can go substantially dimmer, typically down to 880 microseconds, which puts the triac transient well within the UPB window. When this happens, the UPB signal will remain jammed until the offending dimmer is brightened.

UPB Message Structure

UPB messages are variable length, containing from 8 to 26 bytes, as shown in the following table.

Message Field	Bytes	Bit #	Description
Preamble (sync)	1		
Control	2	15	Link
		14-13	Repeater Request (0,1,2,4) For Repeater devices only
		12-8	Length (6-24)
		7	Not Used
		6-4	ACK Request (ACK Pulse, ID Pulse, Message)
		3-2	Total Repeats (Only senders repeat)
		1-0	Current Repeat (Only senders repeat)
Network ID	1		
Destination ID	1		
Source ID	1		
Message Data ID	1	7-5	Message Set ID 000 UPB Core Commands 001 Device Control Commands 010 Reserved 011 Reserved 100 UPB Core Reports 101 Reserved 110 Reserved 111 Extended Message Set
		4-0	Message ID or Extended Set ID
Message Data Arguments	0-18		
Checksum	1		

UPB Protocol

The raw bitrate of UPB signaling is 480 bps (bits per second), with two bits being sent every 8.33 milliseconds. Therefore, it takes 133 milliseconds to send an 8-byte message, and 16.625 milliseconds for each additional byte, up to 433 milliseconds for a maximum-length 26-byte message.

Each UPB message contains zero to 18 bytes of payload data, plus 8 bytes of message overhead. The payload data rate is therefore zero to 332 bps.

Communication is two-way, with error detection made possible by the checksum at the end of a message and message acknowledgement.

The UPB protocol allows for 250 House Addresses, 255 Device Addresses, and 254 Link Addresses. A four-digit password is used for security during setup procedures.

UPB Network Setup

Setting up a network of UPB devices requires special tools. PCS sells a Contractor Setup Device for this purpose, and also offers an RS232 Serial Interface device that includes setup software and 'Explorer' software.

INSTEON Compatibility with UPB

INSTEON on the powerline and UPB do not interfere with one another, in short because INSTEON is narrowband and UPB is broadband.

INSTEON is designed to be highly tolerant of the kind of broadband noise spikes that UPB uses for signaling, because that type of noise is very common in the powerline environment. Even though the last part of UPB's signal window overlaps the beginning of INSTEON's by 200 microseconds or more, noise spikes in this region do not materially affect the Costas loop receivers that INSTEON devices use to demodulate BPSK INSTEON signals.

UPB devices are insensitive to INSTEON signals because INSTEON appears as a low amplitude carrier at 131.65 KHz, well below UPB's threshold for detecting its own signal spikes.

INSTEON Compared with UPB

The most notable differences between INSTEON and UPB are that

1. INSTEON is dual mesh, UPB is powerline only;
2. INSTEON repeats¹ messages by simulcasting, UPB does not repeat messages;
3. INSTEON cannot be jammed by triac transients; and
4. INSTEON device setup does not require special tools.

INSTEON Uses Both Powerline and Radio Communications

As explained above (see [INSTEON Is a Dual Mesh Network](#) under [INSTEON Comparisons](#) and [INSTEON Uses Both Powerline and Radio Communications](#) under [INSTEON Compared with X10](#)), the fact that INSTEON is dual mesh adds greatly to the reliability and flexibility of an INSTEON network. With INSTEON, if powerline messaging fails, radio provides a backup, and vice-versa. But with UPB, there is no recourse except for retries.

Furthermore, it is not clear how UPB integrates with handheld controllers or other battery-operated devices not connected to the powerline. Without a common messaging protocol across media, UPB would have to rely on ad-hoc bridging devices, presumably implemented differently by different vendors.

INSTEON Repeats Messages by Simulcasting

Because INSTEON devices synchronously repeat¹ messages on the powerline by simulcasting them, the strength of the INSTEON signal increases as more devices join the network. (See [INSTEON Simulcasts Messages](#) in the [INSTEON Comparisons](#) section above for more about simulcasting.)

UPB, on the other hand, relies solely on the high amplitude of its signal spikes to propagate messages throughout a home, and, they claim, even across powerline phases. However, PCS does make and sell special UPB repeater devices, so UPB signal boosting is presumably necessary in some environments.

Triac Transients Do Not Jam INSTEON

INSTEON powerline signaling is designed to be immune to most impairments found on the powerline. INSTEON devices use Costas loop BPSK receivers that can recover INSTEON signals even when they are buried in noise and assaulted by transients like

those caused by triac firing, electric motors or UPB signaling. In contrast, as reported in [APPENDIX I – UPB Powerline Noise Immunity Evaluation](#), UPB signaling can be stopped in its tracks by devices as simple as manual lamp dimmers from Home Depot.

INSTEON Is Easy to Set Up

The UPB devices that we evaluated require special equipment and software to install and set up. In contrast, two of the design pillars of INSTEON are *Easy to Install* and *Simple to Use*. Because INSTEON devices all have unique ID numbers assigned at the factory, network installation is completely unnecessary. All that users need do is to link buttons on controllers to functions on controlled devices. INSTEON provides a simple *Plug-n-Tap™* method that works the same way for most devices, so that users can easily set up control associations all by themselves. Of course, it is also possible to perform setup using a PC, so INSTEON offers the best of both worlds.

INSTEON versus UPB Comparison Table

The following table summarizes the differences between INSTEON and UPB.

Property	INSTEON	UPB
Media	Powerline and Radio	Powerline Only
Module Types	All are peers	Most devices are peers Repeaters Network installer tools needed
Message Propagation	Simulcasting Repeaters ¹	Single Transmitter Optional Signal Boosters
RF Data Rate, bps	38.4K Instantaneous at 904 MHz	None
RF PHY	FSK in ISM Band	None
Powerline Data Rate, bps	13,165 Instantaneous 2,880 Sustained 1,440 Standard Msg Payload 1,698 Extended Msg Payload 1,034 User Data	480 Instantaneous 480 Sustained 0 (0-byte payload) 332 (18-byte payload)
Powerline PHY	131.65 KHz Carrier BPSK	40 V spike, 1 per zero crossing PPM
X10 Compatible ²	Yes	No, but coexists
Powerline Phase Bridging	RF or Hardware	Hardware
Acknowledged	Yes	Yes
Addressing	24-bit pre-assigned module ID (16,777,216 Devices)	250 House Addresses 255 Device Addresses 254 Link Addresses
Network Enrollment	Automatic	Special Installation Tools
Groups	Databases in modules	?
Commands	2-byte (65,536)	Variable
Device Types	3-byte (16,777,216)	Variable
Global Clock	Powerline Zero Crossing	Powerline Zero Crossing
Retries	Yes	Yes
Message Length	15, 33 bytes	8 to 26 bytes
Collision Avoidance	Yes	Yes

Property	INSTEON	UPB
Triac Control	Yes	Yes
SRAM	256 Bytes	?
Flash	4K x 14 bits (7K Bytes)	?
EEPROM External	4K Bytes	?
EEPROM Internal	256 Bytes	?
Watchdog	Yes	?
Application Development	SALad Language in EEPROM SALad sIDE SmartLabs Device Manager, (DirectX and HTTP)	Custom microcontroller code

Other Powerline Networks

This section compares INSTEON with [LonWorks](#), [HomePlug](#), [Intellon](#), and [CEBus](#).

LonWorks

LonWorks is a networking technology developed in 1988 by Echelon, Inc., (www.echelon.com), a public company (NASDAQ: ELON) headquartered in San Jose, California. LonWorks is mostly deployed in building and factory automation, commercial control, and meter reading applications.

About LonWorks

LonWorks is a sophisticated, high-performance routed network that uses special loop-free (learning) routers and repeaters to reliably deliver messages. The LonWorks stack implements the full seven-layer OSI (Open Systems Interconnection) model. At the PHY (physical) layer, LonWorks can talk over twisted pair, coax, fiber optics, powerline, infrared, and radio, with the majority of actual installations using dedicated twisted pair wiring.

OSI Layer	Layer Name	LonWorks Software Services
1	Physical	TP, CX, FO, PL, IR, RF
2	Link	MAC: predictive CSMA, CA; Optional CD, priority
3	Network	Connectionless, domain-wide broadcast, loop-free, available learning routers
4	Transport	ACK/NAK multicast, unicast, authentication server; transaction control (ordering, duplicate detection)
5	Session	Sets up, coordinates, terminates application communication
6	Presentation	Syntax conversion
7	Application	QoS, authentication, privacy

The LonWorks platform, intended as a BACnet (Building Automation Control network) replacement, uses a protocol called *LonTalk*, formalized as ANSI/CEA 709.1, and IEEE 1473-L standards. LonTalk packets can contain a large amount of information, as shown in the table below.

Field	Bytes
Link Header	1
Address Information	4
Service Type	1
Session Header	2
Presentation Header	2
Data	2 to 218
CRC	2

Echelon offers *Neuron* chips to manufacturers for creating products, but the cost of the chips (under \$10) would have to be substantially lower to permit mass-market consumer devices in the home.

For product developers, Echelon offers a *Node Builder* package for \$5,995, and a *Mini PL EVK* for \$395 for powerline applications. Programming of Neuron chips is done in *Neuron C*.

