ENGR 5720G

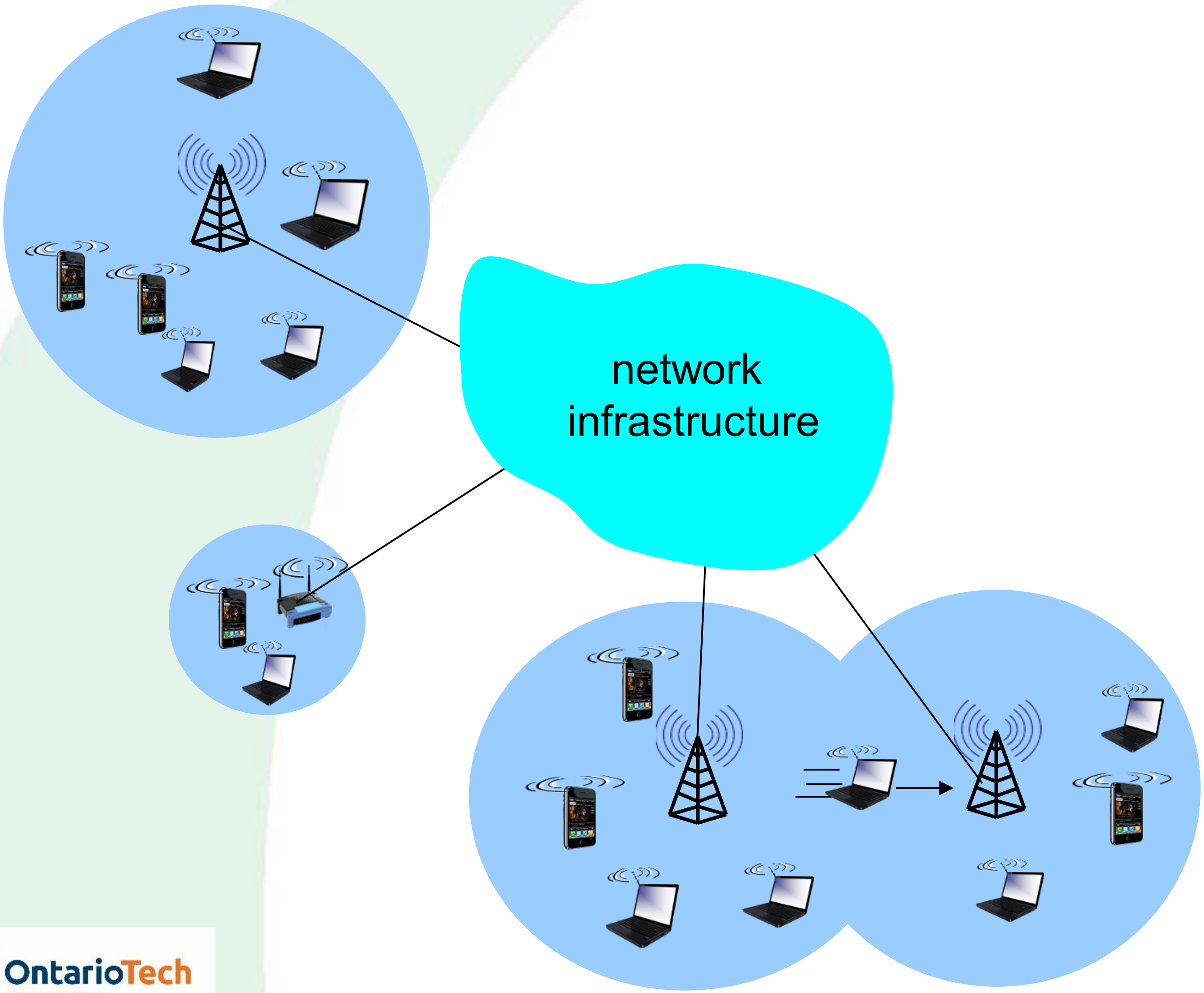
Mobile and Pervasive Computing

IoT Connectivity

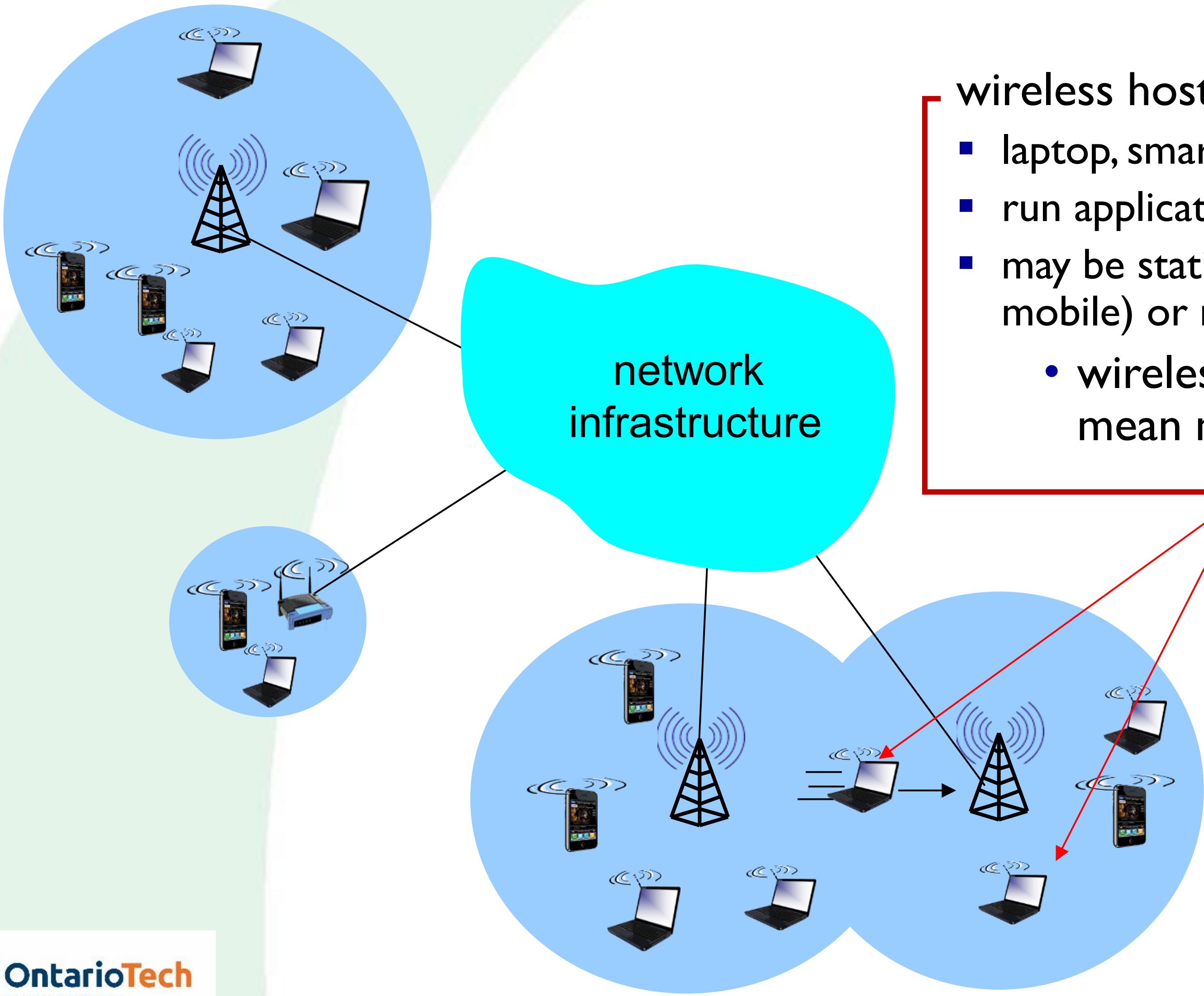
Wireless Communications

Since most connectivity in Mobile and Pervasive Applications will be wireless I will focus on wireless communications.

