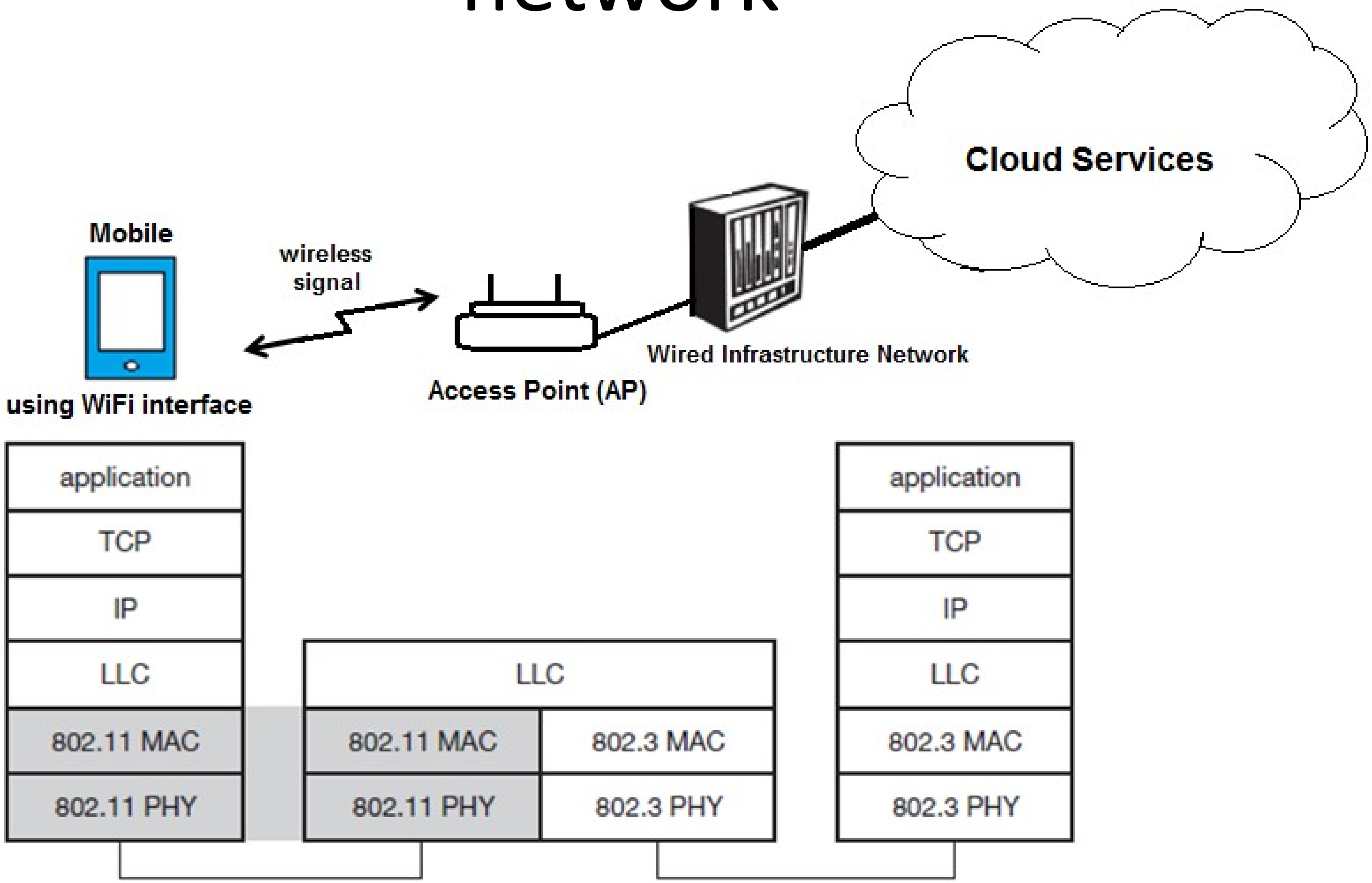
UbiComp IoTs
L6
(IoT - part A)

By: Dr. Abdussalam Nuri Baryun
ITMC421
Mobile Computing Dept.
University of Tripoli

Current Technologies and standards used within UbiComp Applications

- WiFi (IEEE802.11)
- Bluetooth (IEEE802.15.1)
- GPS
- Zigbee and IEEE802.15.4
- RFID
- Sensors and actuators (wired or wireless)
- Motors (DC and AC motors)
- MEMS (Micro Electro Mechanical Systems)
- (there are other technologies, read textbook, papers and sheets provided within lecture)

Bridging WiFi to infrastructure network



WiFi Communication Scenarios

For addressing, the following four scenarios are possible Cases:

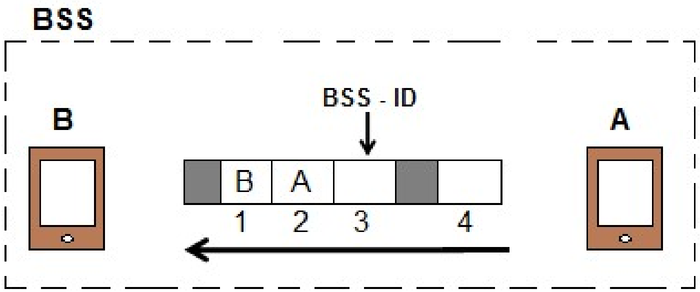
Case 1) Ad-hoc network: If both DS bits are zero, the MAC frame constitutes a packet which is exchanged between two wireless nodes without a distribution system. DA indicates the destination address, SA the source address of the frame, which are identical to the physical receiver and sender addresses respectively. The third address identifies the *basic service set (BSS ID)*, the fourth address is unused (see the following Figure).

Case 2) Infrastructure network, from AP: If only the 'from DS' bit is set, the frame physically originates from an access point. DA is the logical and physical receiver, the second address identifies the BSS, the third address specifies the logical sender, the source address of the MAC frame. This case is an example for a packet sent to the receiver via the access point.

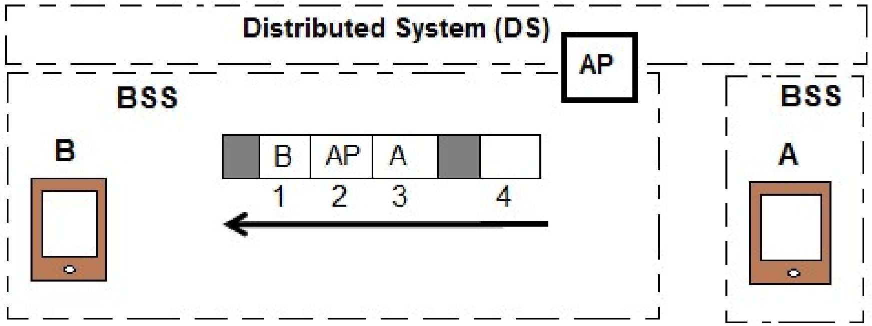
Case (3) Infrastructure network, to AP: If a station sends a packet to another station via the access point, only the 'to DS' bit is set. Now the first address represents the physical receiver of the frame, the access point, via the BSS identifier. The second address is the logical and physical sender of the frame, while the third address indicates the logical receiver.

Case (4) Infrastructure network, within DS: For packets transmitted between two access points over the distribution system, both bits are set. The first **receiver address (RA), represents the MAC address of the receiving access** point. Similarly, the second address **transmitter address (TA), identifies the** sending access point within the distribution system. Now two more addresses are needed to identify the original destination DA of the frame and the original source of the frame SA. Without these additional addresses, some encapsulation mechanism would be necessary to transmit MAC frames over the distribution system transparently.

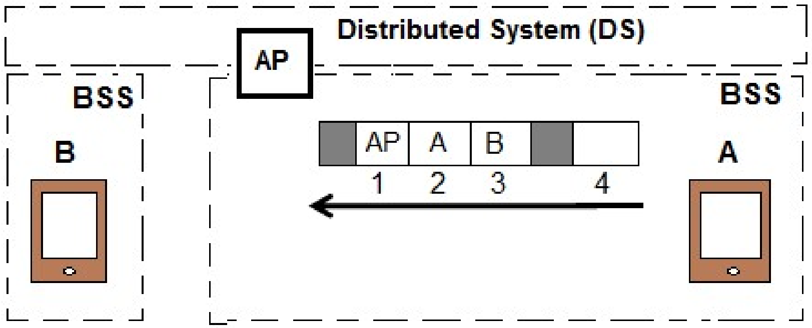
WiFi Communication Cases



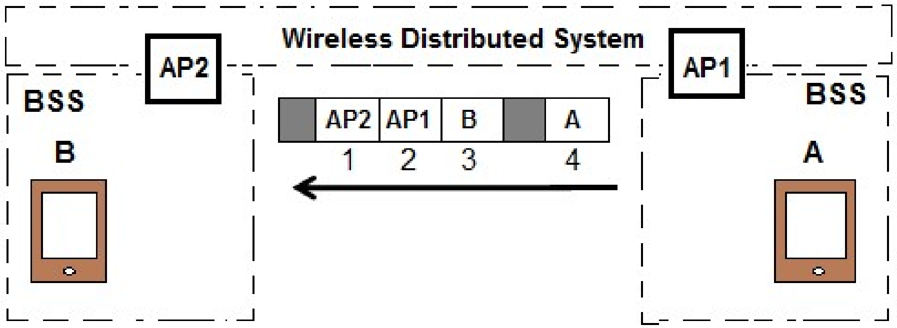
Case (1) two WiFi nodes communicating without DS
(to DS=0, from DS=0, address_1= DA, Address_2= SA)



Case(2) distributed system through AP sends to WiFi node B
(to DS=0, from DS=1, address_1= DA, address_2= TA, address_3= SA)

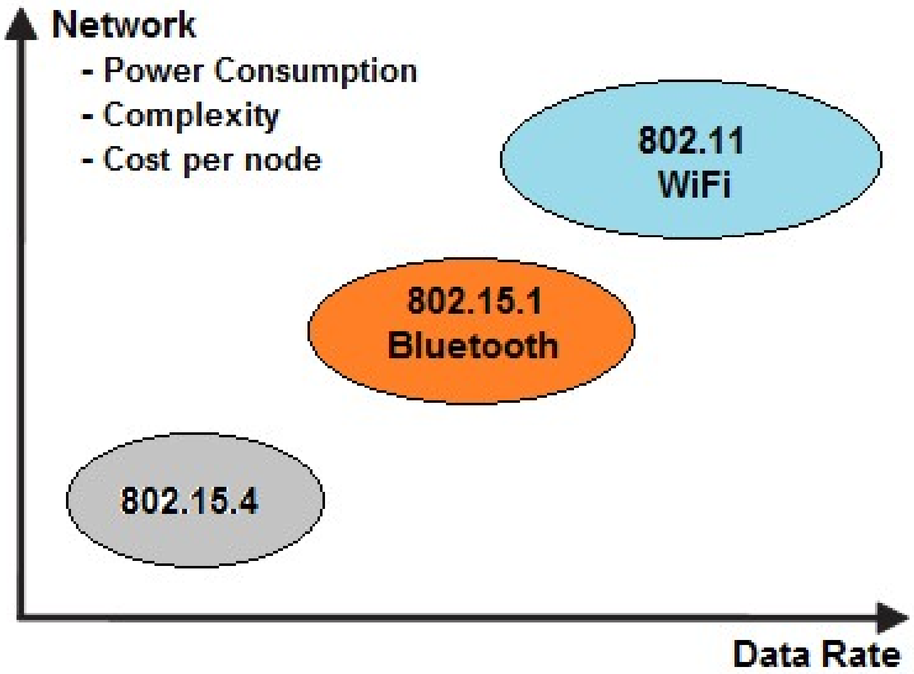


Case (3) WiFi node A sends to DS through AP.
(to DS=1, from DS=0, address_1=RA, address_2=SA, Address_3=DA)



Case (4) frame is going from AP1 to AP2 in the wireless DS.
(to DS=1, from DS=1, address_1=RA, address_2=TA, address_3=DA, address_4=SA