LonWorks system installation is normally performed by professionals using a *LonPoint Network Integration Package*, which includes hardware control interfaces, *LonMaker Windows* software for configuration and TCP/IP communications, and *LonWorks Network Services* for connecting to the Internet or an intranet. Recently, however, Echelon has announced *Interoperable Self Installation* software to simplify the network setup procedure.

INSTEON Compared with LonWorks

INSTEON is optimized for home control, while LonWorks has found applications mostly in commercial and industrial settings. LonWorks is reliable, fast, and flexible, but this sophistication comes at a price. Although LonWorks has been around since 1988, its price-performance point has so far proven to be too high for widespread adoption in the consumer home control market.

INSTEON, on the other hand, has been architected from the ground up to meet consumer expectations for simplicity, affordability and reliability. Because it is a modern design, INSTEON is not bound with any legacy issues, but instead can take advantage of the most recent advances in technology. In the near term at least, INSTEON and LonWorks networks will likely remain in different market segments, although they could be interconnected with appropriate bridging devices if a marketplace demand emerges.

HomePlug

The HomePlug Powerline Alliance (www.homeplug.org) has released two standards for high-speed, broadband information transport over the powerline.

About HomePlug

HomePlug 1.0, based on Intellon *PowerPacket* technology, can transport data over the powerline at a bitrate of 14 Mbps (megabits per second). Using state-of-the-art techniques (see the [Intellon](#) section below), this protocol demonstrates that the powerline is capable of supporting high-performance broadband networking. Consumers, however, have adopted WiFi radio (IEEE 802.11) much more enthusiastically for wireless computer networking, with WiFi already being used in 17% of U.S. homes. The next WiFi standard, 802.11n, will operate at a minimum speed of 100 Mbps.

HomePlug AV, which is even more advanced with a raw data rate of 200 Mbps, is capable of transporting multimedia and HDTV streams over the powerline. Despite successful demonstrations of the technology, it is too early to tell if HomePlug AV will be widely adopted by consumers, or if a radio technology such as UWB or WiFi will find greater acceptance.

Recently the HomePlug Powerline Alliance has released a request for proposals for *HomePlug Command and Control (HPCC)*, which is intended as a low-speed powerline network with design goals similar to those of INSTEON. Over the next year or so, HomePlug will conduct a 'bakeoff' among those of its Alliance members who submit proposals. SmartLabs has elected not to submit INSTEON to HomePlug, concentrating instead on the adoption of INSTEON as a de facto standard through broad consumer marketing.

Companies who wish to contribute intellectual property to HomePlug's powerline specifications must be *Participant Members* in the HomePlug Powerline Alliance, at a cost of \$15,000 per year. Access to the specifications requires *Adopter Membership* for \$5,000 annually.

INSTEON Compared with HomePlug

INSTEON is a home command and control network, while HomePlug uses the powerline for advanced computer networking and media transport, so there is no overlap in the marketplace. On the powerline itself, the two signals do not interfere with each other, so there is no problem with coexistence.

HomePlug's Command and Control initiative will not result in actual products for at least a year, so it is too early to tell what influence a new proposed single mesh powerline standard would have in the marketplace of the future.

Intellon

Intellon (www.intellon.com), of Ocala, Florida, got its start in 1989 by developing the winning designs for adoption as the CEBus powerline and RF signaling standards (see the next section, [CEBus](#) for more information). Although CEBus did not find acceptance in the marketplace, Intellon went on to improve its spread spectrum technology to the point where over a hundred megabits per second could be transported reliably over the powerline.

About Intellon Technology

Intellon's broadband technology, called *PowerPacket*, is at the heart of the HomePlug 1.0 broadband powerline networking specification (see the [HomePlug](#) section above). As a founding sponsor and member of the board of directors of the HomePlug Powerline Alliance, Intellon is a primary driver of the even-more-advanced HomePlug AV technology.

As one might expect, PowerPacket is very sophisticated and complex to be able to transport so much data reliably in such a hostile environment.

At the physical (PHY) level, PowerPacket uses multiple carrier OFDM (orthogonal frequency division multiplexing, also used by WiFi and DSL), with interleaving and both Viterbi and Reed-Solomon forward error correction (FEC). The control software dynamically negotiates an optimum payload data rate by choosing carriers, switching between DQPSK and DBPSK modulation, and varying the FEC rate. The signal is spread from 4.5 MHz to 21 MHz on the powerline, with a digitally filtered PSD (power spectral density) to match HomePlug's requirements.

The media access control (MAC) layer uses CSMA (carrier sense multiple access) with a randomized contention window and exponential backoff, like Ethernet. Other MAC services include priority resolution, ACK/NAK (acknowledgements), ARQ (automatic repeat requests), segmentation of slow frames for QoS (quality of service), and 56-bit DES (digital encryption standard) encryption using cipher block chaining.

This technology is clearly a *tour de force*. Highly advanced, it is well suited for its intended broadband applications, but it is far too expensive to find its way into devices like light switches.

INSTEON Compared with Intellon Technology

Intellon's broadband technology does not compete with INSTEON as explained above in the section [INSTEON Compared with HomePlug](#). Intellon does still offer legacy CEBus-inspired SSC (spread spectrum carrier) chips, but the marketplace has not adopted CEBus for reasons explained in the [CEBus](#) section below, so the chips are mainly used in proprietary building automation and vehicle brake control systems. Consequently, INSTEON does not overlap Intellon's technology in the electronic home improvement marketplace, although both signals can coexist on the powerline without interference.

CEBus

In 1984, the EIA (Electronic Industries Association, now called CEA, for Consumer Electronics Association) formed a Technical Steering Committee to develop an ambitious standard for interconnection of all kinds of devices in the home. Called CEBus, for Consumer Electronics Bus, this comprehensive effort strove to unify communications over coax, twisted pair, powerline, radio, infrared, audio/video, and fiber optic media with a command language called CAL (Common Application Language).

About CEBus

The CEBus committee formed working groups for each of the seven different media, and the working groups then conducted competitions among industry proponents to demonstrate technologies for adoption as signaling standards on the media. Intellon won the competition for powerline and radio with a spread spectrum technology that was very innovative for the time.

On the powerline, the Intellon technology used a custom designed 'chirp' signal that occupied spectrum between 100 KHz and 400 KHz. Virtually the same idea was adopted for radio, but spread about a carrier in the unlicensed 900 MHz ISM (industrial, scientific and medical) band.

After the working group 'bakeoffs', the EIA membership ratified the CEBus standard as EIA-600 in 1994 and published it as a stack of documents almost a foot high.

Although Intellon did offer chips implementing the powerline and radio parts of the standards, devices built with them were complex and expensive. One dimmer switch used eleven integrated circuits on two circuit boards, far too complex and expensive to meet mass-market consumer needs.

Major industry players did not embrace CEBus, so it never gathered the momentum necessary to become adopted in the marketplace. Over-complexity and over-reaching by the committee of engineers were likely reasons for its demise, but timing was probably another, since the Internet was virtually unknown in those days and relatively few homes had PCs.

INSTEON Compared with CEBus

Much has happened since CEBus failed to gain traction in the mid-1990s. Most U.S. homes have computers, most of those are connected to the Internet, and a growing number are wirelessly networked using WiFi. With cellphones, satellites, DSL, and broadband cable, much of what CEBus set about interconnecting is now interconnected by other means. Nevertheless, it is still true that infrastructure devices like light switches, security sensors, and door locks have yet to be networked. The failure of CEBus to achieve that goal underscores the importance of simplicity and affordability in a consumer environment.

INSTEON is not only simple and affordable—it is also reliable. Consumers can start out controlling just a few lights, and then add incrementally to their network of INSTEON devices. Whenever they and the marketplace are ready, consumers can interconnect their INSTEON-networked devices with their PCs, the Internet, their cellphones, their A/V gear, and whatever else that comes along. INSTEON makes sense now, and it will bring even more benefits as other technologies advance.

Radio Networks

This section compares INSTEON to [ZigBee](#), [Z-Wave](#), and a number of [Other Radio Networks](#), including [WiFi](#), [Bluetooth](#), [WiMax](#), [UWB](#), and [UHF](#).

ZigBee

ZigBee is a routed radio networking standard developed by a group of ZigBee Alliance members (www.zigbee.org). Ratified in December 2004 and released to the public in June 2005, the ZigBee specification defines the Application/Framework and Network/Security layers built atop the pre-existing IEEE 802.15.4 radio standard. ZigBee was designed to be used in widely diverse of applications, ranging from battery-operated devices to commercial and industrial building management (see [ZigBee Interoperability](#), below).

IEEE 802.15.4, which was completed in May 2003, defines a *Low-Rate Wireless Personal Area Network (WPAN)* that includes a direct sequence spread spectrum (DSSS) physical radio (PHY) layer and a media access control (MAC) software layer. Several chip vendors offer 802.14.4 radios that include microprocessors and 128K of onboard memory for the ZigBee stack. 802.15.4 radios also find uses in numerous non-ZigBee applications.

ZigBee's design goals were:

- A wireless network for industrial controls, medical devices, alarms, building automation, and home automation;
- A self-organizing mesh network;
- A low data rate; and
- Low power consumption (greater than one-year battery life).

ZigBee has defined multiple kinds of devices, including

- Network coordinators, one per network, at the root of the network tree;
- FFDs (full function devices), which can be routers; and
- RFDs (reduced function devices), which cannot be routers.

Only FFDs can form a mesh network, so ZigBee also defines a *star* network that can include RFDs at the edge of the network, and a hybrid network called a *cluster tree*. Within these networks there can be beaconing or non-beaconing in order to support battery-operated devices that wake up intermittently.