Elements of a wireless network



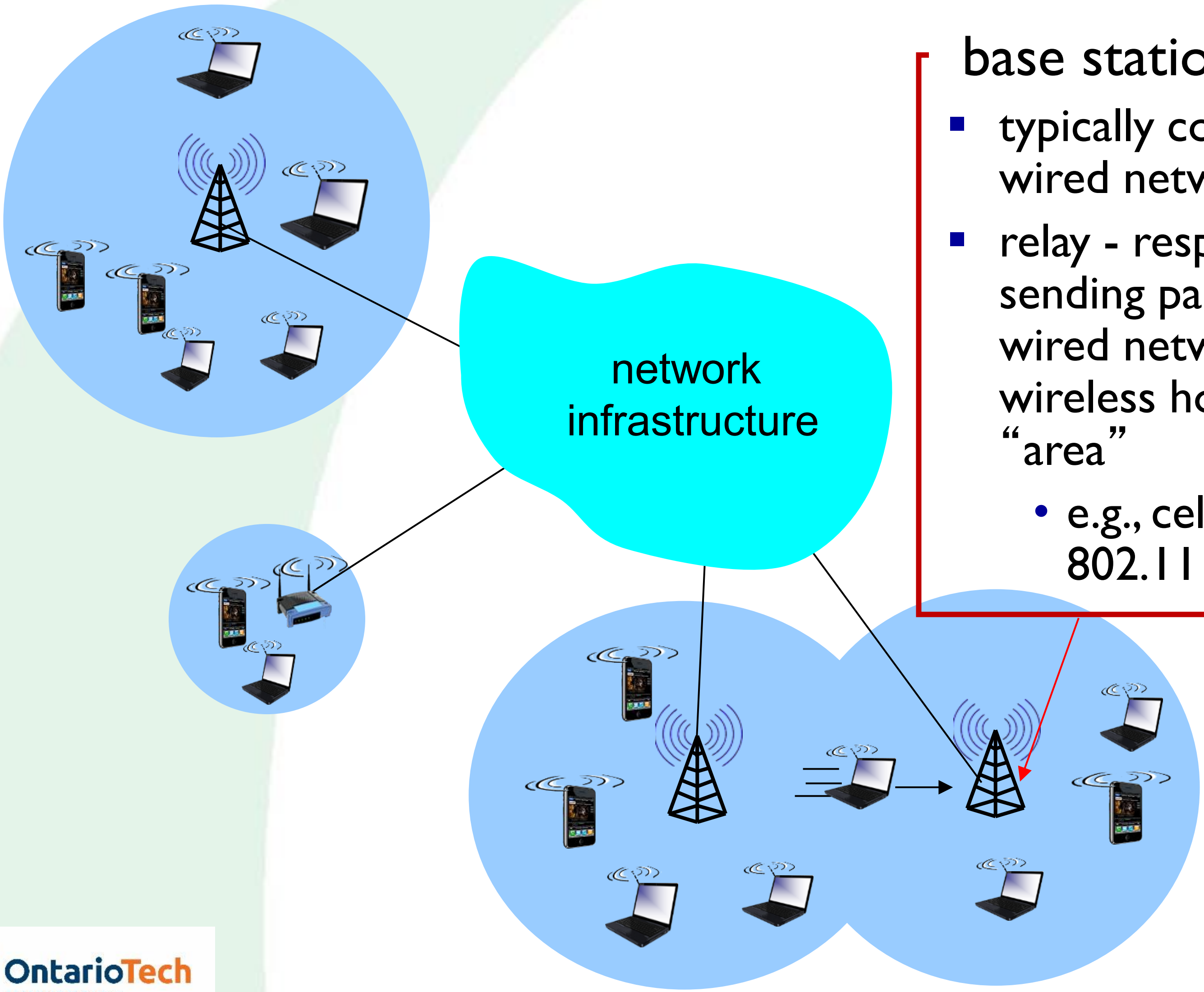
Elements of a wireless network



wireless hosts

- laptop, smartphone
- run applications
- may be stationary (non-mobile) or mobile
  - wireless does *not* always mean mobility

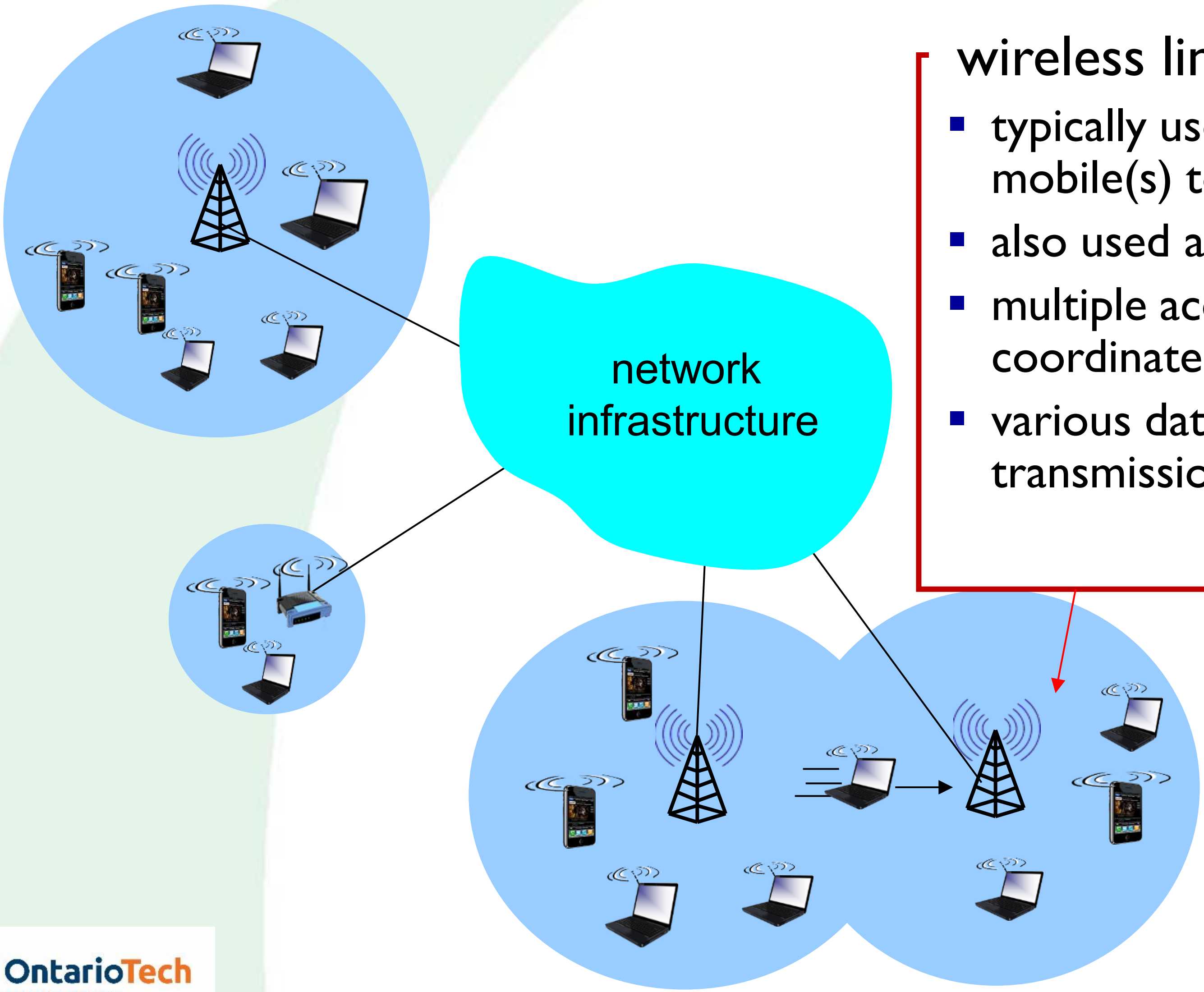
Elements of a wireless network



base station

- typically connected to wired network
- relay - responsible for sending packets between wired network and wireless host(s) in its "area"
  - e.g., cell towers, 802.11 access points