Comparing WiFi with other personal technologies



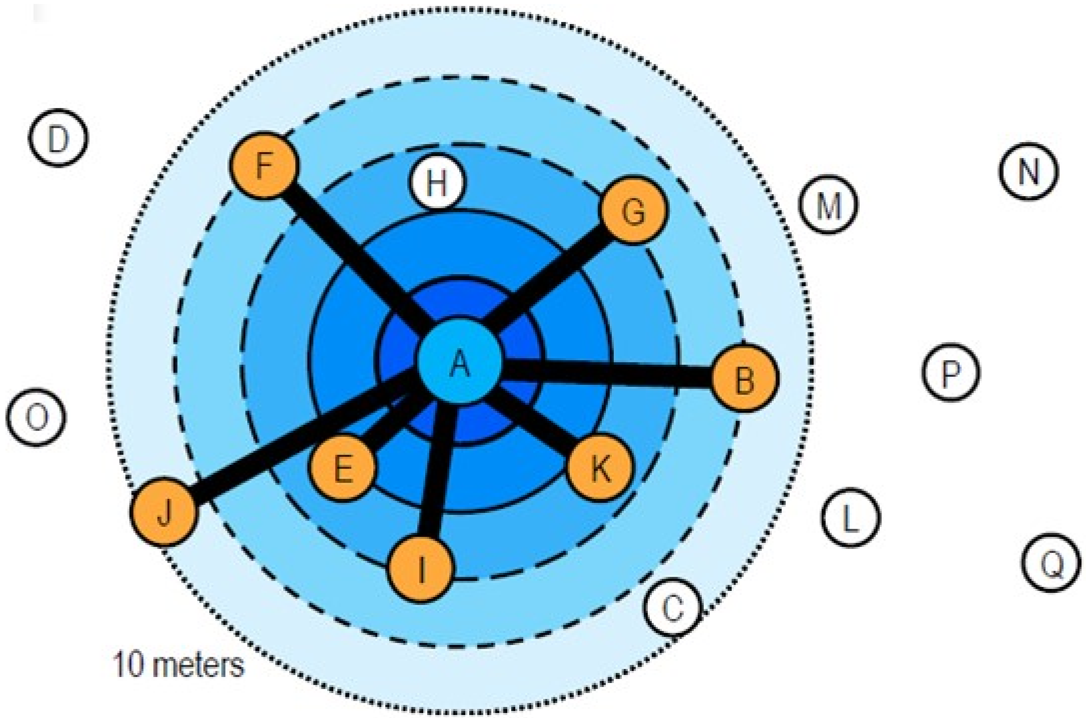
	Data Rate	Coverage (Range)
IEEE802.15.4	20 to 250 Kbps	10–100 m
Blutooth	1 Mbps	2–10 m
WiFi	1 to 11 Mbps	30–100 m

Comparing WiFi with other personal technologies

WPAN / Properties	WiFi	Bluetooth	IEEE802.15.4
Max Network Size	32	8	65,535
Range	100 m	10 m	70 – 100 m
MAC protocol	CSMA/CA	TDMA/TDD	CSMA/CA
Modulation (PHY)	DSSS/CCK	FHSS/BPSK	DSSS/QPSK DSSS/BPSK
Number of channels	14	79	27
Data Rate	11 Mbps	1 Mbps	250 Kbps
Latency	3 ms	200 – 300ms	30 ms
RTT	4 - 200ms		
Power Consumption	hours	days	months
Network Complexity	high	medium	simple

Bluetooth Network Architecture: Piconets & Scatternets

- One master attached with slaves up to 7 is a piconet



Bluetooth Scatternet

- Bluetooth devices participate in multiple piconet, making scatternet.

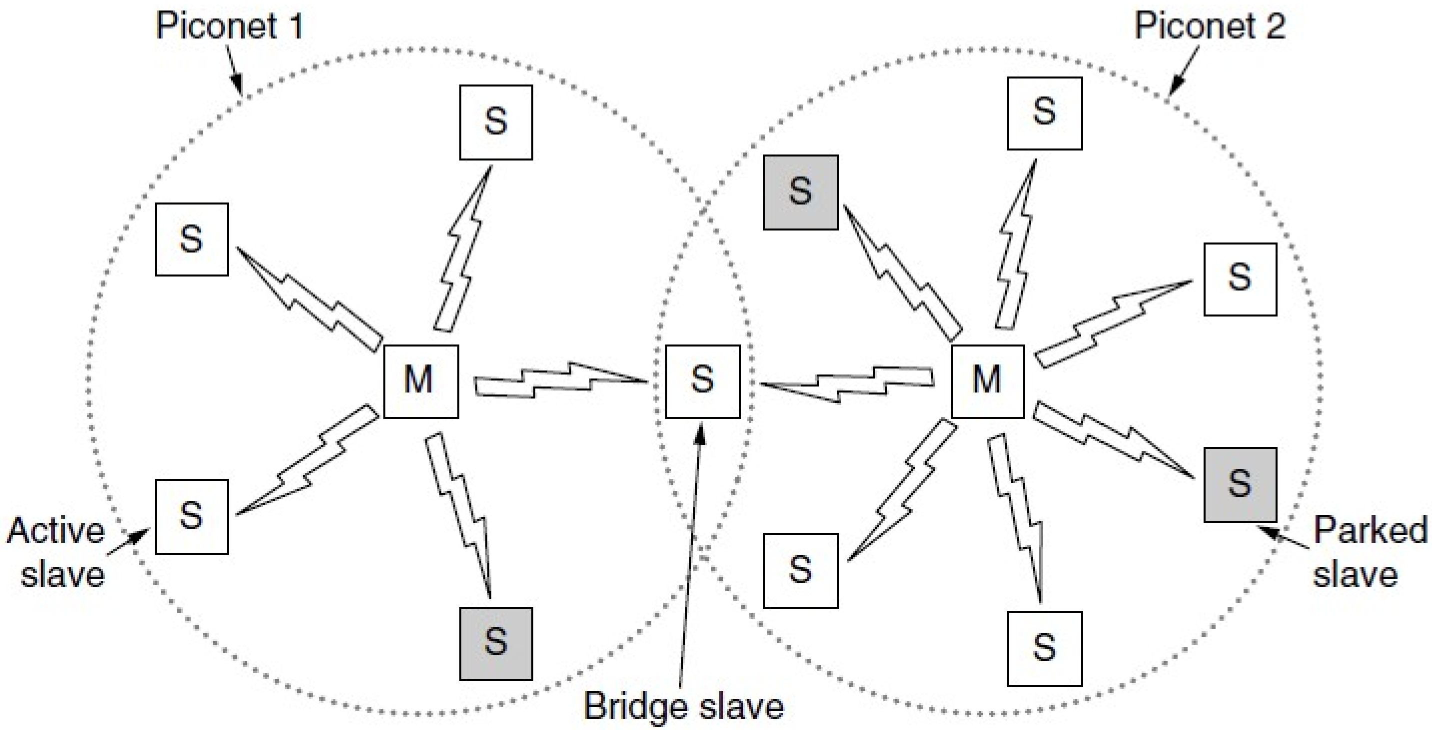
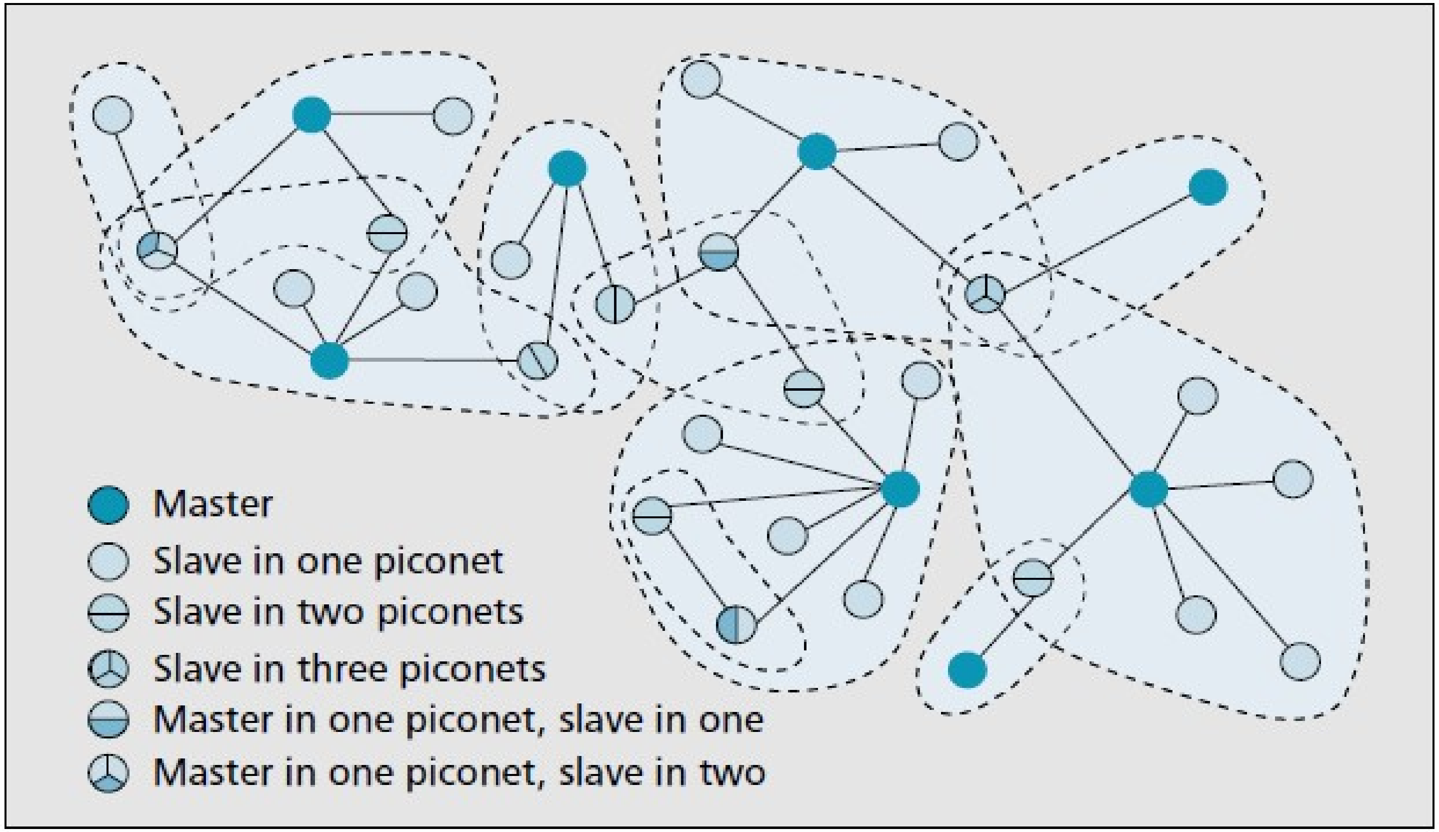


Figure 4-34. Two piconets can be connected to form a scatternet. **Reference [1]**

Bluetooth Scatternet

- A slave participates in max 3 piconet. A master in one piconet can participate as slave in other two piconets.