ZigBee 802.15.4 radios can operate at 2.4 GHz worldwide, or at 915 MHz in the U.S. and 868 MHz in Europe. With two frequencies in each market, consumers will have to choose devices that all use the same frequency, or else the devices will have to be frequency agile.

ZigBee product developers must be members of the ZigBee Alliance. An entry-level *Adopting Membership* is \$3,500 per year, but participation in ZigBee standards setting requires a *Participating Membership* for \$9,500 or a *Promoter Membership* for \$40,000 annually.

How ZigBee Works

ZigBee, so named because its messages zigzag around like a bee, is a routed radio-only network. ZigBee uses IEEE 802.15.4 spread spectrum radios, but with extensive additional software specified by the ZigBee Alliance.

IEEE 802.15.4 Radios

The IEEE 802.15.4 Low-Rate Wireless Personal Area Network (WPAN) standard, released in May 2003, defines the Physical (PHY) and Media Access Control (MAC) layers for Direct Sequence Spread Spectrum (DSSS) radios.

In a given market, these radios can operate at two distinct frequencies, 868 MHz and 2.4 GHz in Europe, and 915 MHz and 2.4 GHz in the U.S. and some other countries. The data rate varies depending on the frequency. In the table below, the payload data rates have been calculated assuming 15 bytes of overhead per packet, although in practice the non-secure overhead can vary from 15 to 35 bytes (see below).

Property	Value		
	868 MHz	915 MHz	2.4 GHz
Frequency	868 MHz	915 MHz	2.4 GHz
Region	Europe	U.S.	Worldwide
Channels	1	10	16
Modulation	BPSK	BPSK	O-QPSK
Bandwidth	600 KHz	1.2 MHz	2 MHz
Raw Data Rate	20 Kbps	40 Kbps	250 Kbps
9-byte payload data rate	8 Kbps	15 Kbps	94 Kbps
14-byte payload data rate	10 Kbps	19 Kbps	121 Kbps
23-byte payload data rate	12 Kbps	24 Kbps	151 Kbps

These radios have ranges from 10 to 75 meters, with 50 meters being typical. In the home environment, the lower frequencies will often propagate two or more times farther than 2.4 GHz because of less absorption by construction materials.

At the PHY level, 802.15.4 radios can also perform receiver energy detection, link quality indication and clear channel assessment.

The MAC layer supports 64,000 nodes per network. All devices must have an 8-byte IEEE address, but 2-byte short addresses can optionally be allocated during specified network association and dissociation procedures.

802.15.4 defines both contention-free and contention-based channel access methods. For contention-free communications, an optional guaranteed timeslot (GTS) can be used for high priority messages, but the radios normally use CSMA-CD (carrier sense multiple access with collision detect) for contention-based messaging.

Packets are variable size, with a maximum length of 128 bytes, or 104 bytes maximum payload. For battery operation, 802.15.4 defines an optional superframe that allows low-power devices to wake up periodically.

Packet overhead (non-payload) data takes from 15 to 35 bytes, with an additional 7 bytes for optional security encryption, as shown in the following table.

Stack Layer	Field	Size, Bytes	Total Bytes
PHY	Preamble	4	6
	Start of Frame	1	
	Frame Length	1	
MAC	Frame Control	2	9 - 29
	Data Sequence Number	1	
	Address Information	4 – 20	
	Frame Check Sequence	2	
Security	Frame Count	4	7
	Sequence Count	1	
	Integrity Code	2	

ZigBee Software

The ZigBee Alliance has defined Application/Framework and Network/Security software layers atop the radio PHY and MAC layers. The ZigBee specification was ratified December 14, 2004 and made available to the public in June 2005.

The network layer supports star, mesh, and cluster-tree (hybrid star/mesh) network topologies. The mesh network is self-forming and self-healing with multiple routes for data. Network traffic can be intermittent, periodic, or guaranteed timeslot for repetitive low-latency data.

Multiple network topologies are the consequence of supporting multiple kinds of physical and logical devices defined by ZigBee in order to reduce cost for at least some devices.

There are two physical ZigBee device types—full function devices (FFDs) and reduced function devices (RFDs). Typically line-powered, FFDs can communicate with all other FFDs and RFDs, they can act as network and link coordinators, they can discover other FFDs and RFDs, and they can perform all RFD functions. RFDs are typically battery powered and they can only be installed in star networks because they can only talk to an FFD network coordinator. RFDs often go to sleep to reduce battery drain, and when they wake up they can determine if data is pending, request data from the network coordinator, transfer data from an application, or search for available networks.

ZigBee logical device types include network coordinators, routers and end devices. Network coordinators initialize a ZigBee network, manage network nodes, and store network node information. ZigBee routers transfer messages between paired nodes. ZigBee end devices can only be leaf nodes in a network because they do not participate in message routing.

The following table shows various software services that the ZigBee stack performs at various layers. This software is in addition to the software for managing the underlying 802.15.4 radios.

ZigBee Software Layer	Functions
MAC Support	Send/receive data Manage transaction queue Perform network association Synchronize devices Scan radio channels Get communication status Manage parameters Manage beacon Sync without beacon Manage guaranteed timeslots Reset Manage device orphans Turn radio on/off
Network	Start network Join and leave network Configure new device Assign addresses Synchronize within a network Manage security Manage routing
Application Support	Discovery Binding
Security	Manage access control lists Manage packet freshness timers Manage 128-bit AES encryption

ZigBee Interoperability

ZigBee has aimed for a broad universe of applications, including data monitoring and control, structured wiring replacement, battery-operated devices, commercial and industrial building management, and home automation.

To support these markets, ZigBee has formed a number of committees to develop public *Application Profiles* to define product properties required for different vendors to build interoperable devices. This is an intricate, ongoing task—the only ZigBee Application Profile currently defined is *Home Control, Lighting*. Accordingly, ZigBee allows vendors to create private profiles for their own purposes. The complete list of public Application Profiles that ZigBee is working on is:

- Automatic Meter Reading (AMR)
- Commercial Building Automation (CBA)
- Industrial plant monitoring (HPM)
- Home automation (HA)
- Heating, ventilation, air conditioning (HVAC)
- Home Control, Lighting (HCL)

ZigBee Certification

To earn certification for their products, vendors must be *Adopting Members* of the ZigBee Alliance, at a cost of \$3,500 per year. There are three levels of certification:

- ZigBee-Compliant Platform, for chipsets or module
- ZigBee Friendly, for devices using private Application Profiles
- ZigBee-Logo Certified, for devices using public Application Profiles

ZigBee product certification costs \$1,000 for the first product SKU (stock-keeping unit) and \$500 for each SKU thereafter.

INSTEON Compatibility with ZigBee

INSTEON radio can coexist with ZigBee because INSTEON is narrowband at 904 MHz while ZigBee only occupies 1.5 MHz at 915 MHz. Even if ZigBee did overlap INSTEON, there would be no problem because ZigBee is direct sequence spread spectrum (DSSS). DSSS looks like noise to INSTEON receivers, and INSTEON looks like a narrow signal at one frequency to ZigBee. 2.4 GHz devices like 802.15.4 ZigBee radios are invisible to INSTEON, and vice-versa.

INSTEON Compared with ZigBee

The ZigBee specification was only released to the public in June 2005, so there are not yet any ZigBee home control devices to evaluate head-to-head with comparable INSTEON devices. However, there are still several noteworthy differences between INSTEON and ZigBee that will have consequences in the home control marketplace. The main differences are that:

1. INSTEON is dual mesh, ZigBee is radio only;
2. INSTEON propagates messages by simulcasting, ZigBee routes messages;
3. INSTEON devices are peers, ZigBee has full-function devices and reduced-function devices;
4. INSTEON requires no separate network enrollment step;
5. INSTEON uses FSK radios in one band per market, ZigBee uses DSSS in two bands per market; and

Taken together, these distinctions entail higher cost and complexity for ZigBee radio. Granting that the elevated cost buys ZigBee radio devices with somewhat higher performance than INSTEON radios, INSTEON's dual mesh architecture provides a powerline backup that ZigBee can't rely on when its radio-only messaging does fail.

On balance, ZigBee's cost and complexity make it better suited for large industrial and commercial networks, whereas INSTEON is optimized for the home.

INSTEON Uses Both Powerline and Radio Communications

ZigBee's 802.15.4 radios, although sophisticated, are not perfect—sometimes a ZigBee radio message will not get through, and then the only recourse is to retry. INSTEON, however, has backup—the powerline can go where radio may be blocked. This means that INSTEON radios can be simpler and lower cost than ZigBee radios, and the software can occupy a smaller footprint while running on ultra-low-cost microcontrollers.

INSTEON Uses Simulcasting Instead of Routing

Conventional wisdom has it that network protocols must prevent messages from clobbering one another at all costs, with the result that system designers often choose message routing as the best way to achieve a well-behaved network. While this may be true for high-speed networks, at the modest data rates INSTEON requires for home command and control, more than one copy of a message can be sent in the same synchronized timeslot. Simulcasting of messages avoids any need for network controllers and routers, and it has the added benefit that as more devices simulcast a message, more energy goes into sending that message, so the more likely it is for the message to get through.