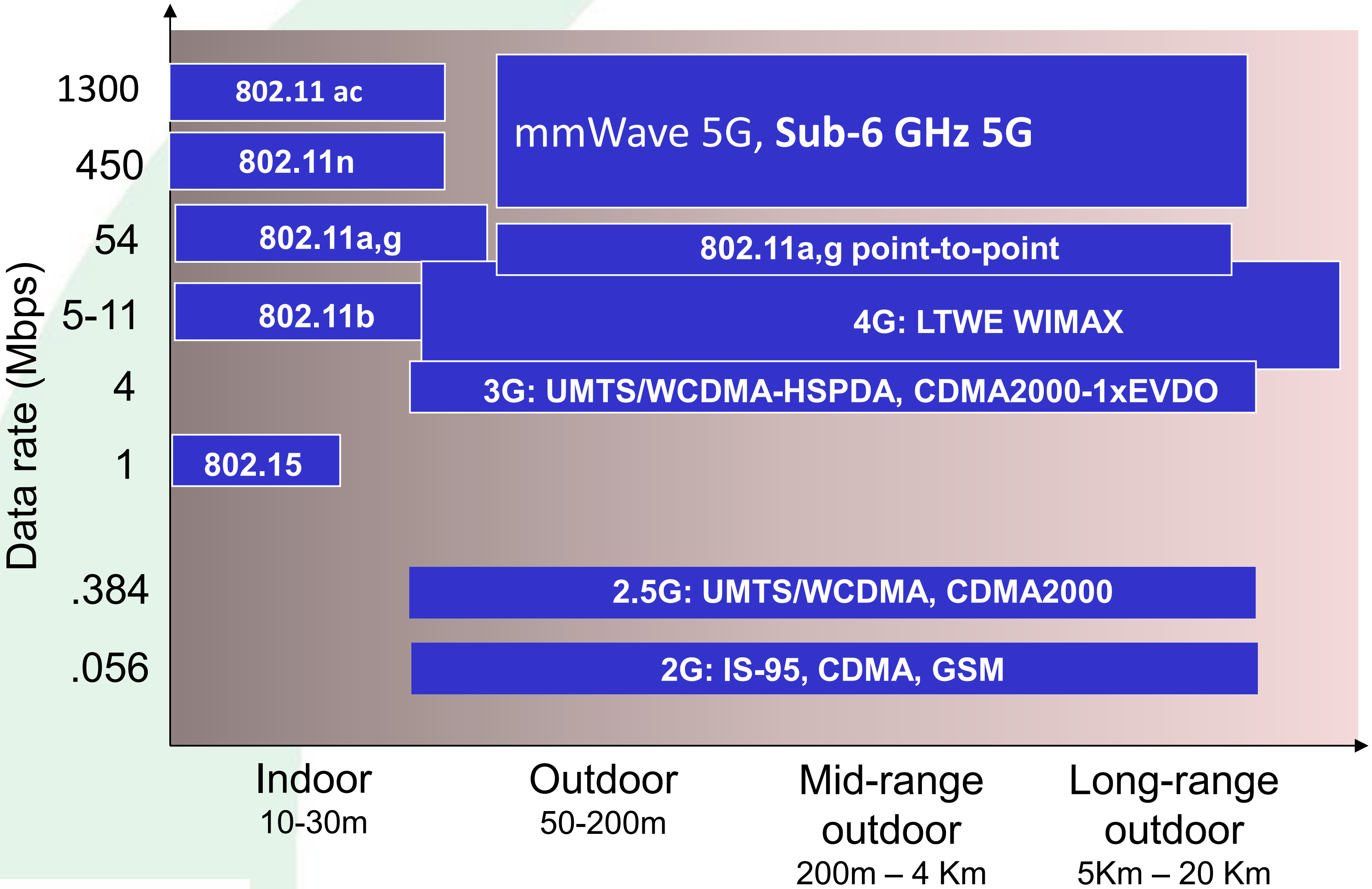
Elements of a wireless network



wireless link

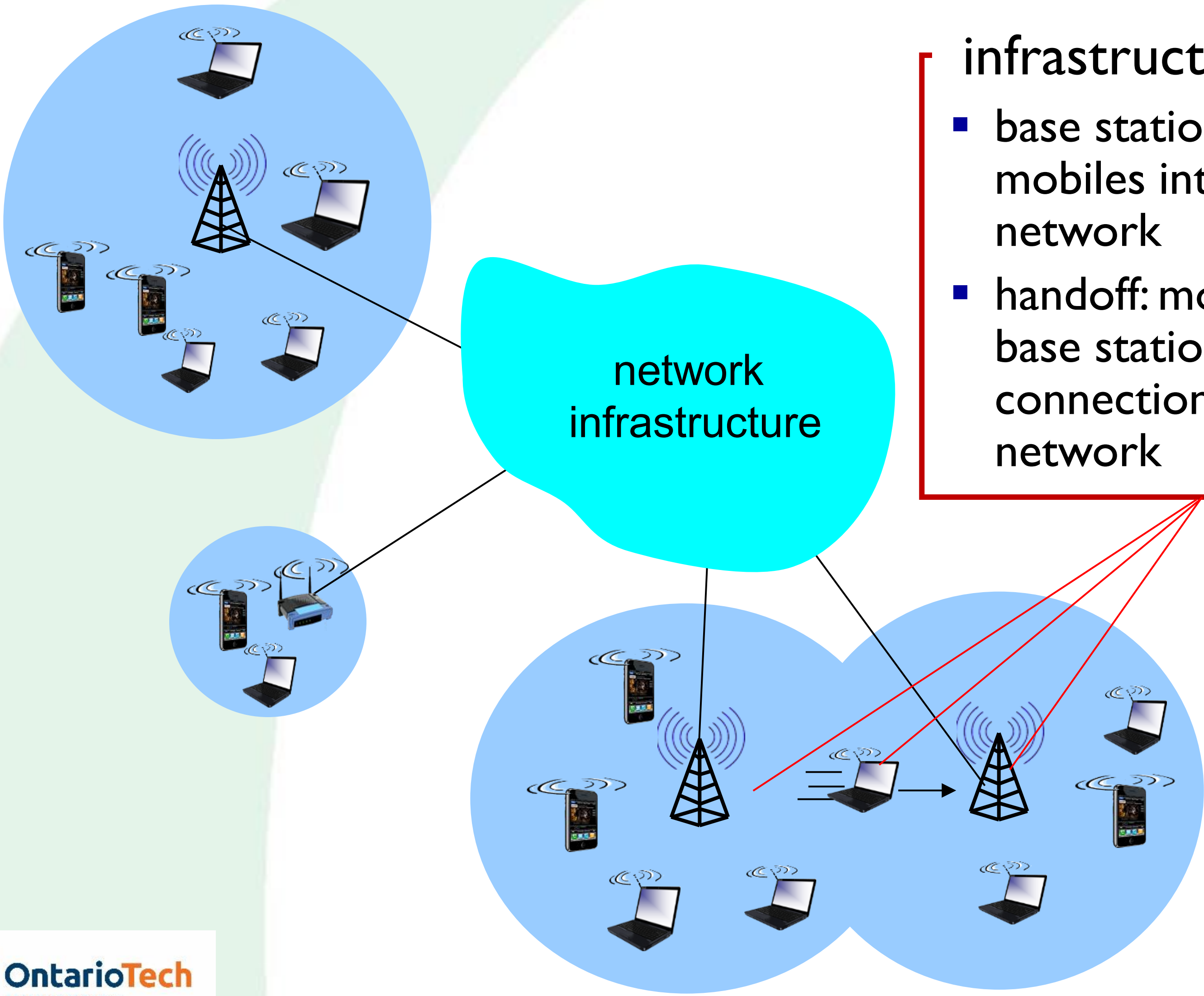
- typically used to connect mobile(s) to base station
- also used as backbone link
- multiple access protocol coordinates link access
- various data rates, transmission distance