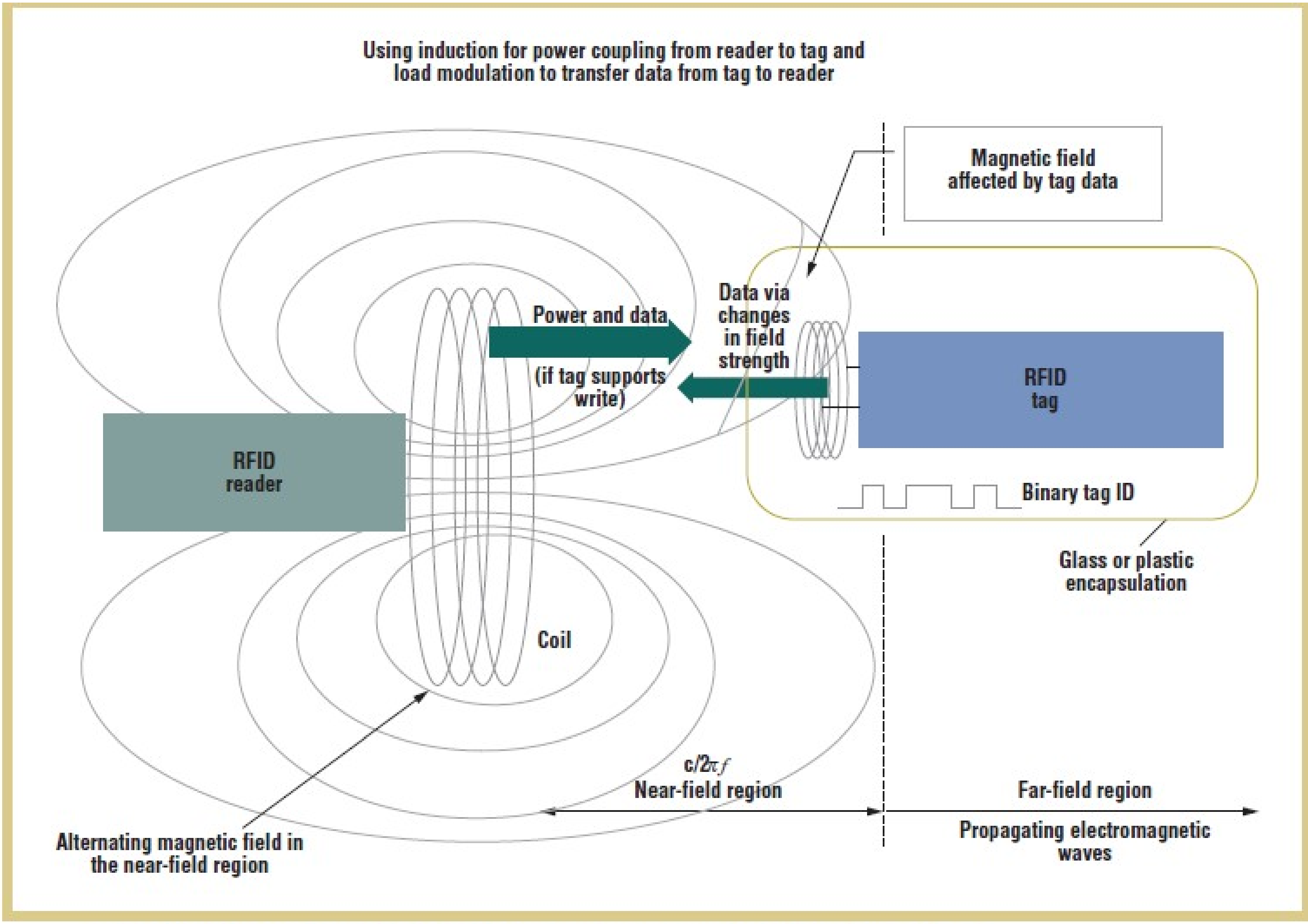
Reference [3]

Wearable Computing

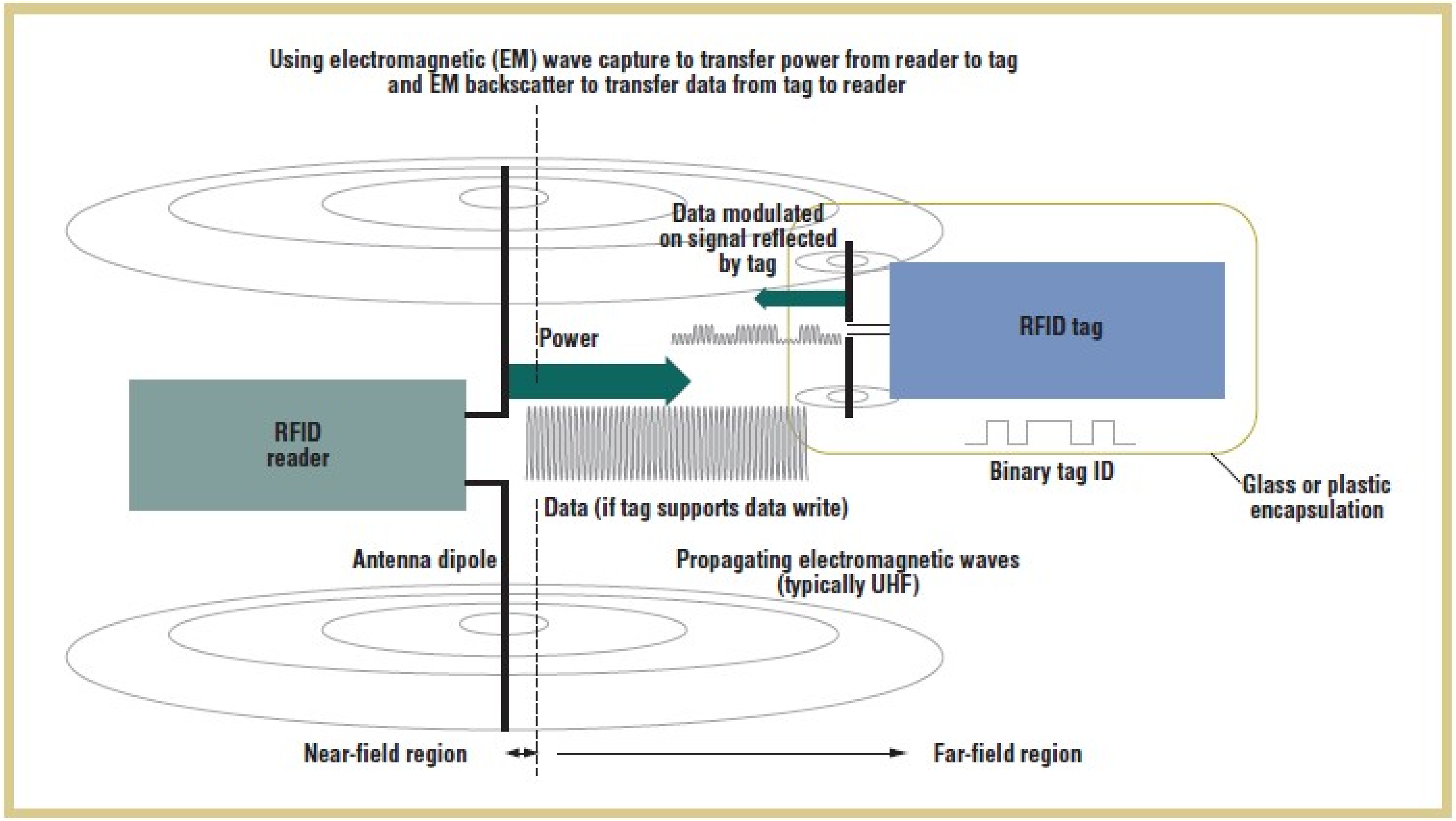
- RFID let computers quickly identify tagged objects without requiring lengthy user input or complex machine-vision algorithms.
- A platform that could use a wide variety of sensors to determine a wearer's location, including GPS for outdoors and RFID and bar code scanners for indoor locations.
- The wearer would explicitly scan a tag to receive information about a location.

Tagging the Physical World

- Physical tags refer to digital tags, which are networked electronic devices with an identity, e.g.,
- A RFID (Radio Frequency Identifier) tag. When these are attached to or linked to physical objects,
- they provide a way to audit physical spaces and processes.



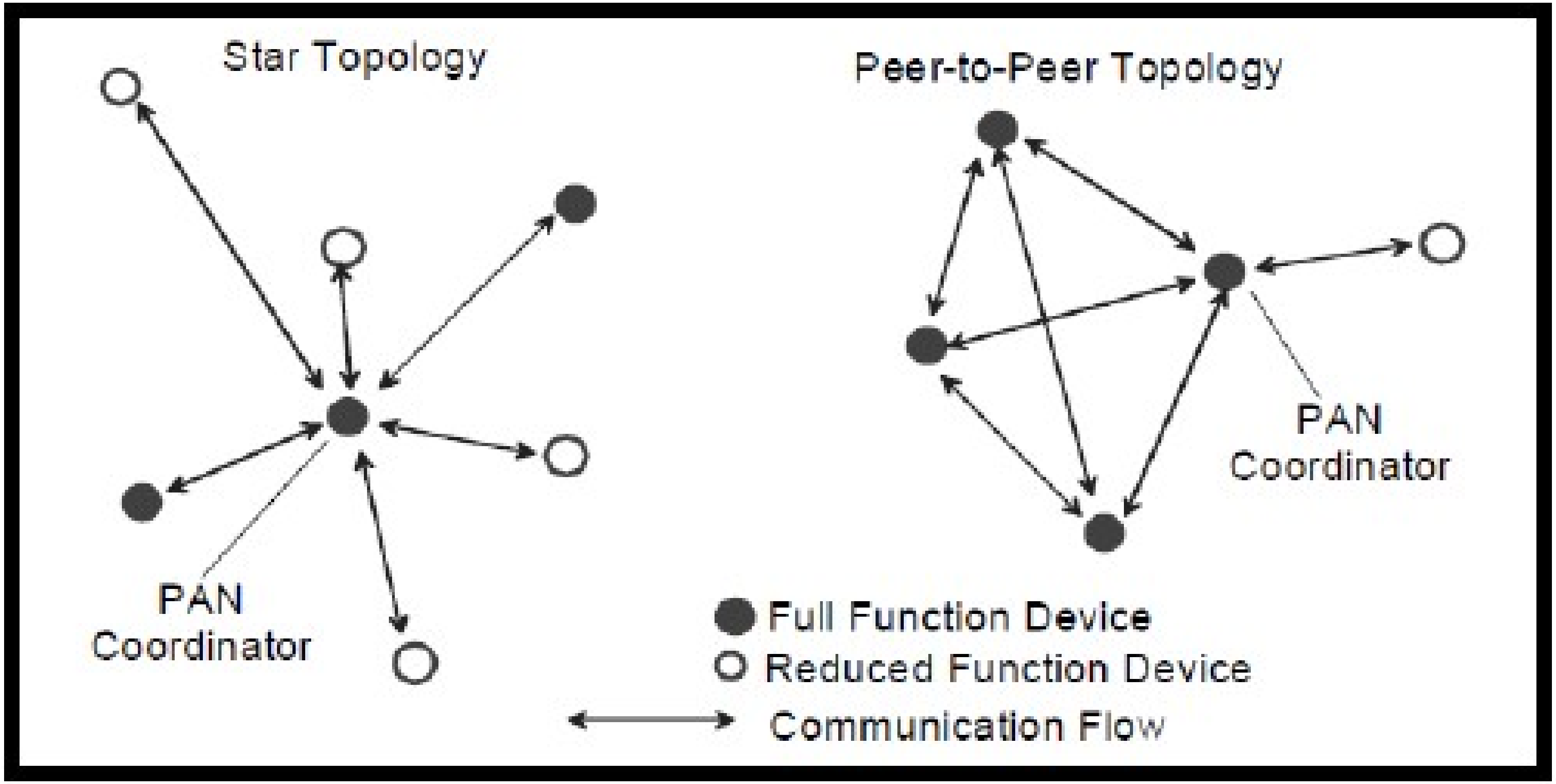
. Near-field power/communication mechanism for RFID tags operating at less than 100 MHz.



. Far-field power/communication mechanism for RFID tags operating at greater than 100 MHz.

- A passive tag consists of three parts: an antenna, a semi-conductor chip attached to the antenna, and some form of encapsulation.
- Active tags require a power source—they're either connected to a powered infrastructure or use energy stored in an integrated battery.