INSTEON Is a Peer-to-Peer Network

Because of the complexity of the ZigBee software stack, chips that contain it along with 802.15.4 radios typically have 128K bytes of onboard memory and a relatively high-performance microprocessor core. In an attempt to reduce the overall cost of a network of ZigBee devices, ZigBee has defined Full-Function Devices (FFDs) and Reduced-Function Devices (RFDs), with multiple stack protocols and options to support the different kinds of devices. Each ZigBee network must have at least one FFD to act as a network coordinator. Any RFDs must be at the perimeter of the network defined by the network coordinator, and there is some danger that such leaf RFDs will reconnect to a different coordinator after signal loss.

INSTEON, thanks to the simplicity of simulcasting, does not have any of these problem. The full INSTEON engine, which is the same for *every* INSTEON device, takes up less than 3K bytes, leaving over 5K bytes for application code in an 8K byte microcontroller, more than enough for lamp dimmers and simple controllers.

INSTEON Requires No Network Enrollment

With a unique ID number assigned to every device during manufacture, INSTEON devices automatically join an INSTEON network and help with simulcasting repeated messages as soon as they are powered up. With no need for a separate network enrollment step, users only need to link button pushes on a controller with functions on a controlled device. This straightforward setup can be accomplished by anyone using a simple *Plug-n-Tap™* procedure or optionally using software.

INSTEON Radio Is Simpler

INSTEON uses proven FSK (frequency shift keying) radios at 904 MHz, while ZigBee uses complex DSSS (direct sequence spread spectrum) radios in two bands per market. FSK receivers are more than twice as sensitive as DSSS radios while only requiring one-fourth the silicon.

Even though ZigBee is not the only protocol used by 802.15.4 radios on the market, most chip vendors are currently only offering 2.4 GHz radios containing the ZigBee stack. It is not clear what will eventually happen at 868 MHz in Europe and 915 MHz in the U.S. Since low-band and high-band ZigBee devices will not be able to hear each other, consumers will have to be watchful when they buy new ZigBee devices to expand their networks. If they inadvertently choose the wrong-frequency ZigBee devices, they will have to return them.

INSTEON uses a single frequency of 904 MHz, so there can be no such consumer confusion. This choice has two additional advantages. 904 MHz propagates farther than 2.4 GHz in typical home environments, and WiFi, which already occupies the

2.4 GHz band at higher power, cannot interfere with INSTEON, although it might cause problems with ZigBee.

INSTEON versus ZigBee Comparison Table

The following table summarizes the differences between INSTEON and ZigBee.

Property	INSTEON	ZigBee
Media	Powerline and Radio	Radio Only
Module Types	All are peers	Network Coordinator (1 per network) (FFD) Full-Function Device (RFD) Reduced-Function Device
Message Propagation	Simulcasting Repeaters ¹	Routing Tables
RF Data Rate, bps	38.4K Instantaneous at 904 MHz	20K Instantaneous at 868 MHz 40K Instantaneous at 915 MHz 250K Instantaneous at 2.4 GHz
RF PHY	FSK in ISM Band	DSSS in two ISM Bands
Powerline Data Rate, bps	13,165 Instantaneous PL 2,880 Sustained PL 1,440 Standard Msg Payload 1,698 Extended Msg Payload 1,034 User Data	None
Powerline PHY	131.65 KHz Carrier BPSK	None
X10 Compatible ²	Yes	N/A
Powerline Phase Bridging	RF or Hardware	N/A
Acknowledged	Yes	Yes
Addressing	24-bit pre-assigned module ID (16,777,216 Devices)	64-bit IEEE Address 16-bit short address (65,536 Devices)
Network Enrollment	Automatic	Part of Installation Procedure
Groups	Databases in modules	?
Commands	2-byte (65,536)	Application Profiles
Device Types	3-byte (16,777,216)	Application Profiles
Global Clock	Powerline Zero Crossing	Optional beaconing
Retries	Yes	Yes
Message Length	15, 33 bytes	Variable, 128 byte maximum
Collision Avoidance	Yes	Yes
Triac Control	Yes	?
SRAM	256 Bytes	?
Flash	4K x 14 bits (7K Bytes)	128K Bytes typical
EEPROM External	4K Bytes	?
EEPROM Internal	256 Bytes	?
Watchdog	Yes	?

Property	INSTEON	ZigBee
Application Development	SALad Language in EEPROM SALad sIDE SmartLabs Device Manager, (DirectX and HTTP)	?

Z-Wave

Z-Wave is a routed radio-only network designed by ZenSys AS (www.zen-sys.com), from Denmark. Intended for wireless home control applications, Z-Wave has design goals similar in some respects to those of INSTEON:

- Low cost
- Low power
- Reliable
- Easy network installation
- Easy association process
- No ongoing network management
- Product interoperability

Z-Wave sends messages using radio alone. If for any reason radio messaging fails, Z-Wave can't rely on the powerline to get messages through. Therefore, Z-Wave's designers were obliged to give it a number of relatively complex features in order to make it as reliable as they could.

Most notably, Z-Wave routes its messages through the network using a Source Routing Algorithm (SRA). The SRA requires message initiator devices to know the arrangement of other devices in the network (the topology) so that they can compute the best route for messages to travel. Maintaining and distributing a network topology database is an intricate software task, especially when some devices in the network are mobile. Therefore, to keep costs down, Z-Wave defines different kinds of devices, with the lowest-cost devices, called slaves, unable to initiate messages.

In contrast, INSTEON is a *dual mesh* network, using both radio and powerline to back each other up. INSTEON does not route messages—it *simulcasts* them in precise timeslots synchronized to the powerline zero crossings. Simulcasting is so much simpler than routing that all INSTEON devices can be low-cost peers, with no need for complex master controllers or simplified reduced-function devices.

See [APPENDIX II — Z-Wave Usability Evaluation](#) for a case study of some of the consequences of Z-Wave's network architecture choices.

To create a Z-Wave-certified product, developers must be *Full Members* of the Z-Wave Alliance, which costs \$2,500 per year.

ZenSys is the owner of a U.S. Patent, number 6,879,806, issued April 12, 2005, titled *System and a Method for Building Routing Tables and for Routing Signals in an Automation System*, that covers some aspects of Z-Wave networking.

How Z-Wave Works

Intended for wireless home control applications, Z-Wave radio networking is designed for relatively few nodes (20 to 200) that communicate on average every 5 to 15 minutes. Z-Wave messages are variable length, with a payload averaging 4 to 6 bytes. Message latency requirements are relaxed to 200 milliseconds or more.

Z-Wave Physical Layer

Z-Wave radios use an unlicensed carrier frequency of 868.42 MHz in Europe and 908.42 MHz in the U.S. Data is modulated onto the carrier at 9600 bps using BFSK (binary frequency shift keying).

Z-Wave Messaging

The minimum length of a properly formatted Z-Wave message is 9 bytes, but a routed message requires 12 bytes plus repeater data plus the payload.

The message protocol includes routing, frame acknowledgement, collision avoidance with random backoff, and a frame checksum with retransmission if necessary.

The Z-Wave network is self-organizing and self-healing. To achieve self-organization, Z-Wave nodes have software that discovers the node's neighbors and informs the network's Static Update Controller (SUC) about them. A Source Routing Algorithm (SRA) in devices capable of initiating communication finds message pathways and generates routes based on a network topology database. Self-healing requires software to dynamically generate new routes around temporarily unavailable nodes. Moving nodes have software routines that can request new neighbor searches automatically. This software, which is part of the Z-Wave stack, resides in on-chip memory.

Z-Wave Network Setup

Since Z-Wave devices do not possess unique network addresses (ID numbers) when purchased, there must be a procedure whereby the network's Static Update Controller (SUC) assigns ID numbers to devices being installed on the network. Z-Wave specifies that network installation should be accomplished either centrally using some kind of installer tool, or locally using the devices themselves. However, as described in [APPENDIX II — Z-Wave Usability Evaluation](#) below, different Z-Wave device vendors have adopted different methods for achieving network installation locally.

Z-Wave Association Process

To support the creation of associations between buttons on a controller and actions of a controlled device, Z-Wave specifies that

- The network must provide an *Association Wizard*,
- There must be a *sanity check* of requested associations, and
- All nodes must be able to present their supported capabilities.

Software routines to support these features, to the extent that product developers choose to develop and include them, further enlarge memory requirements.

Z-Wave Chips

ZenSys offers Z-Wave chips with 32K bytes of flash memory and provision for additional external memory. The earlier 100 series chip is being replaced with the 200 series using a smaller die. Both chips have an integrated FSK radio and an 8051 processor core. ZenSys also offers complete RF modules built around the chips for inclusion in products.

Z-Wave Application Development

To create devices that use Z-Wave networking, product developers typically use a C compiler to write application firmware for downloading into the flash memory on the

chip. ZenSys provides a collection of API (application programmer interface) routines in a Windows DLL (dynamically linked library) to help with this task.

Z-Wave Interoperability

To promote product interoperability between devices from different manufacturers, Z-Wave maintains a list of standardized command definitions and device class specifications. Z-Wave devices must pass a certification procedure costing \$750 before they can display the Z-Wave logo. ZenSys recently announced a Compliance Test Tool written by ControlThink that Z-Wave Alliance members can use to evaluate their own devices prior to certification.

Despite these measures, similar Z-Wave devices from different manufacturers have different behaviors, as described in some detail in [APPENDIX II — Z-Wave Usability Evaluation](#) below.