Characteristics of selected wireless links



OntarioTech UNIVERSITY

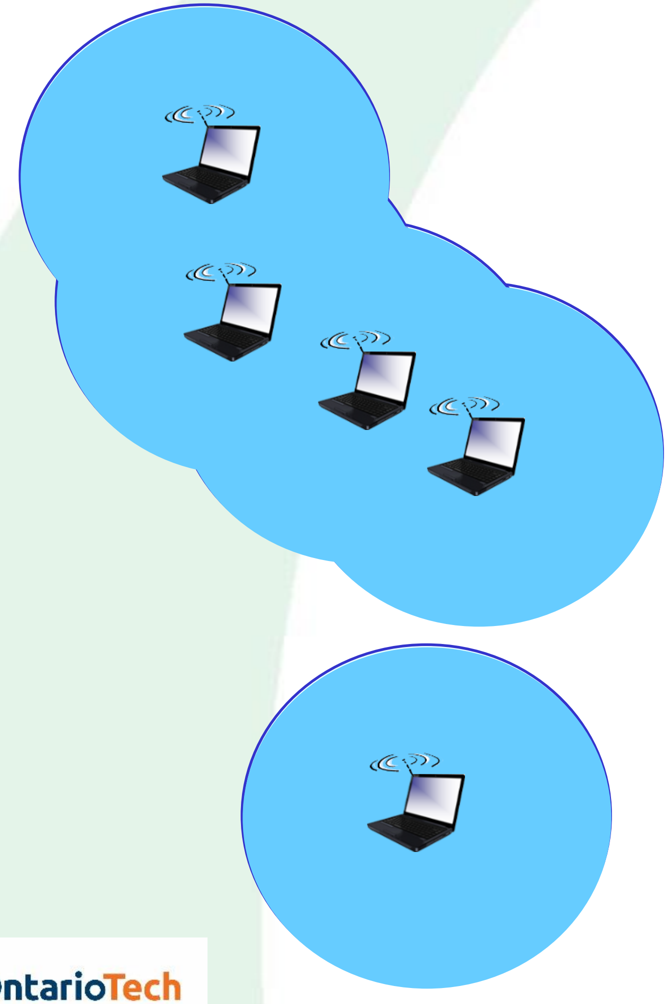
Elements of a wireless network



infrastructure mode

- base station connects mobiles into a wired network
- handoff: mobile changes base station providing connection into a wired network

Elements of a wireless network



ad hoc mode

- no base stations
- nodes can only transmit to other nodes within link coverage
- nodes organize themselves into a network: route among themselves

Wireless Link Characteristics (1)

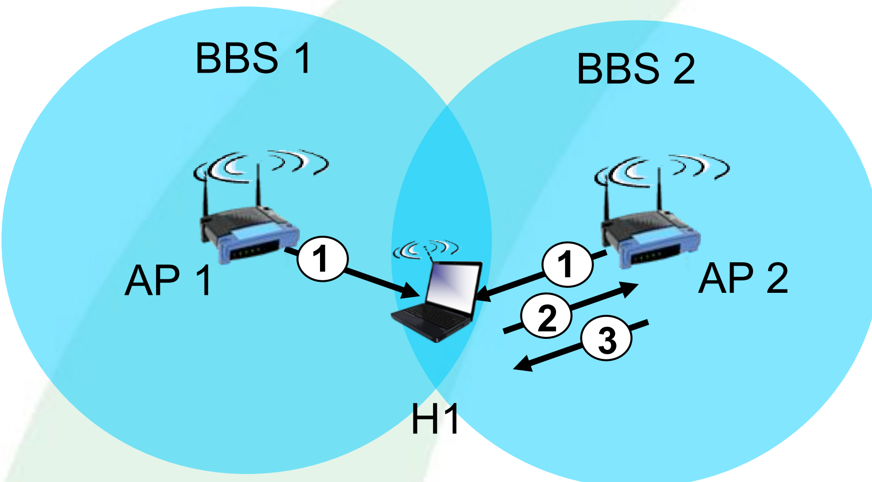
important differences from wired link ….

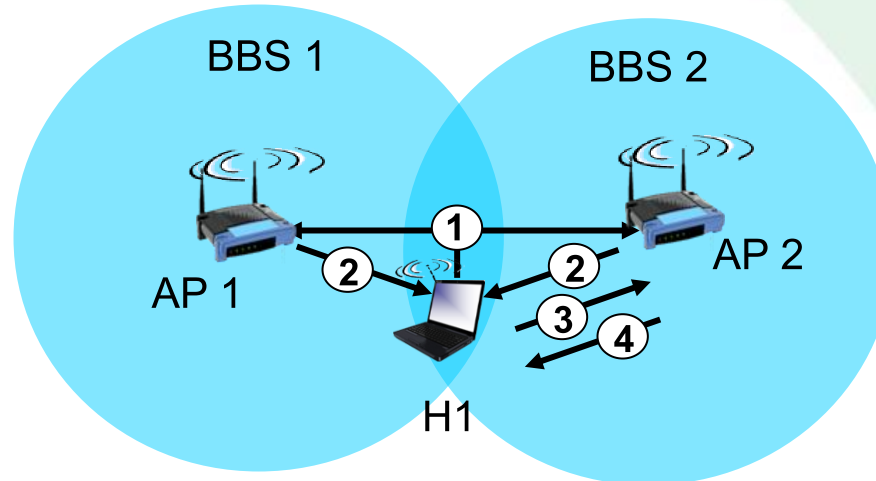
- *decreased signal strength:* radio signal attenuates as it propagates through matter (path loss)
- *interference from other sources:* standardized wireless network frequencies (e.g., 2.4 GHz) shared by other devices (e.g., phone); devices (motors) interfere as well
- *multipath propagation:* radio signal reflects off objects ground, arriving ad destination at slightly different times

…. make communication across (even a point to point) wireless link much more "difficult"

802.11 Basics – An indoor Wireless Data Protocol

802.11: passive/active scanning





passive scanning:

(1) beacon frames sent from APs

(2) association Request frame sent: H1 to selected AP

(3) association Response frame sent from selected AP to H1

active scanning:

(1) Probe Request frame broadcast from H1

(2) Probe Response frames sent from APs

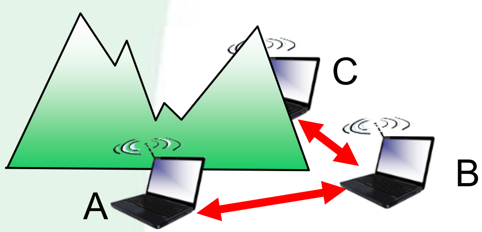
(3) Association Request frame sent: H1 to selected AP

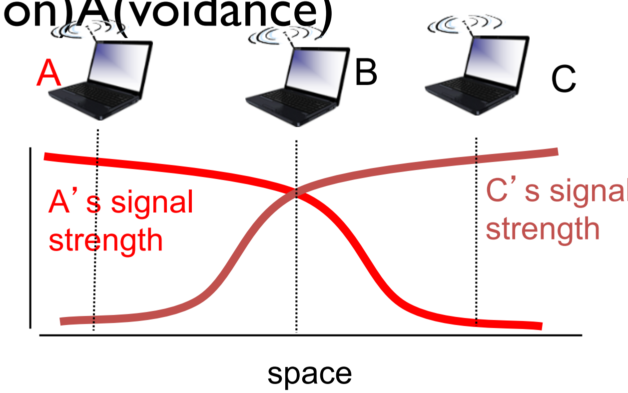
(4) Association Response frame sent from selected AP to H1

OntarioTech UNIVERSITY

IEEE 802.11: multiple access

- avoid collisions: 2$^+$ nodes transmitting at same time
- 802.11: CSMA (**Carrier Sense Multiple Access) -** sense before transmitting
  - don't collide with ongoing transmission by another node
- 802.11: *no* collision detection!
  - difficult to receive (sense collisions) when transmitting due to weak received signals (fading)
  - can't sense all collisions in any case: hidden terminal, fading
  - goal: *avoid collisions:* CSMA/C(ollision)A(voidance)





IEEE 802.11 MAC Protocol: CSMA/CA

802.11 sender

1 if sense channel idle for **DIFS (Distributed Interframe Space)** then

transmit entire frame (no CD)