IEEE802.15.4 PAN Topology



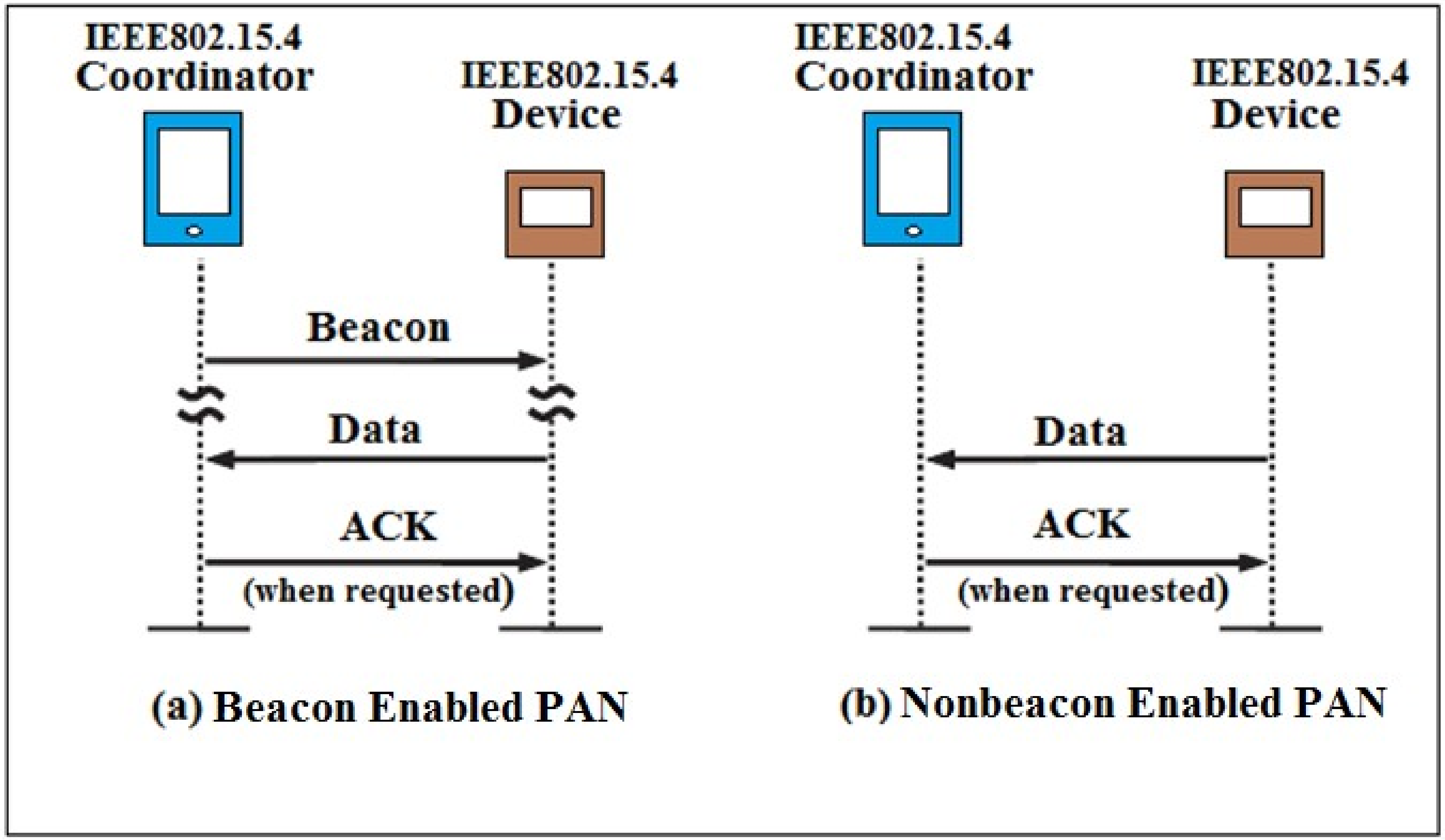
IEEE802.15.4 Devices

- 802.15.4 devices can be divided into two categories, which determine the topology and media access used by the network.
  - **Full function devices (FFDs)** can communicate directly with any other devices in the network.
  - **Reduced function devices (RFDs)** can only communicate with FFDs.
- The 802.15.4 standard allows networks to form rather a one hop star topology, or a multi hop peer to peer topology. The former is most appropriate in networks with few FFDs, whereas the latter is more resilient to node failure when many FFDs are available.
- The FFD may operate in three modes serving as a personal area network (PAN) coordinator, a coordinator, or a device.

ZigBee Basic Communication

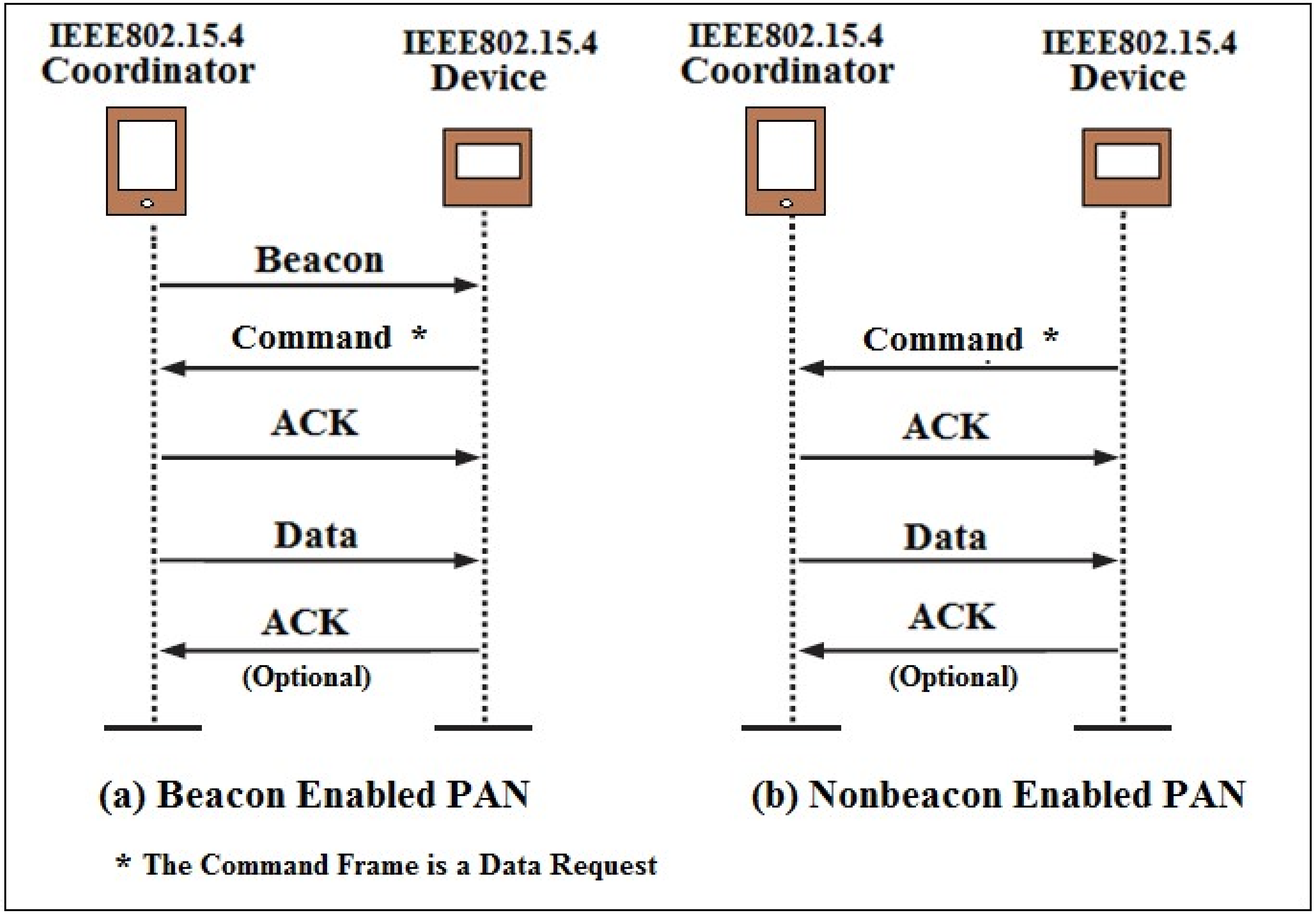
- There are three types of data transfer in IEEE 802.15.4:-
  - Data transfer to a coordinator from a device.
  - Data transfer from a coordinator to a device.
  - Data transfer between two peer devices.

Sending Data to a coordinator



IEEE802.15.4 Communication to the Coordinator

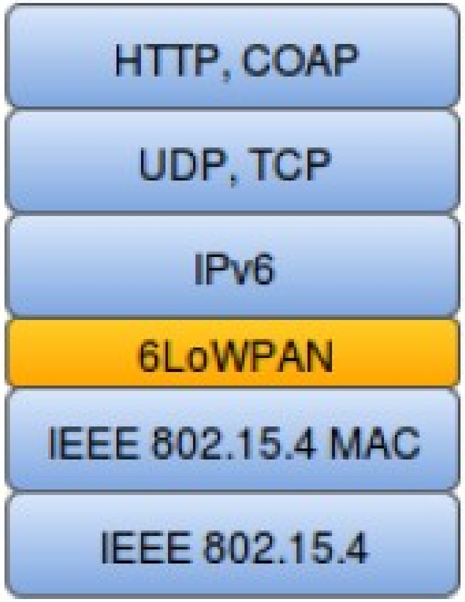
Sending Data to a Device



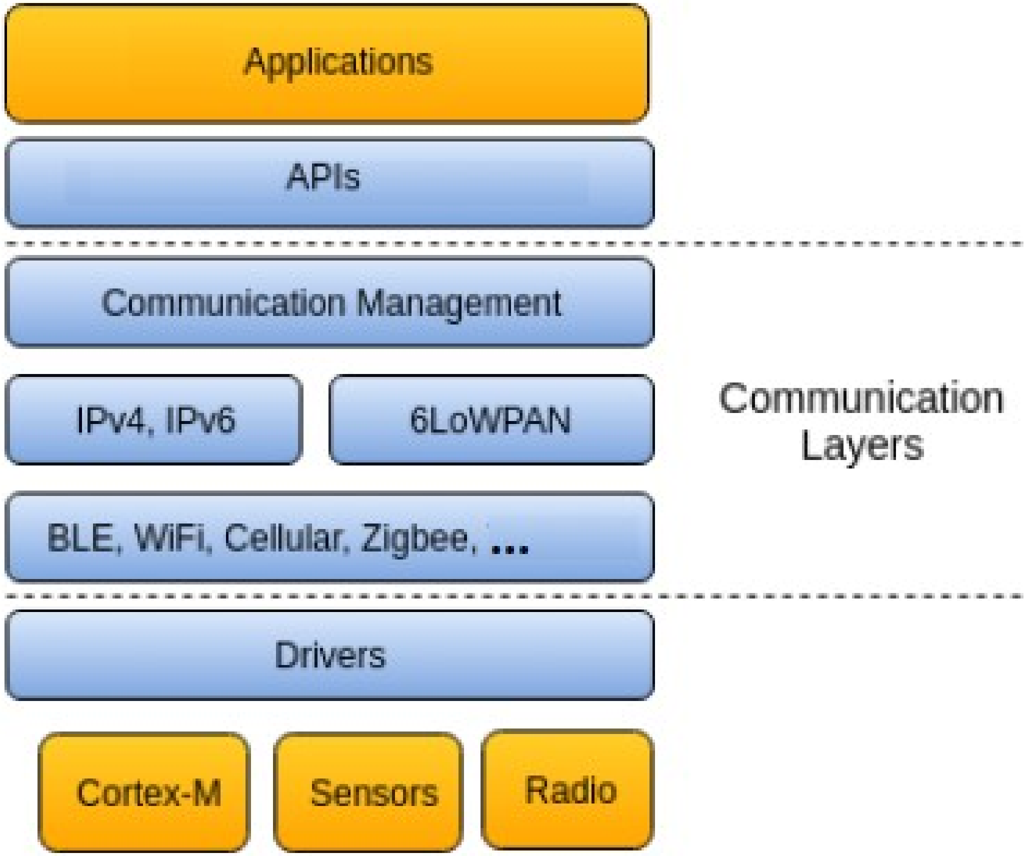
IEEE802.15.4 Communication To the Device

IoT Requirments

- Network communication Protocols. Support for IPv6, 6LoWPAN, and CoAP.
- Distributed System technologies. A need for a technology that can seamlessly integrate assorted devices into a monolithic communication network.
- New UbiComp application protocols that are compatible with current technologies and standards.