Z-Wave Device Types

Managing the routing of messages on a network is complicated, especially if some devices in the network are mobile. Z-Wave uses source routing, so any device capable of initiating communication must know which routes are currently possible, choose the best route, and then embed the routing information into the messages that it sends. Such routing algorithms have been well developed for a variety of networks, but they require a lot of code. Lots of code means lots of memory on a chip, and therefore higher build costs for devices that use the chip.

Z-Wave chips have 32K bytes of flash memory, with provision for adding more memory externally. To reserve as much memory as possible for the application code in a Z-Wave device, the Z-Wave communications stack must be kept as small as possible. Therefore, Z-Wave defines a number of different device types that have varying capabilities and stack sizes. The main Z-Wave device categories are *Controllers*, *Routing Slaves*, and *Slaves*.

Z-Wave Controllers can initiate communication with all nodes, and so they have the largest stack. The master controller, called a SUC (static update controller), performs network management, distributes network topology information to secondary controllers, and supports central or local device enrollment. Mobile controllers use a *portable controller stack* that allows devices to request rediscovery of moving nodes. There can also be a SIS (SUC ID server), which can automatically distribute network topology information to multiple controllers, but that software usually runs on a PC. Unless there is a SIS in the Z-Wave network, users have to manually copy network topology data from the master controller to any secondary controllers in the network whenever they add or remove Z-Wave devices.

Routing Slaves can initiate communication with a subset of nodes using a smaller Z-Wave stack. They depend on the SUC for network topology information.

Slaves have the smallest stack and can only respond to communications.

Z-Wave also defines *Installers* for doing centralized network setup and *Bridge* devices for connecting to other kinds of networks.

By defining devices with reduced functionality in order to minimize cost, Z-Wave has given up the simplicity of peer-to-peer networking. This tradeoff is understandable in response to the complexity of routing, but because different devices have different capabilities, users have to know more about how the network functions. Perhaps the

most restrictive issue for users is the requirement for a single master controller in a Z-Wave network.

INSTEON Compatibility with Z-Wave

INSTEON and Z-Wave radios are invisible to each other because they both use narrowband FSK radios but on different frequencies. The situation is similar to two different FM radio stations at different points on the radio dial. In the U.S., Z-Wave radios are tuned to 908.42 MHz while INSTEON uses 903.991418 MHz.

In Europe Z-Wave uses 868.42 MHz. INSTEON's radio specification for Europe has not been finalized, although it will be chosen for minimum interference with all other signals in the band, including Z-Wave.

INSTEON Compared with Z-Wave

The main differences between INSTEON and Z-Wave are that

1. INSTEON is dual mesh, Z-Wave is radio only;
2. INSTEON propagates messages by simulcasting, Z-Wave routes messages;
3. INSTEON devices are peers, Z-Wave has network controllers and slaves; and
4. INSTEON requires no separate network enrollment step.

INSTEON Uses Both Powerline and Radio Communications

Radio communication is far from 100% reliable, as cellphone users are frequently reminded from firsthand experience. Z-Wave radios, because they use narrowband FSK (frequency shift keying) signaling, are considerably less sophisticated than cellphones. Furthermore, because they share the unlicensed 900 MHz ISM (industrial, scientific and medical) band, they are required to transmit at low power. Metal in the home can block or reflect radio waves—installing radio wall switches in metal junction boxes can cause particular difficulties.

That is why INSTEON is dual mesh—if radio fails, powerline provides a backup, and vice-versa. In combination, two independently simple, low-cost signaling methods can be much more reliable than sophisticated, high-cost methods employed on single media with no backup.

Because it is radio-only, Z-Wave had to resort to complex network self-organization, self-healing and routing procedures in an attempt to maximize reliability. INSTEON, on the other hand, even though it employs the same sort of narrowband FSK radios in the 900 MHz range as Z-Wave, achieves greater overall reliability with far less complication, because it relies on dual mesh powerline backup and repeated message simulcasting.

INSTEON Uses Simulcasting Instead of Routing

Simulcasting is much simpler than routing, and more robust because multiple devices simulcasting the same message add to the signal power. For a complete discussion of how simulcasting works see *INSTEON The Details*, available at www.insteon.net.

As explained above, routing entails complexity. Devices that initiate communication using a source routing algorithm, which is what Z-Wave uses, must know the topology of the network. Maintaining and distributing a network topology database

is not trivial, especially with mobile devices. To keep costs down, Z-Wave has defined different classes of devices, some of which cannot participate in routing.

INSTEON Is a Peer-to-Peer Network

INSTEON devices are two-way simulcasting repeaters¹, which means they all handle INSTEON messages in *exactly* the same way, with no need for network controllers or routers. A Z-Wave network, on the other hand, must contain a Static Update Controller (SUC) along with other kinds of devices, including slaves or routing slaves.

If you want more than one controller in your Z-Wave network, and you don't have a SIS (SUC ID Server), you must choose a single controller as a master and use that controller alone for network maintenance. To set up your other controllers as secondaries you must go through an involved controller replication procedure. Thereafter, any time you add or delete network devices using your single master controller, you must repeat the entire controller replication process for all of your other controllers.

In contrast, INSTEON lets you add or remove INSTEON devices of any kind at any time, because no matter how a device appears to the user, to the INSTEON network all devices are peers. Therefore, setting up INSTEON devices requires far less customer knowledge and involvement.

INSTEON Requires No Network Enrollment

As a consequence of routing using network controllers, Z-Wave requires that new devices be enrolled in the Z-Wave network before controller buttons can be associated with device functions. Some Z-Wave vendors have found ways to hide the network enrollment step from users by combining it with the control association procedure, but others have elected to keep the two procedures separate. This situation is confusing if consumers have Z-Wave products from different vendors who have implemented different setup methods.

With INSTEON this problem doesn't arise because there is simply no need for network enrollment. There is no routing, and there is no network controller. At the factory, INSTEON devices are each given a unique ID number that serves as a permanent network address. All INSTEON devices *automatically* become part of an INSTEON network and start simulcasting repeated messages as soon as a customer powers them up.

INSTEON versus Z-Wave Comparison Table

The following table summarizes the differences between INSTEON and Z-Wave.

Property	INSTEON	Z-Wave
Media	Powerline and Radio	Radio Only
Module Types	All are peers	Controller Static Controller (SUC) Slave Routing Slave Enhanced Slave Installer Bridge
Message Propagation	Simulcasting Repeaters ¹	Routing Tables
RF Data Rate, bps	38400 Instantaneous	9600 Instantaneous

Property	INSTEON	Z-Wave
RF PHY	FSK in ISM Band	FSK in ISM Band
Powerline Data Rate, bps	13,165 Instantaneous PL 2,880 Sustained PL 1,440 Standard Msg Payload 1,698 Extended Msg Payload 1,034 User Data	None
Powerline PHY	131.65 KHz Carrier BPSK	None
X10 Compatible ²	Yes	N/A
Powerline Phase Bridging	RF or Hardware	N/A
Acknowledged	Yes	Yes
Addressing	24-bit pre-assigned module ID (16,777,216 Devices)	32-bit Home ID 8-bit Node ID (232 Devices per Network)
Network Enrollment	Automatic	Part of Installation Procedure
Groups	Databases in modules	Controller Database
Commands	2-byte (65,536)	Command Classes
Device Types	3-byte (16,777,216)	Device Classes
Global Clock	Powerline Zero Crossing	No
Retries	Yes	Yes
Message Length	15, 33 bytes	Variable
Collision Avoidance	Yes	Yes
Triac Control	Yes	Yes
SRAM	256 Bytes	2048 Bytes
Flash	4K x 14 bits (7K Bytes)	32K Bytes
EEPROM External	4K Bytes	Up to 24K Bytes
EEPROM Internal	256 Bytes	0
Watchdog	Yes	Yes
Application Development	SALad Language in EEPROM SALad sIDE SmartLabs Device Manager, (DirectX and HTTP)	API in 8051, C compiled Compliance Test Tool

Other Radio Networks

This section compares INSTEON with [WiFi](#), [Bluetooth](#), [WiMax](#), [UWB](#), and [UHF](#).

WiFi

In recent years, WiFi (IEEE 802.11) has become the de facto standard for wireless LANs (local area networks) in the home, in offices, and at an increasing number of commercial 'hotspots' around the world.

About WiFi

When you buy a new laptop computer, it will probably have WiFi 802.11b/g built in. The reason is that your local Starbucks is a WiFi hotspot—when you go there with your WiFi laptop you can surf the web. If you have a broadband cable or DSL (digital subscriber line) connection to the Internet at home, you may have already bought a WiFi router for it so that you can check your email wirelessly from any room in the house, or even from the back yard. Wireless WiFi LANs let you share files, print, and use the Internet from anywhere within radio range without stringing wires or plugging in—no wonder it's already in 17% of U.S. homes.

IEEE 802.11, the standard underlying WiFi, actually comes in several different version, a, b, g, and n. 802.11a is for licensed operation in the 5 GHz band, and is mainly used by businesses. 802.11b and g are the WiFi versions that are shipping now. Version b, capable of communicating at 11 Mbps (megabits per second), appeared on the market in 1999, followed by version g at 54 Mbps in 2002. Version n, at 100 Mbps or more, will be next, with first shipments expected in 2006.