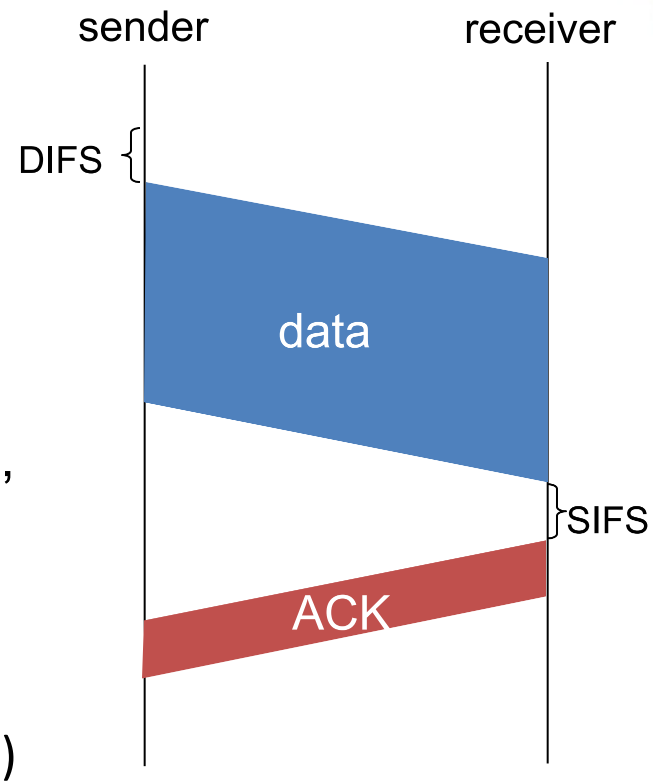
2 if sense channel busy then

start random backoff time

timer counts down while channel idle

transmit when timer expires

if no ACK, increase random backoff interval, repeat 2



802.11 receiver

- if frame received OK

return ACK after **SIFS (**Short Interframe Space)
(ACK needed due to hidden terminal problem)

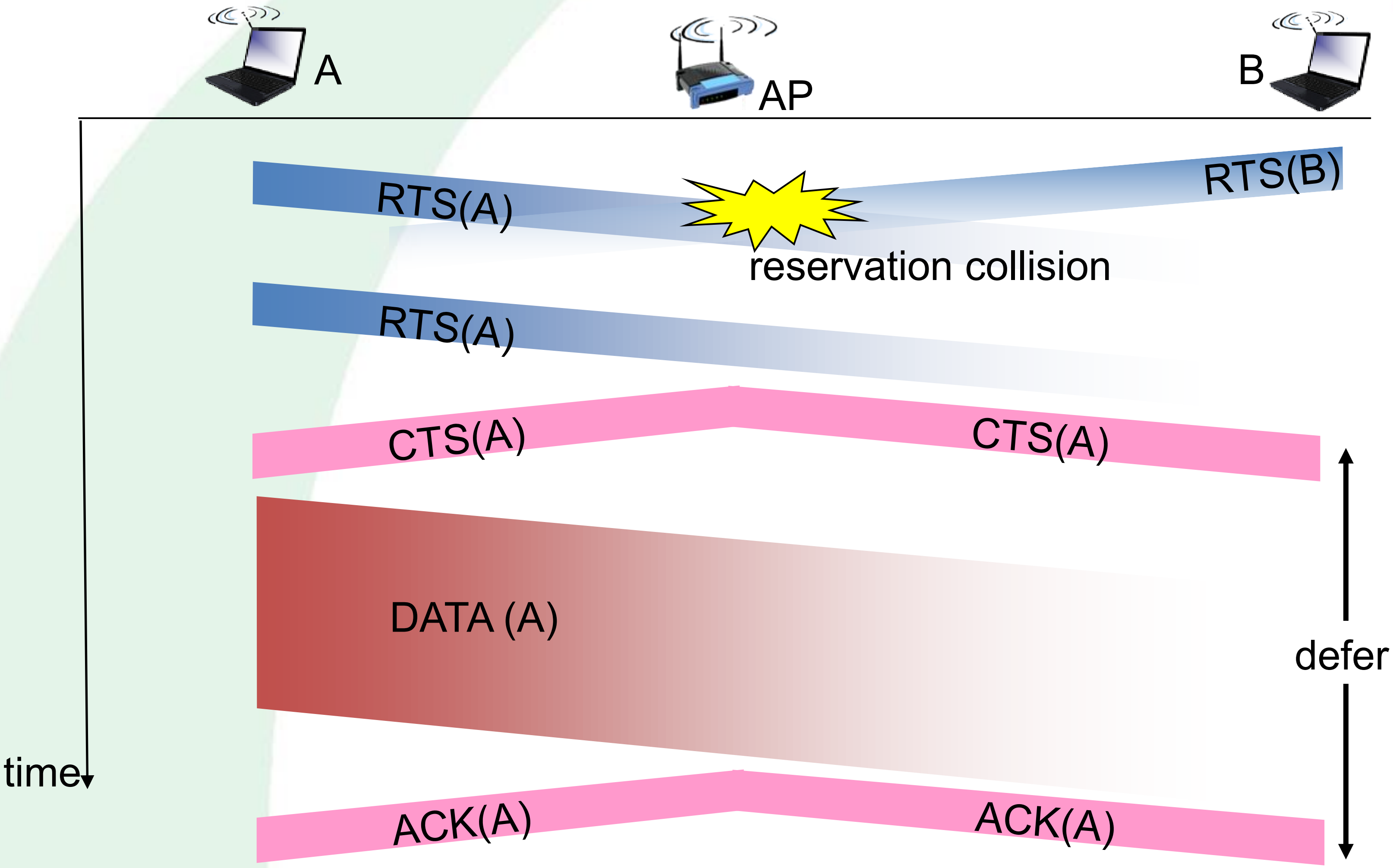
Avoiding collisions (more)

idea: allow sender to "reserve" channel rather than random access of data frames: avoid collisions of long data frames

- sender first transmits *small* request-to-send (RTS) packets to BS (**Base Station**) using CSMA
  - RTSs may still collide with each other (but they're short)
- BS broadcasts clear-to-send CTS in response to RTS
- CTS heard by all nodes
  - sender transmits data frame
  - other stations defer transmissions

avoid data frame collisions completely using small reservation packets!

Collision Avoidance: RTS-CTS exchange



Ontario Tech University

IoT Protocol Stack

- Infrastructure (ex: 6LowPAN, IPv4/IPv6, RPL)
- Identification (ex: EPC, uCode, IPv6, URIs)
- Comms / Transport (ex: **Wifi**, **Bluetooth**, LPWAN)
- Discovery (ex: Physical Web, mDNS, DNS-SD)
- Data Protocols (ex: **MQTT**, **CoAP**, **AMQP**, Websocket, Node)
- Device Management (ex: TR-069, OMA-DM)
- Semantic (ex: JSON-LD, Web Thing Model)

https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about

Web vs. IoT

Web
Hundreds / thousands of bytes

XML
HTTP
TLS
TCP
IPv6

- Inefficient content encoding
- Huge overhead, difficult parsing
- Requires full Internet devices

Internet of Things
Tens of bytes

Web Objects
CoAP
DTLS
UDP
6LoWPAN

- Efficient objects
- Efficient Web
- Optimized IP access

LTE (Long Term Evaluation)Basics
Wireless Protocol for Mobile Devices

Why is LTE important to us?

- LTE or the futuristic 5G network is the most ubiquitous network available worldwide.
- More likely most IoT mobile applications will have to take advantage of LTE communications.
- LTE supports mobility, security and authentication though SIM cards.

LTE Key Features

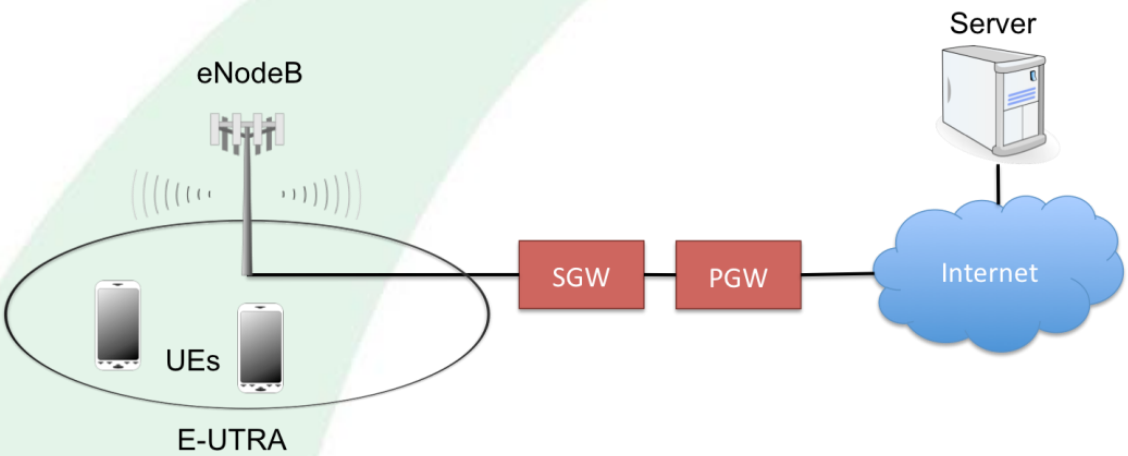
- High Spectral Efficiency→ more customers, less costs
- Co-existence with other standards
- Flexible radio planning (cell size of 5km→30/100km)
- Reduced Latency→ less RTT, multi-player gaming, audio/video conferencing
- Reduced costs for operators operational expenditures (OPEX) & capital expenditures (CAPEX) for mobile network operators
- Increased data rates via enhanced air interface (OFDMA,SC-FDMA,MIMO)
- All-IP environment→ SAE or EPC