The 6LoWPAN stack



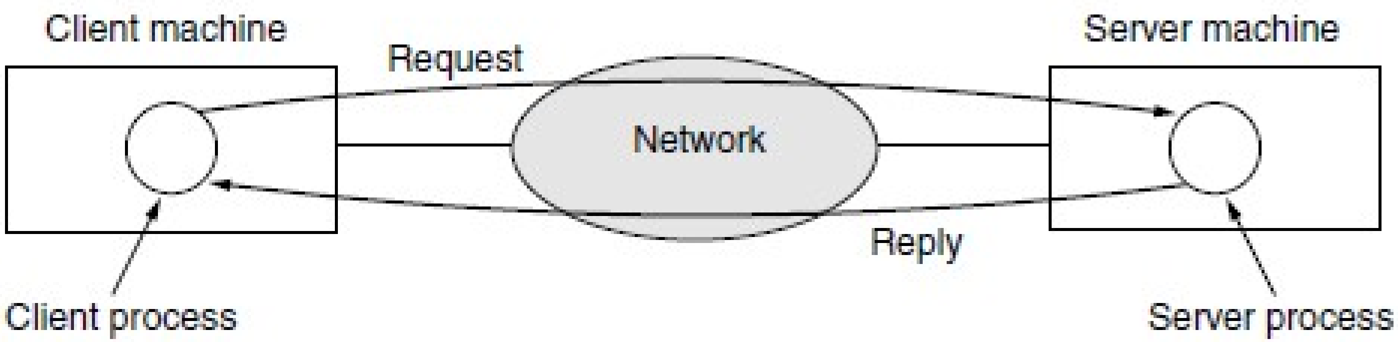
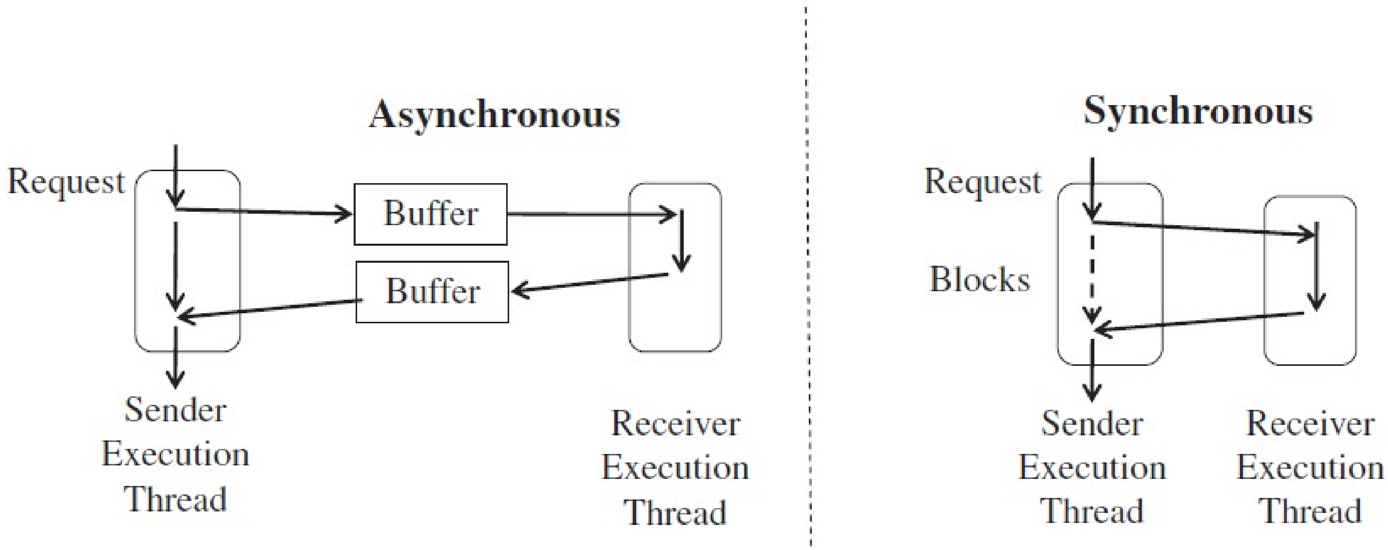


Figure 1-2. The client-server model involves requests and replies.



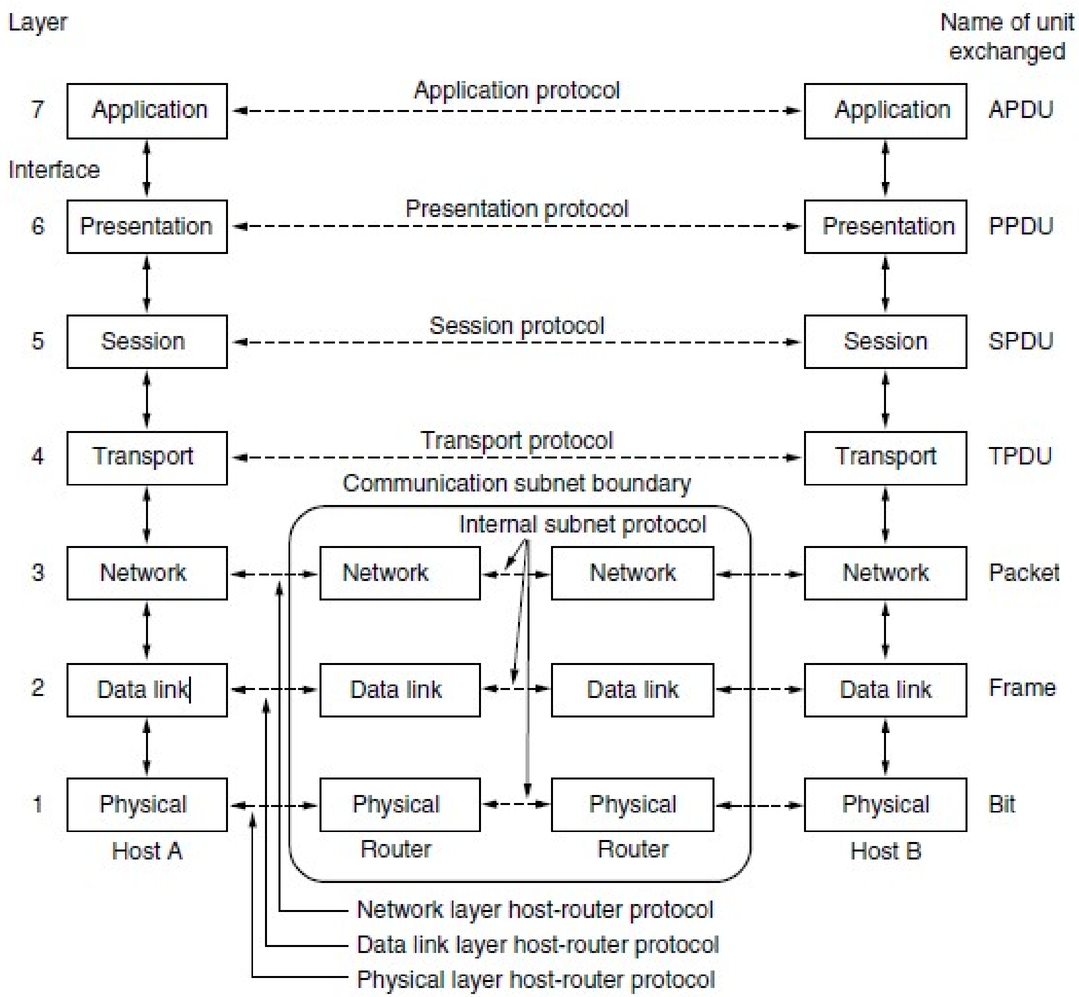


Figure 1-20. The OSI reference model.

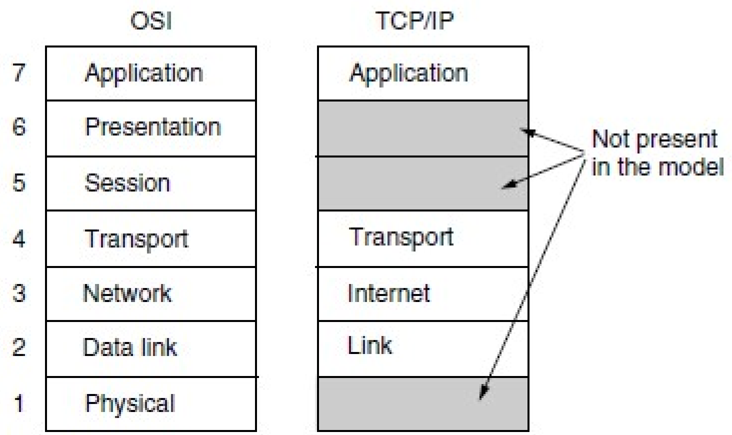
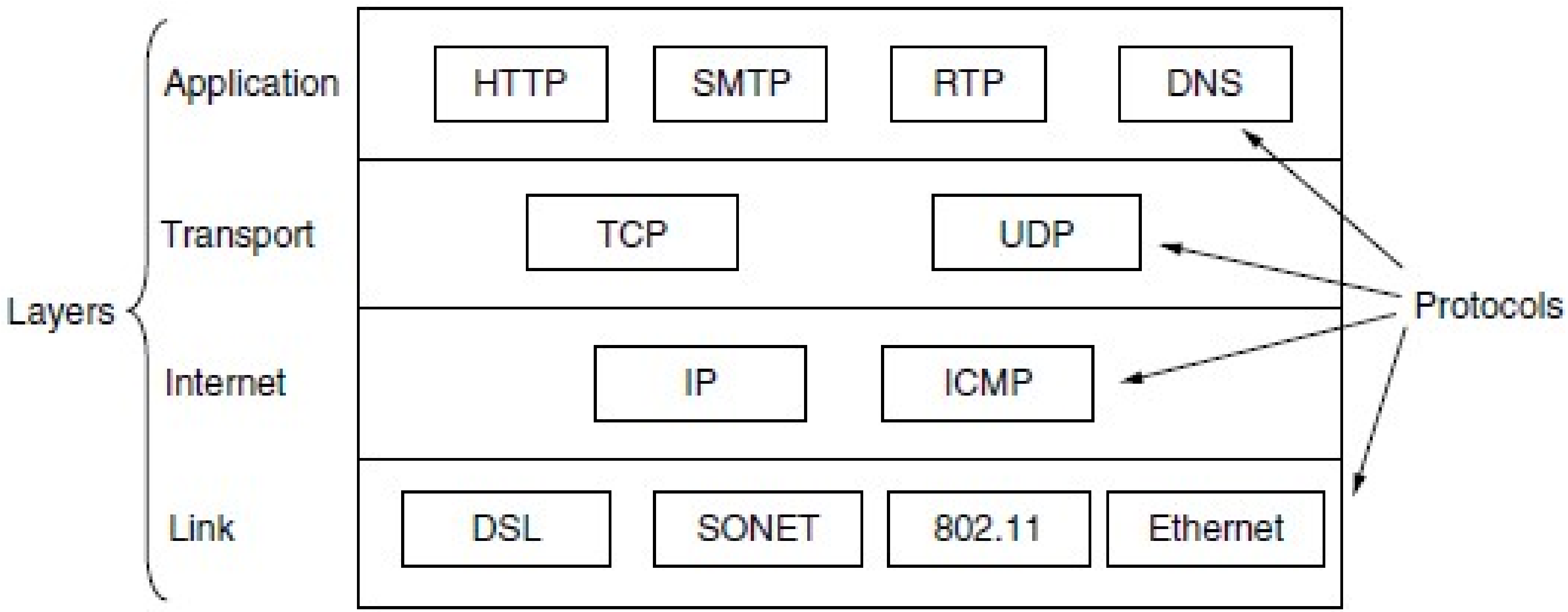


Figure 1-21. The TCP/IP reference model.