WiFi Version	Peak Data Rate	Radio Band
802.11a	54 Mbps	5 GHz
802.11b	11 Mbps	2.4 GHz
802.11g	54 Mbps	2.4 GHz
802.11n	100 Mbps minimum	2.4 GHz

The typical range for a WiFi radio inside the home is 50 meters, or up to 100 meters unobstructed line-of-sight. WiFi is fairly power hungry—a typical receiver is always on, dissipating over half a watt—so battery-operated WiFi devices are rare. The number of nodes per WiFi LAN is limited to 32, and each node will need up to one megabyte of system resources plus a moderately powerful microprocessor to run the WiFi software. By using technology like PBCC (packet binary convolutional coding), CCK (complementary code keying), and DBPSK (differential binary phase shift keying), WiFi achieves impressive performance and reliability, but not without significant cost.

To form network connections and transport data, WiFi uses the same protocol as the Internet, TCP/IP.

About TCP/IP

For remotely connecting computers together, TCP/IP is used literally all over the world. TCP/IP ships data over all kinds of networks, from the smallest home office

WiFi LAN to the global Internet. It is so ubiquitous that people are starting to use it for VOIP (Voice Over IP) telephones and IPTV televisions.

TCP/IP is really two standards, TCP (for Telnet Control Protocol) and IP (for Internet Protocol). IP transports a block of information called a *datagram* from point A to point B by handling two basic functions, addressing and fragmentation. A header in the datagram carries an address field that network routers use to select a path for transmission to the ultimate destination. Other fields in the header are used to fragment and reassemble datagrams when necessary to get them through 'small packet' bottlenecks. Packets can be labeled with a *Type of Service* for traffic prioritization, and there are other options for security and routing restrictions. Datagrams can theoretically contain up to 65,536 bytes, but in practice they are limited to 576 bytes, or 64 bytes of header and 512 bytes of data.

IP handles each datagram independently from any other, without any logical connections, virtual circuits, or guarantee of delivery. TCP makes up for these shortcomings by providing reliable, streaming connections. Streaming hides the underlying datagrams and makes data look like a sequence of bytes, much like a file. Reliability comes from retransmission if data becomes lost. TCP also handles network adaptation and flow control to maximize throughput without overloading the network.

Anyone who uses the Internet knows that this technology works superbly well. Why not use it, then, to connect *everything*? Shouldn't light switches have IP addresses? The answer is that yes, they should, but the light switch itself should not be networked using TCP/IP, or else the light switch would have to be built like a PDA. TCP/IP is *complicated*. A minimal implementation that can run on a medium-performance microprocessor such as an ARM7 requires 2K of RAM and 14K of code memory—and this does not take into account any application-level software or the modem hardware and firmware to connect to the physical communication media. Light switches of such cost and complexity cannot compete in the mass marketplace.

But we can still have the best of both worlds. The solution is to add a gateway device to your INSTEON network. The gateway device could be a PC connected to an INSTEON bridge device like SmartLabs' PowerLinc Controller, or it could be a lower-cost dedicated controller. The gateway only needs sufficient resources to run home management software that can control devices on your INSTEON network via an Internet or LAN TCP/IP interface.

An example of such home control software is the *SmartLabs Device Manager* (SDM), which interfaces to your INSTEON network using a collection of text commands or DirectX methods, properties, and events. Simultaneously, SDM uses TCP/IP to connect to the Internet or a LAN as an HTTP (hypertext transport protocol) server. With SDM running on a gateway device, your INSTEON devices can show up via HTTP on a web browser, or else in a custom software user interface that uses DirectX. INSTEON devices in your home can be plentiful and low-cost, but they can still appear as if they are on the Internet or part of a LAN, thanks to home management software running on a single gateway device.

INSTEON Compared with WiFi

Using WiFi to network together low-cost devices like light switches is theoretically possible, but WiFi is overkill by a wide margin. The *slowest* WiFi specification, 802.15b, delivers data at 11 megabits per second, far in excess of what a light switch needs to dim a light, or a thermostat needs to control a heater. TCP/IP

transport, arguably the gold standard for packet routing, requires a *minimum* of 30 bytes of overhead per packet, with 2K of RAM buffers and at least 14K of code space just for the software stack.

WiFi radios work very well, but they are complicated, and so power-hungry that the only way to implement battery-operated devices is with rechargeable batteries and frequent recharging. Advanced as it is, WiFi nevertheless has no powerline backup. As with other single-media networks, if communication fails for any reason, the only recourse with WiFi is to retry.

In contrast, INSTEON's data rate is optimized for home control. It is fast enough that users don't experience a noticeable delay when controlling devices, but not so fast that simulcast messages might jam each other. And simulcasting is how INSTEON avoids routing altogether. All INSTEON devices within range repeat¹ identical messages at precisely the same time, with each device adding to the strength of the signal, so simulcasting is not only radically simpler than routing, it is more robust. Simplicity translates into low cost—the INSTEON engine for communicating both by powerline and radio occupies a mere 2600 bytes, with an entire application like a lamp dimmer residing in only 7K bytes, all running on a single low-end microcontroller.

Perhaps the strongest argument against increasing the cost of devices by building WiFi into them is that you can get the same benefit but without incurring the cost in every device. With a single gateway between a WiFi network and INSTEON, all of the devices on an INSTEON network can still appear as part of the WiFi network. WiFi is great at networking together computers, broadband access points, printers, and other high-performance gear. With an INSTEON gateway on the WiFi network, everything else in your home can join the LAN and appear on the Internet without costing a second mortgage.

Bluetooth

Bluetooth radio, defined by the Bluetooth SIG (special interest group, www.bluetooth.org) and standardized as IEEE 802.15.1, is a wireless ad-hoc point-to-point personal area networking (PAN) technology.

About Bluetooth

Designed for low power, but limited to a 10-meter range, Bluetooth radio is mostly used in wireless headsets for cellphones, automotive hands-free applications, and for PDA and PC cable replacement.

Bluetooth radios operate in the 2.4 GHz band, using frequency hopping spread spectrum (FHSS) to achieve a peak data rate of up to 3 Mbps (megabits per second). A Bluetooth PAN (personal area network) supports only seven nodes, yet the Bluetooth software stack can occupy up to 250 Kilobytes of system resources.

INSTEON Compared with Bluetooth

Bluetooth, because it was designed for personal connectivity, is much faster and more complex than needed for home control applications. Considering Bluetooth's cost, limited range, and lack of powerline backup, INSTEON is a superior choice for networking together mass-market devices like light switches.

Bluetooth coexists with INSTEON without any issues, because Bluetooth occupies the 2.4 GHz band and INSTEON radio uses 904 MHz.

WiMax

WiMax, which stands for *Worldwide Interoperability for Microwave Access*, is a wireless point-to-multipoint metropolitan area network (MAN) built on the IEEE 802.16 Air Interface Specification.

About WiMax

WiMax is a licensed broadband wireless access technology intended as an alternative to cable or telephone DSL (digital subscriber line) services. It uses fixed, fiber-connected base stations, like cellular, to beam up to 268 Mbps (megabits per second) of data to fixed antennas and receiver boxes on the customers' premises.

WiMax is harmonized with the ETSI (European Telecommunication Standards Institute) *HiperMAN Specification*, in an effort to support a global standard.

Utilizing the 3.3 to 10 GHz band for point-to-multipoint communications, WiMax has a range of 2 to 6 miles. For point-to-point connections with a direct line-of-sight, there is a 10 to 66 GHz version with a range of up to 30 miles.

The IEEE 802.16 *Air Interface Specification* is flexible but complex, with ATM (asynchronous transfer mode) or IP (Internet protocol) data transport both supported at a variable quality of service (QoS). Modulation can be OFDM (orthogonal frequency division multiplexing) with 256 or 2048 carriers, similar to DSL, or discrete multi-tone (DMT). The channel can be shared using either TDD (time-division duplexing) or FDD (frequency-division duplexing).

INSTEON Compared with WiMax

WiMax and INSTEON were designed for completely different purposes, so they do not compete in the marketplace. However, WiMax will eventually provide broadband connectivity to the Internet for millions of homes that are not currently served by cable or DSL. With an Internet gateway, INSTEON can offer very powerful remote interfacing to the home. For example, leaking water in your basement could be detected by an INSTEON sensor, which would send an INSTEON message to an IP bridge device like SmartLabs' PowerLinc Controller (PLC). The PLC could then send an email to your cellphone over the Internet using WiMax, instantly alerting you to the problem.

UWB

UWB, which stands for *Ultra-Wideband*, is a short-range high-speed wireless technology especially well suited for streaming media in the home. Potential UWB applications include:

- Streaming MPEG-2 or MPEG-2-HD video from a home set-top box, media gateway, or DVD player to multiple TVs
- Wirelessly interconnecting digital still cameras or camcorders
- Streaming multiple MP3 audio tracks throughout the home
- Implementing a wireless broadband Ethernet LAN using TCP/IP
- Replacing USB2 or IEEE 1394 FireWire cables

About UWB

UWB uses a stream of very narrow, hence very broadband, radio pulses to deliver data at over 480 Mbps (megabits per second) within a radius of under ten meters. By maintaining the radiated RF energy in the 3.1 to 10.6 GHz UWB spectrum under class B limits, UWB is approved by the FCC for unlicensed operation in the U.S.

Despite demonstrations of working UWB products at the January 2005 Consumer Electronics Show, the UWB specification has not been finalized. The reason is that different UWB stakeholders are advocating different versions of the protocol.