OntarioTech UNIVERSITY

4G Long Term Evolution (LTE)

- Long Term Evolution (LTE) – Standard created by the 3rd Generation Partnership Project
  - Deployed globally
  - All packet switched network
  - High throughput and QoS considerations
  - Provides wireless retransmissions of lost data

LTE Network



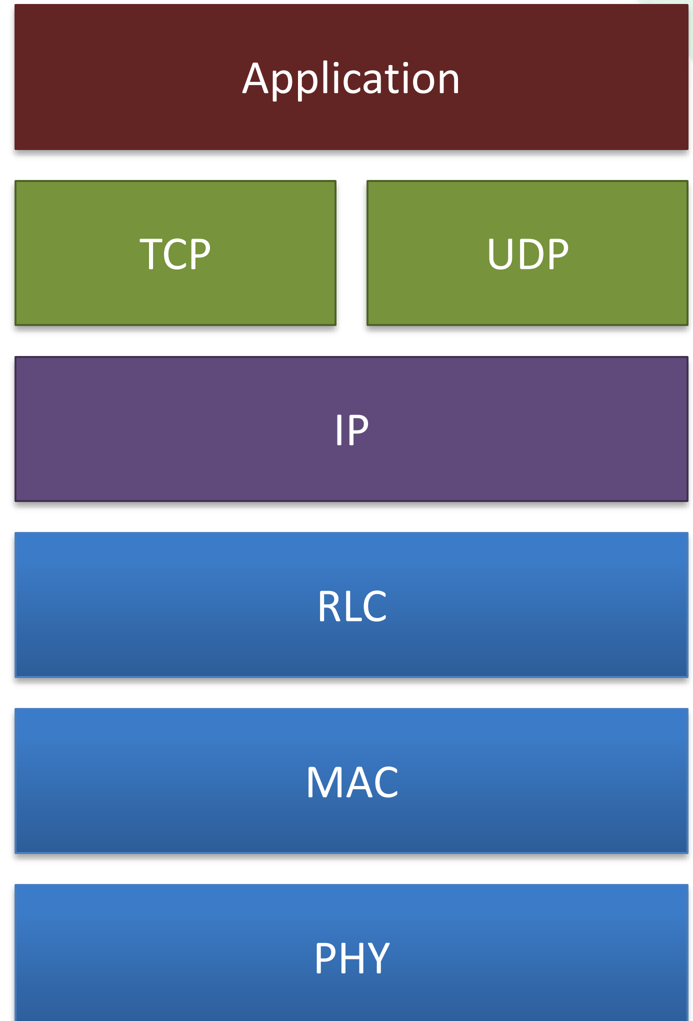
- Packet Delivery Network Gateway (PGW)
  - Connects LTE network to IP networks
- Serving Gateway (SGW)
  - Route packets to and from wireless access points
- Enhanced Node B (eNodeB)
  - Wireless access point
- User Equipment (UE)
  - End user devices

LTE Network Architecture

- LTE is the packet switched protocol for cellular networks.
- From our perspective we see the physical layer as a black box but must be aware of the QoS capabilities of LTE.
- Because of the adoption of IP across all network types we focus on IP protocols.

LTE Network Layers

- LTE wireless network has 6 layers (not all shown)
- PHY – RLC layers carry user and network control data
- LTE transmits collections of Physical Resource Blocks (PRB)s in transport blocks



OntarioTech UNIVERSITY

Cellular IoT

Low Power Wide Area Networks

LPWA to Represent 70% of Cellular IoT Connections by 2020

Market Segment	Connections in 2020 (Billion)	Requirements	Technology
• CCTV(Camera) • In-vehicle Entertainment…	0.2B	l >10Mbps	3G/4G
• IoT Gateway Backhaul • Wearable • …	0.8B	l ~1Mbps l Low power consumption	2G/3G/Cat-1 Cat-M1
• Sensors, Meters • Asset Tracking • Smart Parking • Smart agriculture …	2B	l Low Throughput (<100kbps) l Deep Coverage (20dB) l Low power (10 Years) l Low cost (<$5)	Sigfox, LoRa NB-IoT

LPWA: Low Power Wide Area

LPWA Technologies Overview

	Sigfox	LoRa	EC-GSM	Cat-1	Cat-0	eMTC	NB-IoT
Standardization	Private	Open	3GPP	3GPP	3GPP	3GPP	3GPP
Spectrum	Unlicensed	Unlicensed	Licensed	Licensed	Licensed	Licensed	Licensed
Channel BW	100Hz	7.8~500kHz	200kHz	1.4~20MHz	1.4~20MHz	1.4MHz	180KHz
System BW	100KHz	125kHz	1.4MHz	1.4~20MHz	1.4~20MHz	1.4MHz	180KHz
Peak Data Rate	UL:100bps DL:600bps	180bps~37.5kbps	DL: 74kbps UL:74kbps	DL:10Mbps UL:5Mbps	DL:2Mbps UL:1Mbps	DL:800kbps UL:1Mbps	DL:234.7kbps UL:204.8kbps
Max. number of Message per day	140(Device) 50000(BTS)	50000(BTS)	unlimited	unlimited	unlimited	unlimited	unlimited
Device Peak Tx Power	14dBm	14dBm	26dBm	23dBm	23dBm	23dBm	23dBm
MCL(Maximum Coupling Loss)	UL:156dB DL: 147dB	UL: 156dB DL: 168(SF12, BW7.8) 132(SF6, BW125)	164dB	144dB	144dB	156dB	164dB
Device Power Consumption	Low	Low-Medium	Low	Medium	Medium	Low-Medium	Low

OntarioTech UNIVERSITY

C-IoT (Cellular IoT) is designed to provide wide WAN coverage

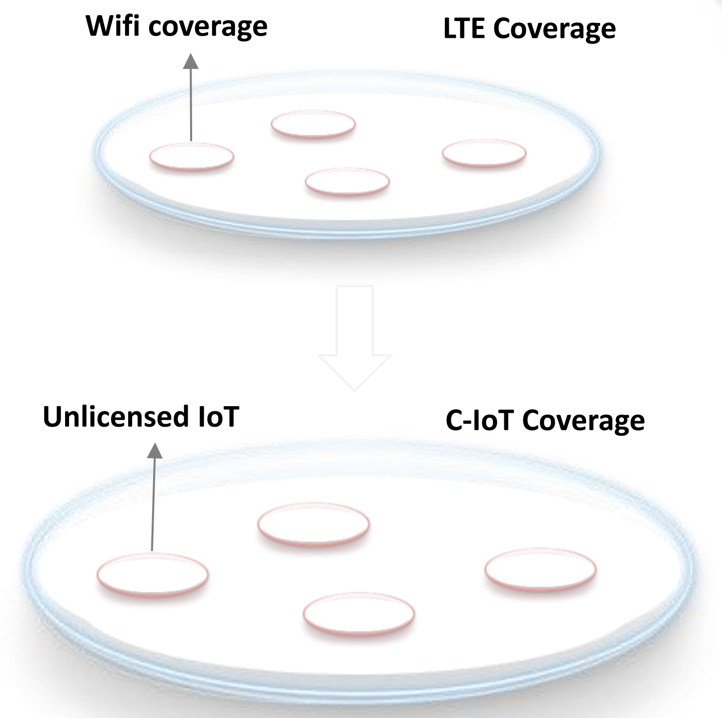
C-IoT

Re-use existing Cellular network

Carrier-grade Reliability

4G-Like Security

Roaming



- Unlicensed technology is for local coverage
- C-IoT is for wide coverage

OntarioTech UNIVERSITY

Bluetooth Low Energy (BLE)