Layers
Application
Transport
Internet
Link
HTTP
SMTP
RTP
DNS
TCP
UDP
IP
ICMP
DSL
SONET
802.11
Ethernet
Protocols

Internet Protocols and Services

Quiz:

- Please define and read the following network protocols and its services:
- UDP, TCP
- HTTP, RTP, SMTP, DHCP, DNS
- IPv4, IPv6, ICMP

- The OSI and TCP/IP reference models have much in common.
- Both are based on the concept of a stack of independent protocols.

Three concepts are central to the OSI model:

- 1. Services.
- 2. Interfaces.
- 3. Protocols.

Service

- A service is formally specified by a set of **primitives** (operations) available to user processes to access the service.
- These primitives tell the service to perform some action or report on an action taken by a peer entity. If the protocol stack is located in the operating system, as it often is, the primitives are normally system calls.

Layer Interface

- A layer's *interface* tells the processes above it how to access it. It specifies what the parameters are and what results to expect. It, too, says nothing about how the layer works inside.

Protocol

- The peer *protocols* used in a layer are the layer's own business. It can use any protocols it wants to, as long as it gets the job done (i.e., provides the offered services).
- It can also change them at will without affecting software in higher layers.

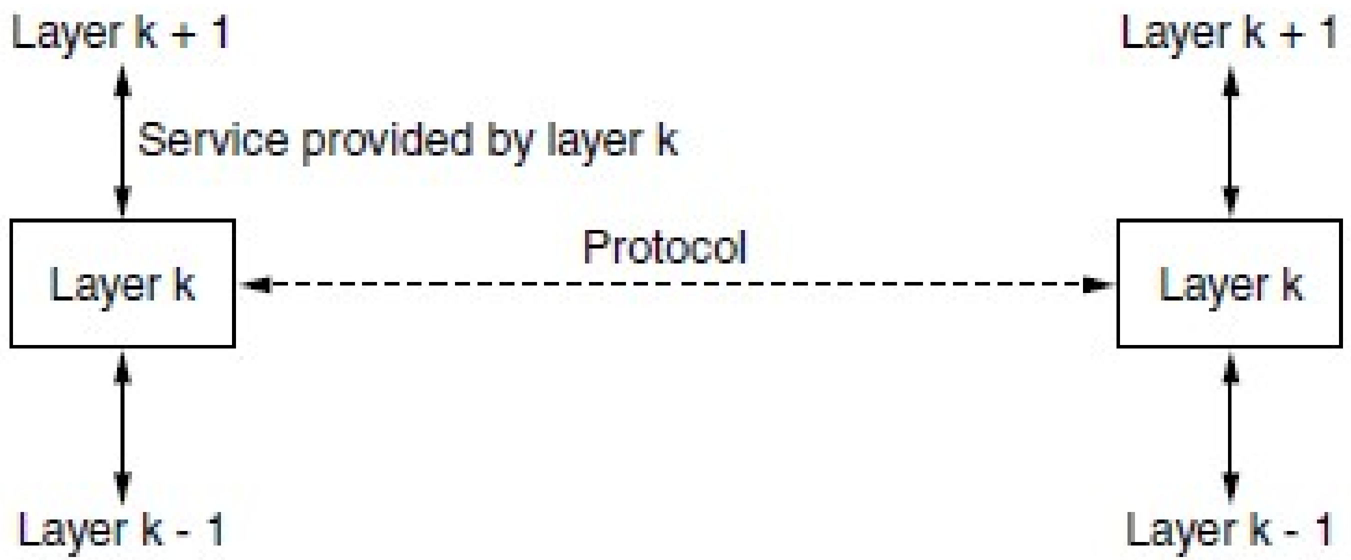
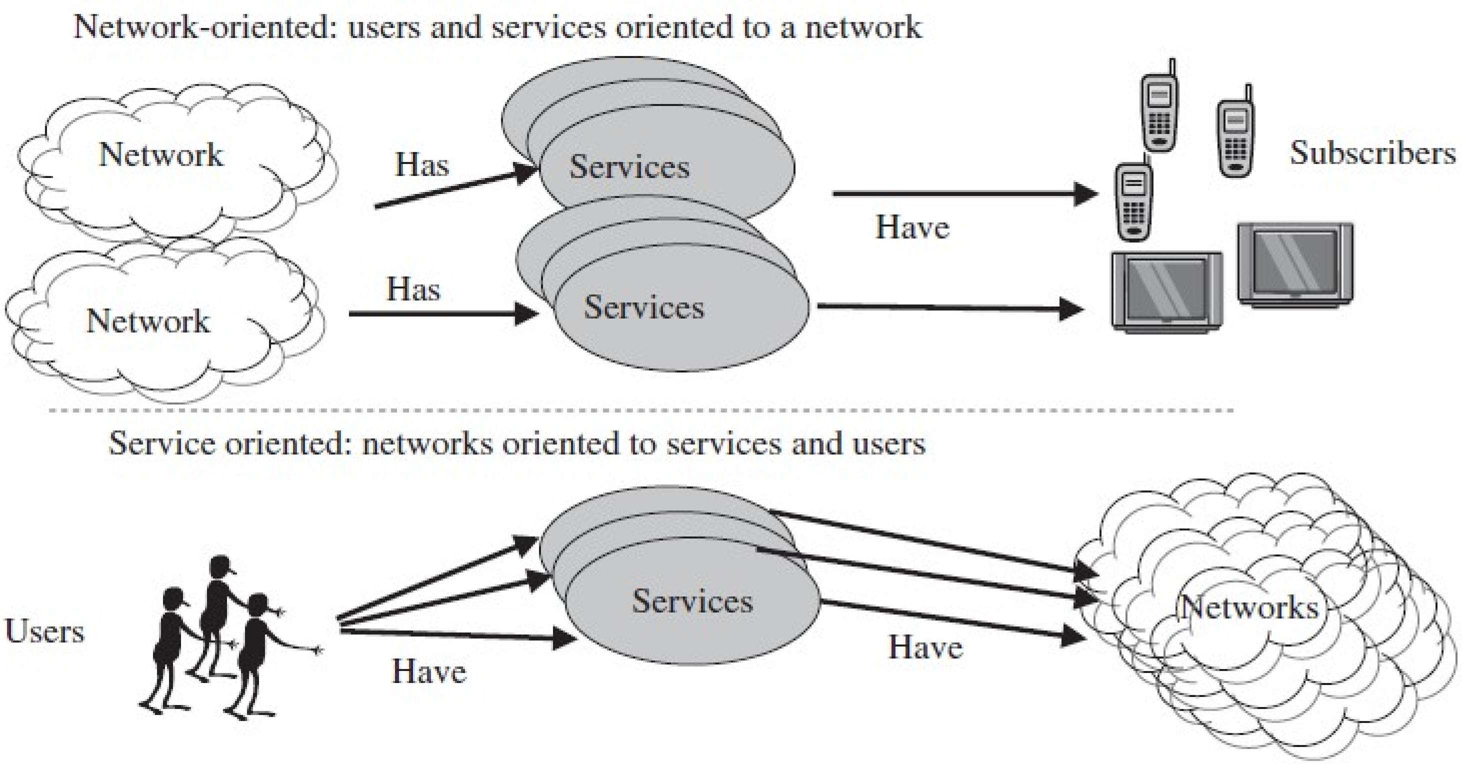


Figure 1-19. The relationship between a service and a protocol.



From network oriented service models to service oriented network models

Example

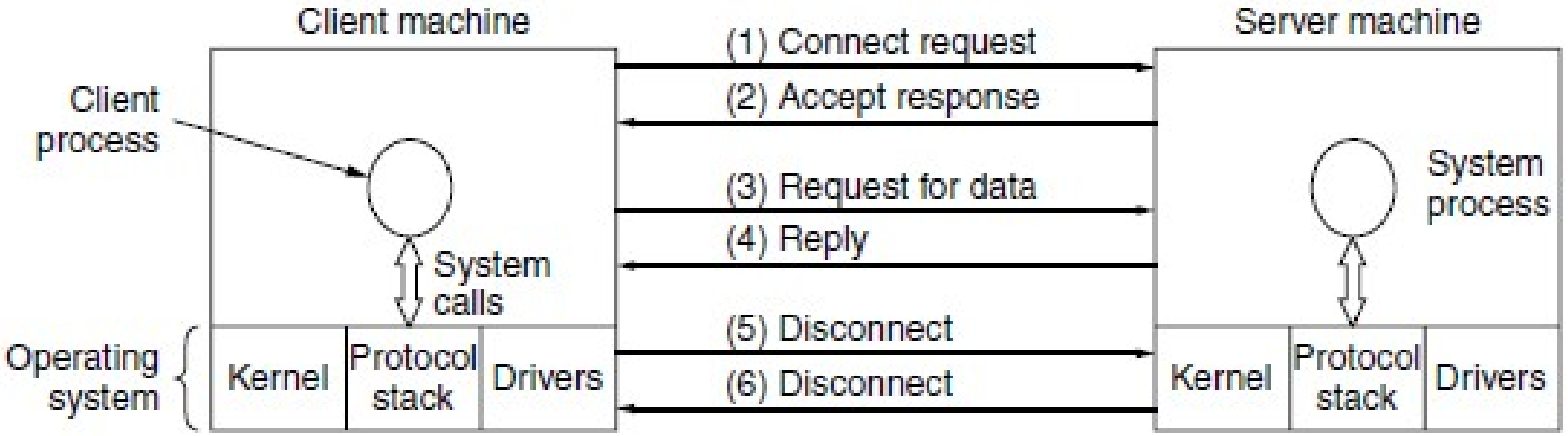
- An object, like a layer, has a set of methods (operations) that processes outside the object can invoke.
- The semantics of these methods define the set of services that the object offers.
- The methods' parameters and results form the object's interface.
- The code internal to the object is its protocol and is not visible or of any concern outside the object.

	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Movie download
	Unreliable connection	Voice over IP
Connection-less	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Text messaging
	Request-reply	Database query

Connection oriented service

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
ACCEPT	Accept an incoming connection from a peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

- TCP: provides connection oriented services


Client machine
Server machine
Client process
System calls
Operating system
Kernel
Protocol stack
Drivers
(1) Connect request
(2) Accept response
(3) Request for data
(4) Reply
(5) Disconnect
(6) Disconnect
System process
Kernel
Protocol stack
Drivers