One camp, led by Freescale Semiconductor, advocates DSSS (direct sequence spread spectrum) data modulation at 29, 57, 86, and 114 Mbps, based on an IEEE 802.15.3a MAC (media access control) specification. The other faction, led by Intel, wants to use OFDM (orthogonal frequency division multiplexing) at 53.3, 106.7, and 200 Mbps, based on a European ECMA-368 PHY/MAC standard released in December 2005. The WiMedia Alliance (www.wimedia.org), an industry trade group that backs the ECMA-368 standard, expects these issues to be resolved in early 2006.

INSTEON Compared with UWB

UWB and INSTEON accomplish very different objectives, so they will not compete in the marketplace. A UWB light switch would make no sense, and INSTEON cannot be used to transmit multiple streams of high definition television. INSTEON networking, however, could indeed be used for controlling consumer electronics equipment that sends and receives UWB media streams.

UHF

UHF, or ultra-high frequency, is shorthand for numerous proprietary radio signaling methods that use the 260 to 470 MHz band.

About UHF

There is a lot of equipment in the world that currently uses radio for signaling. Most of these legacy applications communicate using the UHF band, with 433 MHz being a popular frequency. Applications include:

- Security systems
- Lighting controllers (Lutron, for example)
- Remote keyless entry for cars
- Garage door openers
- X10 RF handsets

To qualify for unlicensed operation, the FCC requires these devices to operate only intermittently and at low power, which results in a range typically limited to a few tens of meters.

Developed independently over many years by different manufacturers, these systems use differing modulation schemes and data encoding methods, with no truly open standards or interoperability.

INSTEON Compared with UHF

In contrast to the myriad of proprietary UHF signaling protocols, INSTEON is an open networking standard, requiring only that products be INSTEON-certified to ensure interoperability. There is no interference between INSTEON and UHF devices because they operate at different frequencies. However, by migrating to the INSTEON standard, products that currently use UHF would be able to interact with all kinds of other INSTEON-networked devices—and the more nodes on a network, the more benefits it can bring to users.

Conclusion

“Everything should be made as simple as possible, but not simpler.”

Albert Einstein (1879-1955)

INSTEON is brilliantly simple:

- INSTEON uses both existing house wiring *and* the airwaves to carry messages, so each signaling method can be kept very basic because they back each other up.
- INSTEON repeats messages by simulcasting, with each new device adding more energy to the INSTEON signal, so an INSTEON network becomes more robust and reliable as more devices are added.
- INSTEON devices are all peers, so they all handle messages in exactly the same way, with no need for special kinds of devices like network coordinators, master controllers, or routers.
- INSTEON device addresses are assigned at the factory, so users never have to deal with network enrollment.
- INSTEON device linking is easy—just press a button on each device and they're linked.
- INSTEON is optimized for home integration and control, with firmware that runs on the smallest microcontrollers, so INSTEON devices can be built in high volume at the lowest-possible cost.

Although INSTEON is simple, that simplicity is never a limiting factor, because INSTEON bridge devices can connect to all kinds of outside resources like computers, the Internet, and other networks in the home whenever needed. SALad-enabled INSTEON devices can be upgraded at any time by downloading new SALad programs. Networks of INSTEON devices can evolve as the marketplace does.

As this white paper has shown, some candidate home control networks are *too* simple—X10 is hidebound by its own legacy, and UPB can be jammed indefinitely by cheap lamp dimmers. Most other networking schemes are far more complicated than INSTEON, but without significant compensating benefits in the electronic home improvement marketplace. This is understandable, because none of them are dual mesh, and most of them are routed, so they had to trade cost and complexity for reliability and performance. The highest-performance networks, like WiFi, Bluetooth, WiMax, UWB, and HomePlug, do not compete with INSTEON because they are intended for different purposes, such as computer networking, media streaming, or cable replacement.

SmartLabs' mission is to make life more convenient, safe and fun. INSTEON provides the infrastructure that can make that dream come true. Anyone can now create products that interact with each other, and with us, in remarkable new ways. Now is the time. *Carpe diem!*

NOTES

1. Battery operated INSTEON RF devices, such as security sensors and handheld remote controls, must conserve power. Accordingly, they may optionally be configured so that they do not retransmit INSTEON messages from other INSTEON devices, but act as message originators only. Such devices can nevertheless both transmit and receive INSTEON messages, in order to allow simple setup procedures and to ensure network reliability.
2. At a minimum, X10 compatibility means that INSTEON and X10 signals can coexist with each other on the powerline without mutual interference. INSTEON-only powerline devices do not retransmit or amplify X10 signals. But X10 compatibility also means that designers are free to create hybrid INSTEON/X10 devices that operate equally well in both environments. By purchasing such hybrid devices, current users of legacy X10 products can easily upgrade to INSTEON without making their X10 investment obsolete.

APPENDIX I — UPB Powerline Noise Immunity Evaluation

To evaluate UPB's vulnerability to jamming by triac dimmers, we installed an HAI UPB model 35A00-1 Wall Switch Dimmer and a Web Mountain Technologies model SW-1 UPB Wall Switch Dimmer in junction boxes on a test wall. The loads were 100-watt incandescent lamps. We looked at the waveforms on the powerline near the UPB wall switches with an oscilloscope.

We were unable to set up the Web Mountain switch because the instruction manual that came with the switch says "Configuration requires UPB setup/configuration software (e.g. UPStart or a Web Mountain NetPlace server) and a model SPIM-01 Computer Interface." We did not have this equipment, and the manual did not offer any alternatives, so we were unable to find a way to get the switch to send UPB signals, even after trying various combinations of switch button presses.

The HAI switch, however, sent UPB signals whenever we pressed the rocker. We set up the oscilloscope to clearly display the UPB signal pulses appearing in the four possible positions before the powerline zero crossing. The earliest pulse occurred 1080 microseconds before the zero crossing.

We could also tell when the HAI switch was sending UPB signals by listening to it. The switch made a clearly audible buzzing noise whenever we saw UPB pulses on the scope.

Holding down the HAI rocker brightened and dimmed the load, and we could see the triac switching transient on the scope. At the dimmest setting, the transient occurred 1440 microseconds before the zero crossing. The lamp that the switch was controlling was not fully dim—it was clear that the lamp was still on even with the room brightly lit.

We then installed a Lutron model S-600H-WH 600W manual dimmer switch in a junction box with ten feet of wiring between it and the HAI UPB switch. The load on the Lutron dimmer was another 100-watt incandescent lamp. When we moved the slider on the dimmer we could clearly see the triac switching transient moving in response on the oscilloscope display.

At the dimmest setting the Lutron triac transient occurred 880 microseconds before the zero crossing, and the lamp appeared much dimmer—its filament was barely glowing.

To test whether the Lutron dimmer's transient would jam the HAI UPB signal, we dimmed the Lutron switch all the way and then pressed the paddle on the HAI switch. With a triac transient occurring within the UPB signal window, the HAI switch did not send any UPB pulses. However, to our surprise, when we brightened the Lutron switch so the triac transient occurred before the UPB window, the HAI switch did begin sending UPB pulses even though we weren't pressing the paddle. We speculated that the HAI switch interpreted the triac transient as a UPB signal collision, so it queued up its previous message or messages until the collision went away.

Next we replaced the Lutron dimmer with a Leviton 6631-LW 600W manual dimmer and repeated the test. The Leviton dimmer's dimmest setting caused a triac transient 860 microseconds before the zero crossing, so it was not surprising to observe the same behavior from the HAI UPB switch. We could reliably prevent UPB

signaling by moving the Leviton slider near the dimmest setting, and when we moved the slider to a brighter setting the HAI switch sent the message that was queued up.

The susceptibility of UPB signaling to jamming from common triac dimmers means that homeowners who install UPB devices must be very careful not to have any non-UPB-compatible dimmers in their homes. In an apartment building, however, neighbors may very well have such dimmers. If your neighbors dim their lights too far, your UPB devices may not work, and there is not much you would be able to do about it. Some time later, when your neighbor brightens his jamming dimmer, your UPB devices might then send long-queued-up commands with rather startling results.

APPENDIX II — Z-Wave Usability Evaluation

To gauge the user experience with Z-Wave devices, SmartLabs evaluated a small Z-Wave system consisting of the following products:

Num	Mfr	Product	Model Number	Form Factor	Powered By	Features
1	HomePro	Remote Control	ZTH100	Handheld	2 AA cells	12 x 2 text LCD
2	HomePro	Lamp Module	ZDP100	Plug-in	110 VAC	Dimmer
1	Intermatic	Remote Control	HA07	Desktop	4 AA cells	Custom Icon LCD
2	Intermatic	Lamp Module	HA03	Plug-in	110 VAC	Dimmer
1	Intermatic	Indoor Appliance Module	HA02	Plug-in	110 VAC	Relay Switch
1	Intermatic	Outdoor Appliance Module	HA04	Plug-in	110 VAC	Relay Switch

We plugged the products into outlet strips on a tabletop, so we did not test the range and reliability of the Z-Wave signal. By combining products from two different manufacturers, HomePro and Intermatic, we checked the ability of Z-Wave certified devices to interoperate. The results fell into four categories:

1. [Z-Wave Network Setup](#)
2. [Z-Wave User Interfaces](#)
3. [Missing Z-Wave Modules](#)
4. [Z-Wave Lamp Dimming](#)

Z-Wave Network Setup

Z-Wave Allows Only One Master Controller

Because Z-Wave is a routed network, new Z-Wave devices must be enrolled into the network before they can be used. You can only have one *master controller* in your Z-Wave network, because the master controller maintains the database that represents the network's topology.