Introduction

- Bluetooth is *a way for devices to wirelessly communicate over short distances*
- Bluetooth low energy **(BLE)** was introduced back in 2011 as the hallmark feature of Bluetooth v4.0
- What distinguishes Bluetooth is its special attention to short-distance communication, low-power consumption
- Usually less than 30 ft. Both hardware and software are affected by this special attention
- Bluetooth is not the solution for every communication needs, but it's the right technology at the right time

Bluetooth Low Energy (BLE)

- BLE has extremely low-power characteristics make it one of the most suitable solutions to enable wireless communications among battery powered IoT objects ubiquitously deployed in the field with the aim of building smart environments

Technology prevalence

- Forecasts show that Bluetooth low energy technology will be implemented in billions of products within just a few years:
  - Phone Accessories > 10 billions
  - Smart Energy (counters and displays) ~ 1 billions
  - Home Automation > 5 billions
  - Health, Wellness, Sports & Fitness > 10 billions
  - Assisted Living > 5 billions
  - Animal Tagging ~ billions
  - P2P Intelligent Transport Systems > 1 billion
  - Industrial Automation/M2M > 10 billions

OntarioTech UNIVERSITY

BLE Cont'd

- Bluetooth low energy is ideal for applications requiring episodic or periodic transfer of small amounts of data
- Therefore, BLE is especially well suited for sensors, actuators and other small devices that require extremely low power consumption
- Bluetooth low energy features:
  - Works well with high numbers of communication nodes with limited latency requirements
  - Very low power consumption
  - Robustness equal to Classic Bluetooth
  - Short wake-up and connection times
  - Good smartphone and tablet support
  - Cost efficient and compatible

Bluetooth vs. BLE

Bluetooth	BLE
2.4 GHz – 79 Ch, 1MHZ Bandwidth	2.4 GHz – 40 Ch, 2MHz
Bluetooth was originally designed for continuous, streaming data applications (lots of data at close range).	Bluetooth 4.0's low power consumption, applications can run on a small battery for four to five years.
Wireless headsets File transfers between devices Wireless keyboards and printers Wireless speakers	Blood pressure monitors Fibit-like devices Industrial monitoring sensors Geography-based, targeted promotions (iBeacon) Public transportation apps

OntarioTech UNIVERSITY

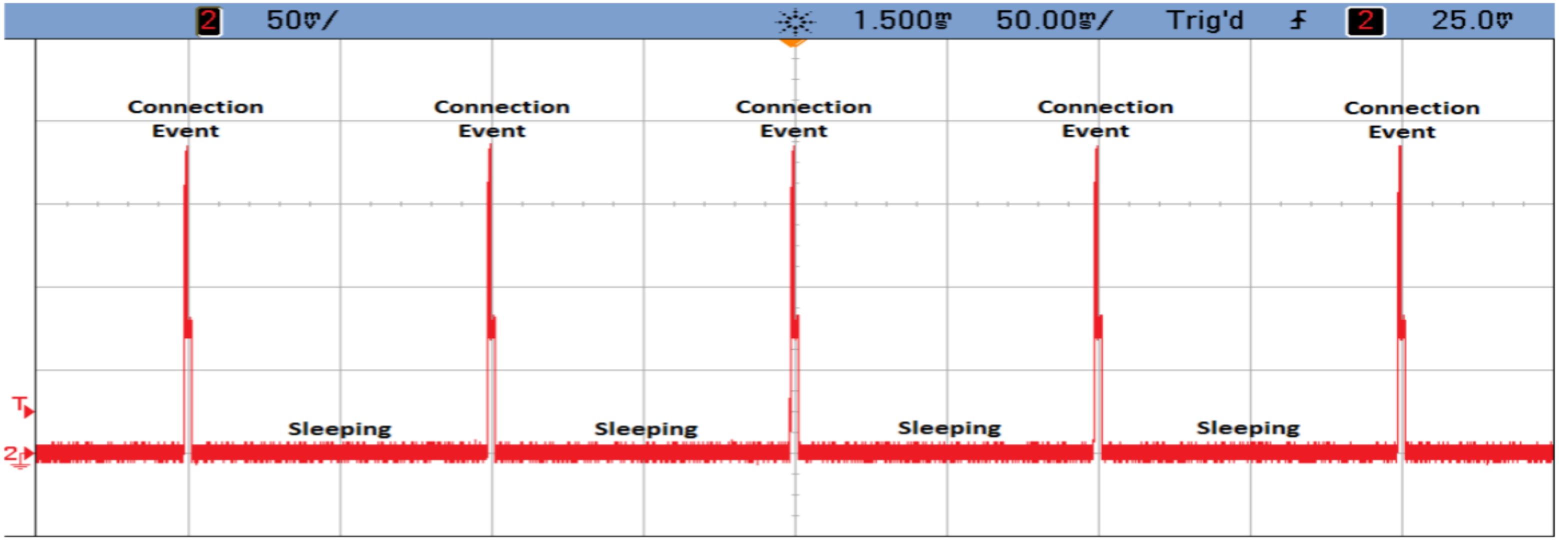
Bluetooth vs. BLE

Specification	Bluetooth	BLE
Network/Topology	Scatternet	Star Bus
Power consumption	Low (less than 30 mA)	Very Low (less than 15 mA)
Speed/Data rate	700 Kbps	1 Mbps
Range	<30 m	50
RF Frequency band	2.4 GHz	2.4 GHz
Frequency Channels	79 channels from 2.400 GHz to 2.4835 GHz with 1 MHz spacing	40 channels from 2402MHz to 2480 MHz (includes 3 advertising and 37 data channels)
Modulation	GFSK (modulation index 0.35) , π/4 DQPSK, 8DPSK	GFSK (modulation index 0.5)
Latency in data transfer	Approx. 100 ms	Approx. 3
Link layer	TDMA	TDMA
Nodes/Active Slaves	7	Unlimited

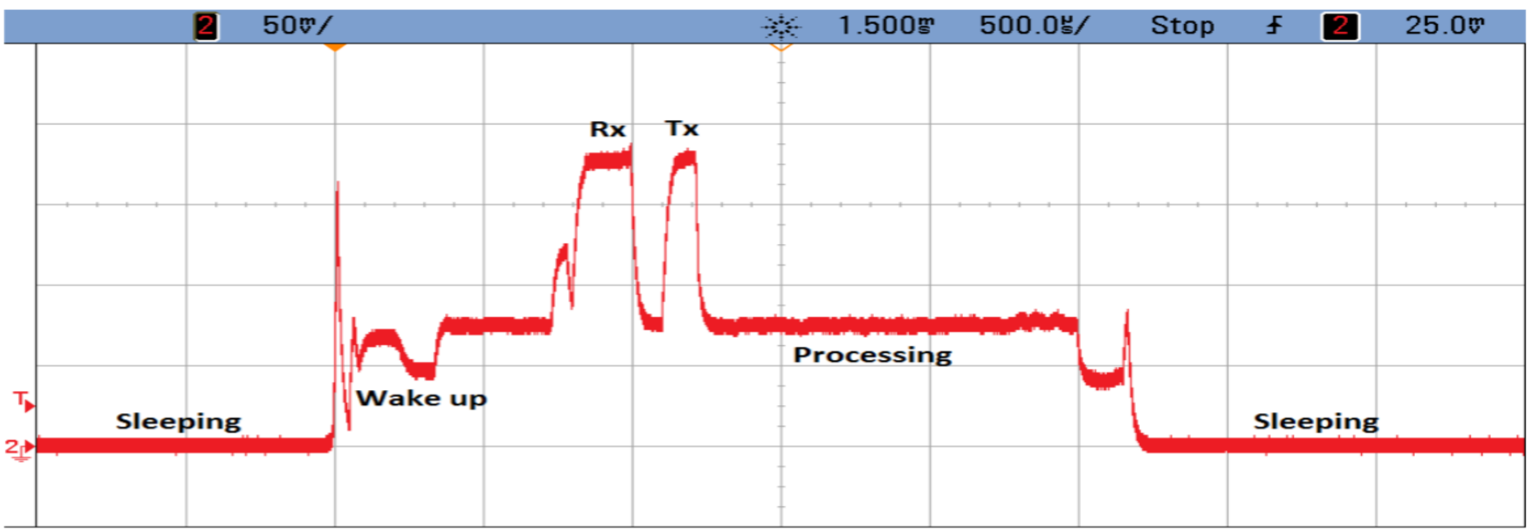
BLE: The lowest power consumption

- Everything from physical design to use models is designed to keep power consumption at a minimum
- To reduce power consumption, a BLE device is kept in **sleep mode** most of the time. When an event occurs, the device wakes up, sends a short message, and goes back to sleep
- Maximum/peak power consumption is less than 15 mA and the average power consumption is about 1 µA
- The active power consumption is reduced to a tenth of the energy consumption of classic Bluetooth
- In low duty cycle applications, a button cell battery could provide 5-10 years of reliable operation