IPv4

- Why IPv4 protocols will not be main protocols for UbiComp networks

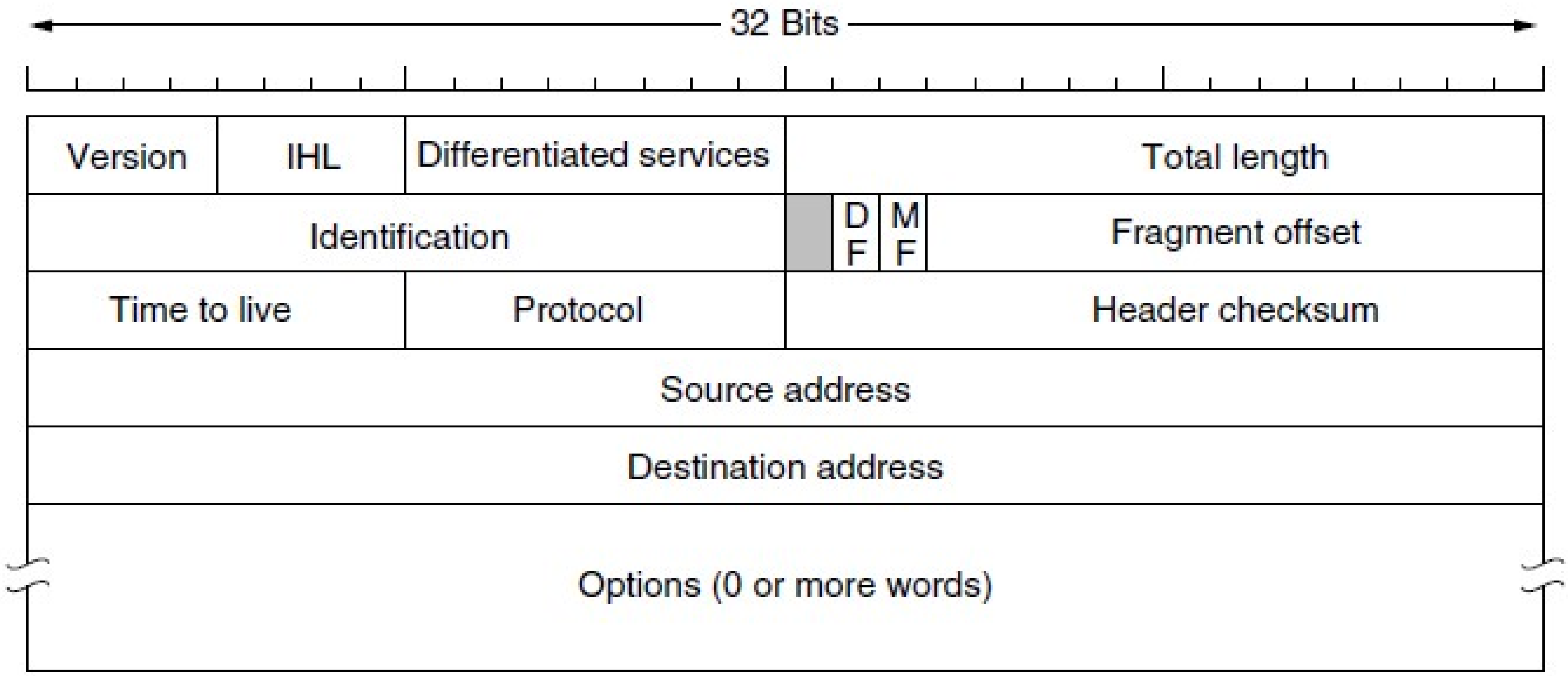


Figure 5-46. The IPv4 (Internet Protocol) header.

IPv6

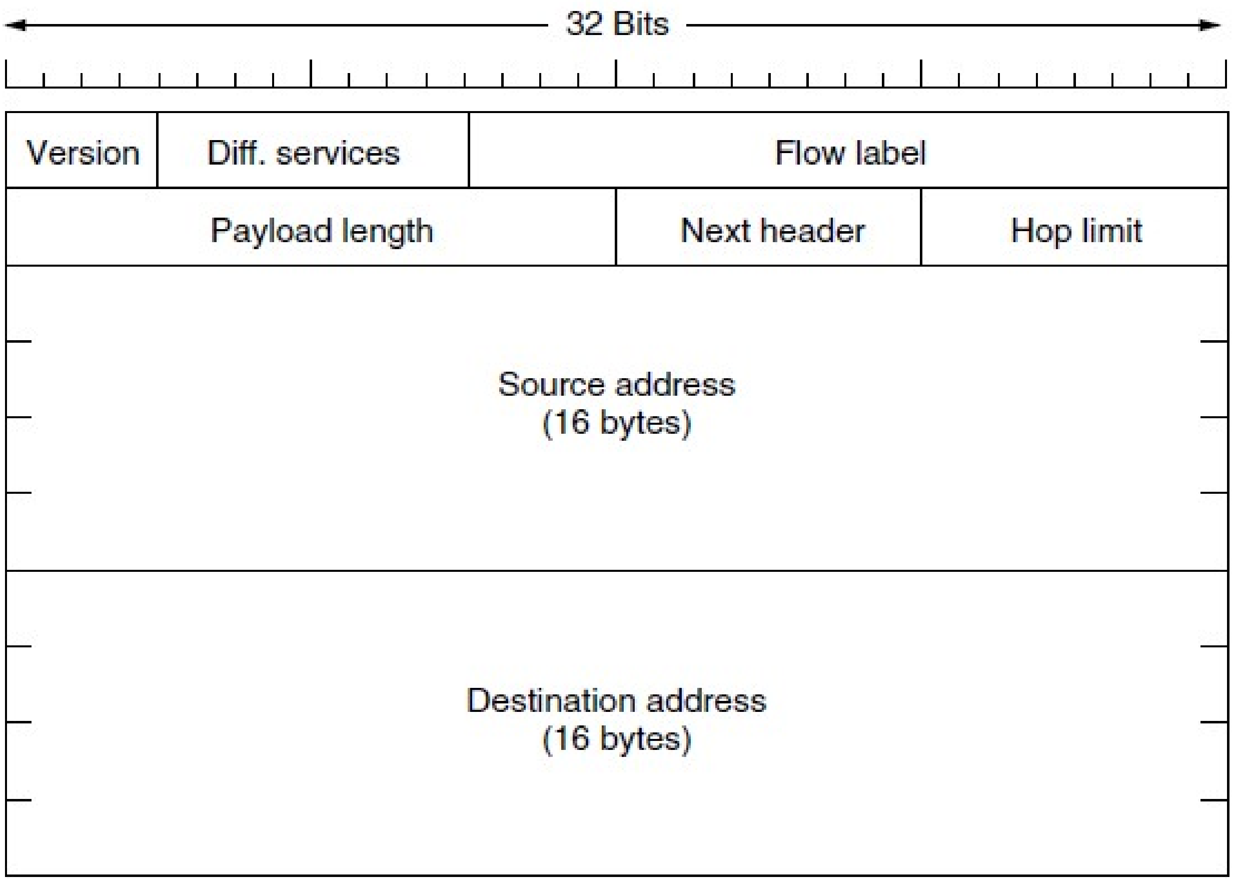


Figure 5-56. The IPv6 fixed header (required).

How to make communicate between IPv4 and IPv6

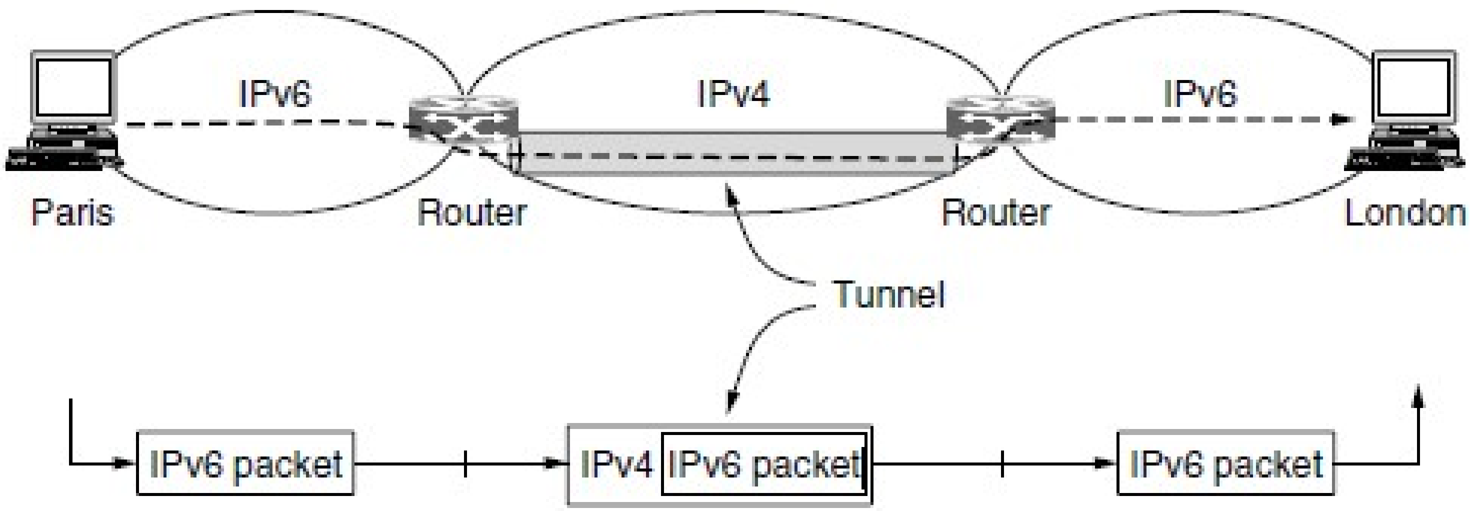


Figure 5-40. Tunneling a packet from Paris to London.

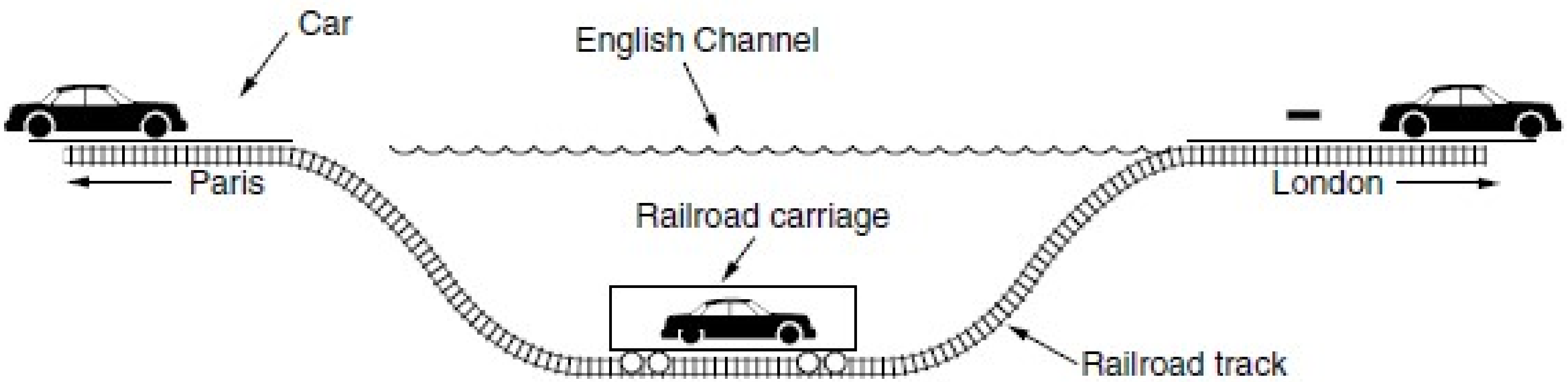


Figure 5-41. Tunneling a car from France to England.

Network Address Translation

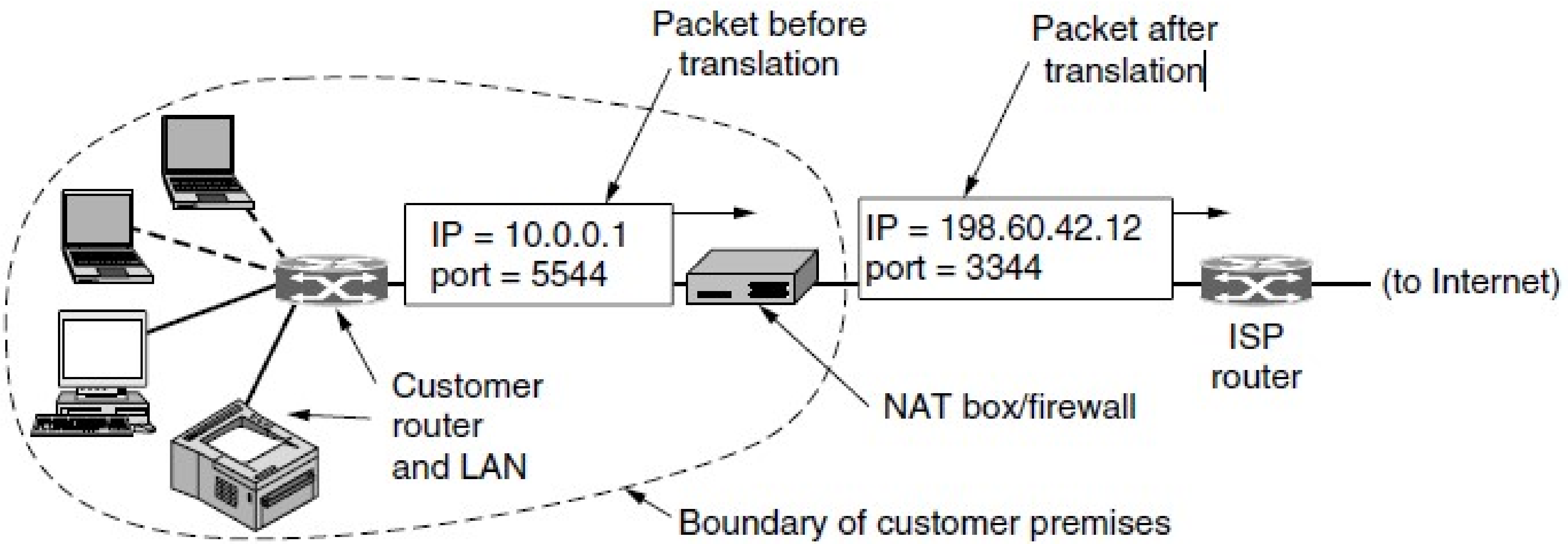
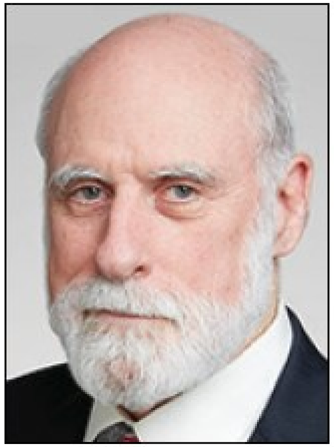


Figure 5-55. Placement and operation of a NAT box.