Whichever device you choose as your master controller, to add *secondary controllers* to your Z-Wave network, you must go through a multistep controller replication process. The upshot is that only your master controller can add or remove Z-Wave devices from your network, so any time you modify your network you have to repeat the controller replication procedure in order to copy the new master controller network topology database to all of your secondary controllers.

Z-Wave Network Enrollment and Button Associations

After you add a new Z-Wave device to your network by enrolling it with your master controller, there still has to be a choice of which button or buttons on the master controller will operate the device. In other words, there are two issues to resolve when adding a new Z-Wave device to your Z-Wave network—network enrollment and button association.

The two different controllers that we evaluated use two very different methods for accomplishing these two tasks. The Intermatic controller requires two distinct steps—first you enroll a new device into the Z-Wave network, then you associate buttons to control the device. We had to refer to the Intermatic instruction manual to find out how to perform these steps, because the controller's buttons do not have obvious labels for performing this setup, and the display cannot show arbitrary text prompts.

In contrast, the HomePro handheld controller merges the two setup steps into one by using the concept of HomePro *groups* and HomePro *scenes*. There are six numbered buttons on the HomePro handset, with each button able to control a HomePro group or scene. The difference between a group and a scene is that scenes can remember brightness settings.

To add a new Z-Wave device to the HomePro master controller, you traverse a text menu tree and follow text prompts on the display. The controller's software can recognize whether a device is new to the network or if it is already enrolled in the network and just being associated with more than one HomePro group button. The software automatically enrolls devices that are new to the network without involving the user, so users believe they are simply performing button associations.

Difficulty Enrolling Previously Enrolled Z-Wave Devices

At first try we were unable to enroll the Intermatic devices using the HomePro handset as the master controller. The reason turned out to be that the Intermatic devices had been previously enrolled using the Intermatic controller as the master. It was not clear from the instruction manual what to do, but we found a menu screen on the HomePro handset that allowed us to reset the Intermatic devices. After resetting we succeeded in getting the Intermatic devices to work with the HomePro handset.

Aware of the problem, we found that when using the Intermatic controller as the master, we could similarly reset devices before enrolling them by following a procedure in the instruction manual.

Issues when Replicating the Master Controller

While we had the Intermatic controller set up as the master, we tested the procedure for replicating it into the HomePro handset as a secondary controller. This required studying the instruction manuals for both controllers, and we had trouble with steps timing out before we could complete them. Eventually we succeeded in setting up the HomePro handset so that it contained the network enrollment information, but the button associations did not transfer over even though we selected that option. We therefore had to manually set up new button associations on the Home Pro handset.

Device Location Issues

Our test setup was on a tabletop, so range was not an issue. However, the instruction manual recommended that if we moved a device from one location to another, we should unenroll it from the Z-Wave network then re-enroll it at the new location. Presumably, this would force the master controller to adjust its network topology database so that routing would be optimized, but the re-installation of a device can be a chore, especially if you forgot how to do it and have to appeal to the instruction manuals.

Another issue that concerned us was enrolling devices far away from the master controller. Z-Wave optionally allows for a controller's radio transmissions to be at low power during network enrollment for security reasons. If a manufacturer enabled this feature, how would you get a Z-Wave device that was not close to the master controller to join the network? Even at normal transmission power levels, how would a master controller know how to route messages to an unenrolled device that is so far away that intermediate routing slaves would be needed to reach the new device? Unless Z-Wave has provided software to handle these conditions, it would seem that a device would have to be kept near to the master controller for initial enrollment, and this could be a problem with wired-in devices like light switches.

Z-Wave User Interfaces

Button Pairs versus On/Off Toggles

The Intermatic controller uses on/off button pairs to control devices, while the HomePro handset uses single buttons that alternate between on/bright and off/dim. Neither controller is 'guest friendly', meaning that a person who had never seen one before would not be able to figure out how to use it without some trial and error.

LCD Displays

Both devices have an LCD display, but the Intermatic display has fixed icons and text messages along with a 'union jack' text area of only 6 characters, as opposed to the 12 character by 2 line text display of the HomePro handset. Neither display has a backlight, so they cannot be read in the dark. The Intermatic display is always on, but the HomePro display turns off entirely after you have not pushed a button for several seconds. Pushing any button turns the HomePro display back on, but then you wonder, did the button you pushed only turn on the display or did it also perform the button's function?

Button Labeling

The Intermatic controller has a hinged cover that hides buttons that you use to perform setup functions. It is not obvious that the cover can be opened, but inside the cover there is space for writing labels for the buttons. When the cover is closed there are six on/off button pairs labeled with numbers, but there is also a *shift key* that causes the buttons to control an additional six groups of devices. There is a small icon on the display that indicates the shift state, but unless you check it before you press a button, you might be controlling the wrong set of six devices.

On the HomePro handset, it is not clear from the button labels what they do. The group toggle buttons are labeled with a single digit, with no provision for writing a descriptive label. To control a HomePro scene, you first have to press a button labeled *S*, but this is far from obvious. Digit buttons alternate between turning groups of devices on and off, but for scenes you have to program one button to turn the scene on and another button to turn the scene off.

The HomePro remote has buttons with icons for *all on* and *all off*, but it is not obvious what they will do unless you already know. There is a menu system, but the only way to navigate it is by trial and error, because there is no indication that there are submenus or additional choices in the current menu. It is easy to get into the menuing system inadvertently without knowing how to get out.

Keypress Responses

Keypresses on the HomePro remote are erratically debounced. For example, pressing the *OK* button once can answer *OK* to *two* menu prompts so quickly that the first menu prompt cannot be read.

The HomePro handset locked up once after multiple presses of the *1* button, displaying *Dimming Group 1*. We had to remove the batteries to reset the unit, and then we had to reset the time.

User Feedback

Both controllers give feedback when a Z-Wave message is sent successfully, but there is no indication as to what the state of the device you are controlling actually ended up being. For example, you can brighten or dim lights by holding down a button for a variable length of time, but unless you can see the light that you are controlling, you will not know how bright it got when you let up on the button.

You can turn lights on and off rapidly using the HomePro handset, but if you try this with the Intermatic controller, the commands may not get through, even though the display shows the *Successful* text icon.

Missing Z-Wave Modules

Communication Retries Lock Up the Controller

There is a severe problem when controlled devices are removed from the Z-Wave network without going through the network unenrollment procedure. In that case Z-Wave controllers will keep trying to send commands to the missing devices for several seconds. During the retry time, both the Intermatic and the HomePro controllers ignore button presses, so you are locked out from doing anything else.

There are several reasons that a Z-Wave slave device might be removed from the Z-Wave network without unenrolling it. One is that the device might simply have stopped working. Another might be that you have some Z-Wave devices that control your Christmas lights, but when you remove the lights and unplug the devices after Christmas, you don't remember that you are supposed to go through the unenrollment procedure. A third possibility is that you've installed a lamp module on a switched outlet and somebody has turned the switch off, so the lamp module is not powered up.

To test the effect of missing modules, we enrolled three modules as a group using the Intermatic controller as the master. With the first module unplugged, the Intermatic controller would control the first one, retry the second one without success for 7 seconds, then control the third one, and finally illuminate a *Not Successful* Message. The Intermatic controller would not respond to button presses during the entire 7 seconds. When we set up the HomePro handset as a secondary controller to the Intermatic master, the HomePro handset controlled the two modules that were present right away, but then the handset ignored button presses for 5 seconds until the LCD displayed a *Group Failed!* message. When we set up the HomePro handset as the master controller, the lockup period was shortened to just under 2 seconds.

No Way to Unenroll Lost or Broken Devices

If the HomePro handset is your master controller, there is no way to unenroll a previously enrolled device without pushing a button on the device. Clearly, if the

device stops working, you have a problem, because there is no button to push. The only remedy is to factory reset the entire HomePro controller, wiping out all of your network enrollment and other setup information, and then to manually re-enroll every device in your Z-Wave network.

Z-Wave Lamp Dimming

The lamp dimmers from Intermatic and HomePro do not dim lights the same way. Both the Intermatic and HomePro dimmers come preprogrammed with different ramp rates (the time it takes to dim or brighten a lamp all the way), and neither ramp rate is adjustable by the user. HomePro modules dim or brighten first, followed by Intermatic. Neither module dims to full off.

When you use the Intermatic controller to dim lights manually by holding down an *off* button, HomePro modules first go to full brightness before they start dimming, but Intermatic modules start dimming from whatever their current brightness is. With mixed devices like this, if you had set the lights to a dim level you liked and then tried to dim the lights further, you would get a surprise when the HomePro modules suddenly went full on before dimming.

If you use the HomePro handset to manually dim both the Intermatic and HomePro modules to a certain level, then turn them off with a quick button push, another quick button push turns the Intermatic modules full on, but the HomePro modules go to the last level you dimmed to. Trying to fix this by manual dimming doesn't work because the modules have different ramp rates.

The HomePro modules we tested dimmed somewhat erratically, especially when operated with the Intermatic controller. The light sometimes 'flashed' at the end of dimming when we let up on the controlling button.

While it is true that Z-Wave devices from Intermatic and HomePro do 'interoperate' in the sense that both controllers can activate both kinds of dimmers, the actual behavior of the lights is not the same. For consistent performance, consumers would have to purchase all Intermatic or all HomePro devices.