Future

- The likelihood that IPv6-only mobile networks might drive IPv6 deployment.
- So far, that speculation has not proven dispositive, but a 5G initiative toward IPv6-only could mark a turning point.



Vinton G. Cerf

- A growing number of devices including mobiles, Cloud servers and other Internet-enabled things are populating the Internet and will need addresses.

- It is time to lay out implementation plans and to acknowledge that some features of its design might be avoided in the interest of ubiquity. This decade’s Internet “moonshot” should be ubiquitous implementation of IPv6-only networking on a global scale.

JINI system

- Is built on the Java Remote Method Invocation system.
- The Jini system is Java-centric—because it builds on the existing Java environment and because it requires features that are widely available only within the Java platform.
- Jini adds to the infrastructure two components: the discovery protocol, which allows an entity wishing to join a Jini network to find a lookup service, and the lookup service, which acts as a place where services advertise themselves and clients go to find a service.

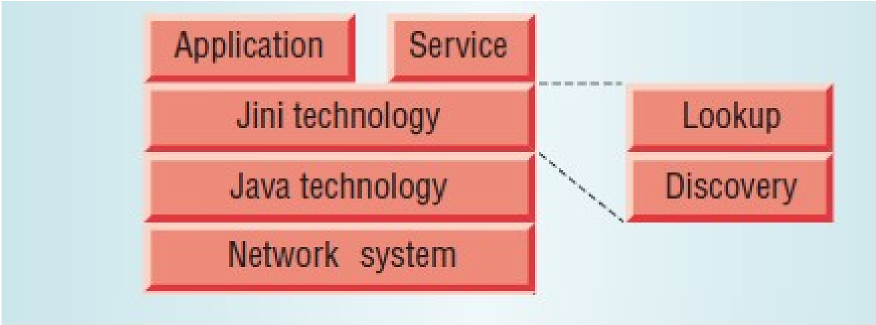
Java and Jini Technology

	Infrastructure	Programming Model	Services
Java	Java virtual machine Java Remote Method Invocation	Beans Swing graphics toolkit	Enterprise Java Beans Java naming and directory services Java transaction service
Jini	Discovery Lookup service	Leasing Transactions Distributed events	JavaSpaces Transaction manager
	Lets objects find and communicate with one another	**Adds simple APIs for remote objects and basic distributed computing**	**Everything else is a service**

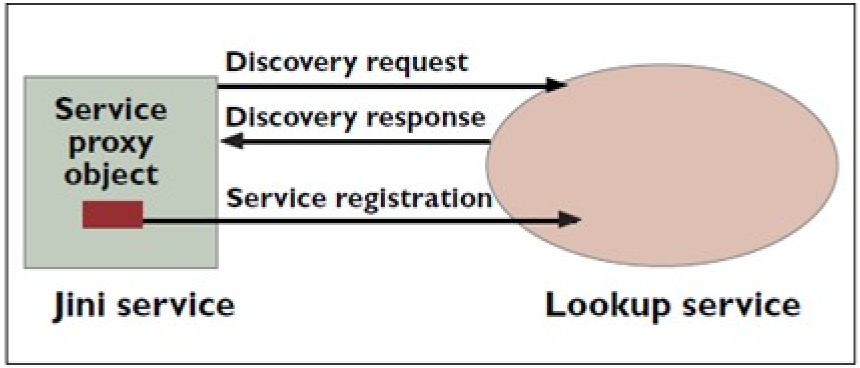
- A service (hardware or software) that wants to join a Jini federation sends out a packet, multicast over the LAN to a well-known port, asking for any lookup service to respond.
- The packet might specify that only lookup services within a particular (named) group respond, but in the simplest case, any and all lookup services on the local network would respond.
- The packet also contains the information necessary for any lookup service to respond to the requester.

Jini Architecture and services

- The service registration process, the device uploads the serialized Java object, called a *service proxy*, to the Jini lookup service.
- The lookup service is the common repository of the services offered by a network.
- Clients use this repository, which can run on any device in the network, to gain access to a particular service.

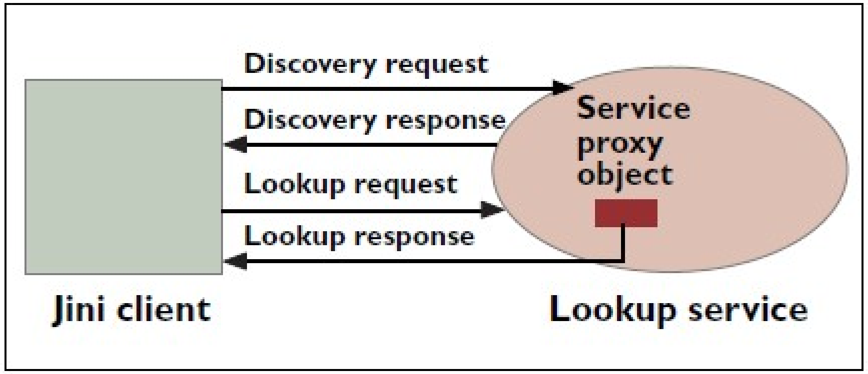


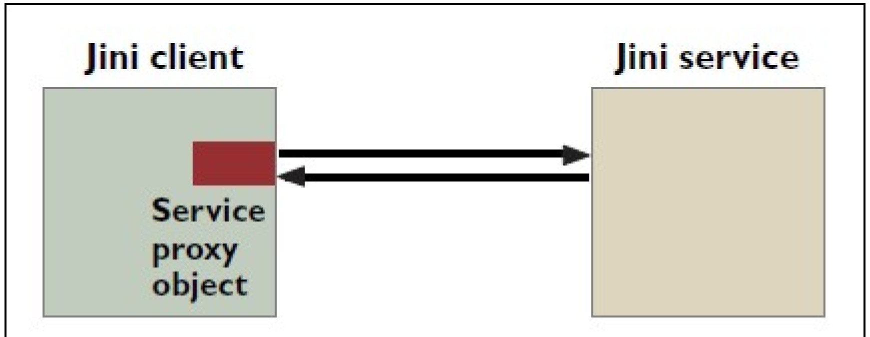
A typical Jini network architecture



Jini stores the proxy for a service in the Lookup service as part of registration

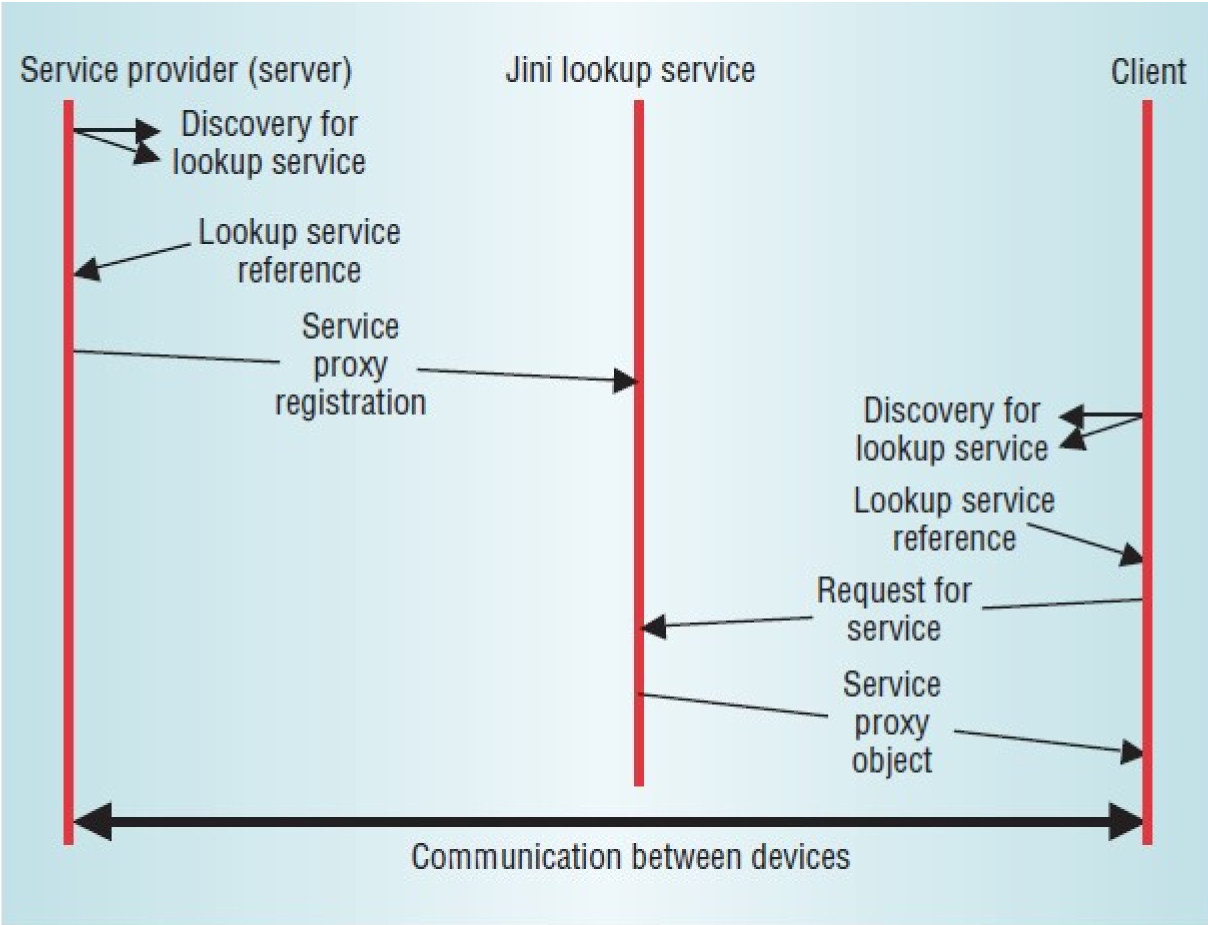
- The proxy, when loaded into the Java virtual machine running on the requester, contains enough information that, if the code needed for the proxy is not present at the site of the requester.
- All the client needs to know about the proxy is the Java interface it supports.
- This interface-based communication means that the proxy and the service can talk by way of whatever protocol they need.
- The way the proxy talks to the service can change over time without the client's needing to be altered or even to be aware of the change.





Network communication is between the proxy provided by the service and the service itself, without intervention from the lookup service.

Jini Communication



Jini event sequence diagram. The Jini lookup service is the common repository of services offered by the network.

Additional Read

- *Rahul Gupta, et al.,* Jini Home Networking: A Step toward Pervasive Computing, 2002
- Vinton Cerf, 2021 Internet Perspective, IEEE network, Jan, 2021.
- Sheets provided within the lecture