Power consumption in BLE

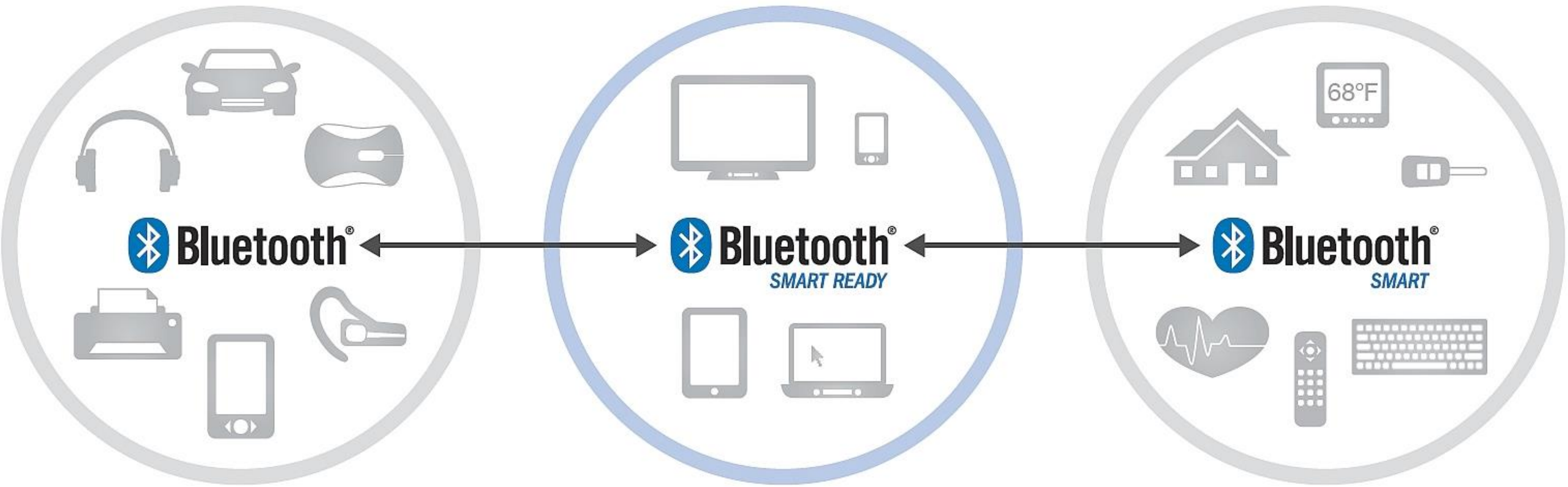


Current Consumption versus Time during a BLE Connection



Current Consumption versus Time during a single Connection Event

Three flavors of Bluetooth



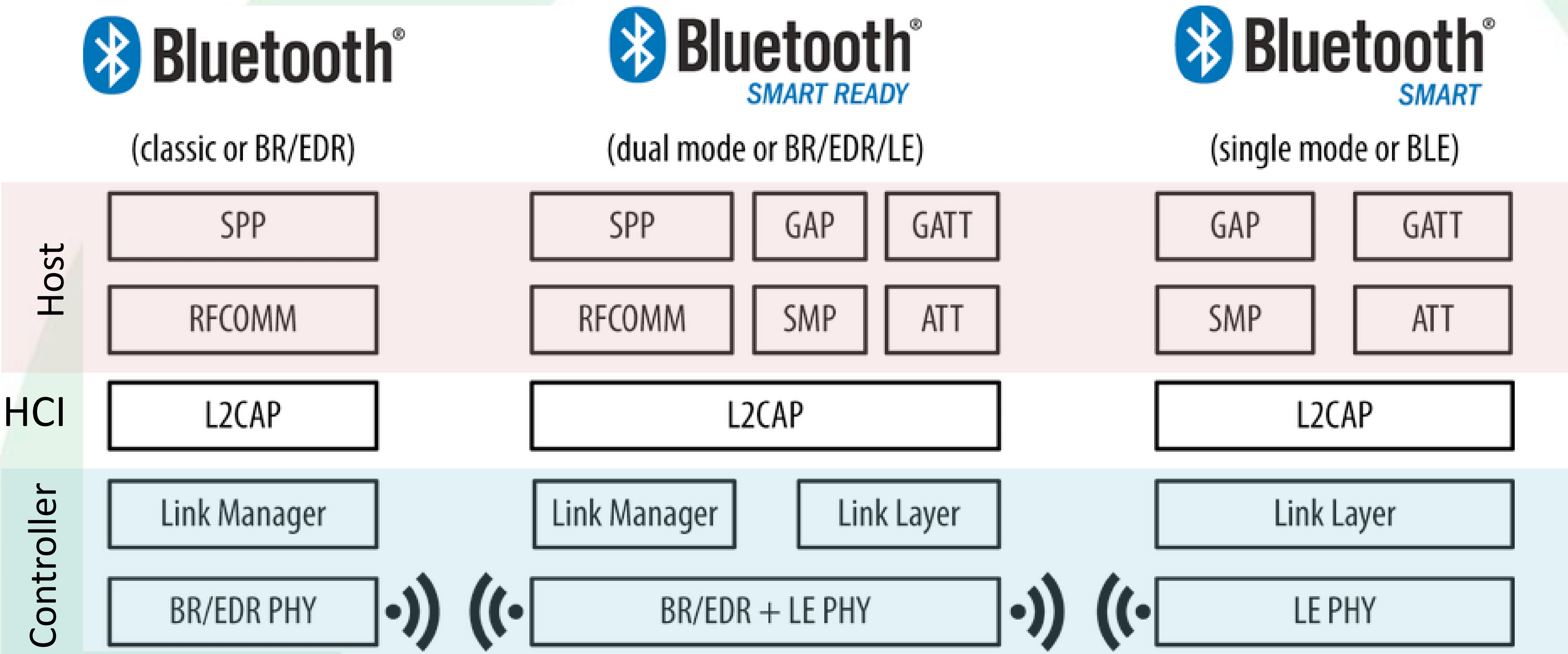
Bluetooth
Traditional wireless devices,
streaming rich content,
like video and audio

Bluetooth Smart Ready
Devices that connect with both
The center of your wireless world

Bluetooth Smart
Sensor devices,
sending small bits of data,
using very little energy

OntarioTech UNIVERSITY

Bluetooth Stack Architecture



1- Classic Bluetooth

- These are typically those devices that need a maintained and often high-throughput connection
- Classic Bluetooth is fantastic for products that require continuous streaming of voice and data.

2- Dual-mode / Smart Ready

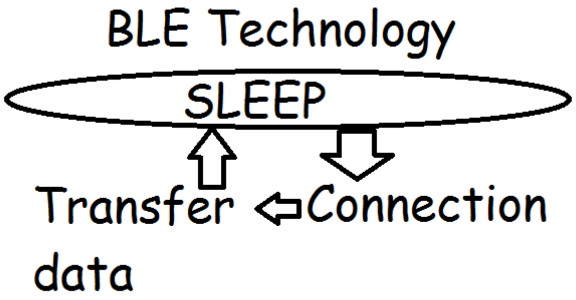
- A dual-mode device is an integrated circuit that includes both a standard Bluetooth radio and a BLE radio
- Each operates separately but not at the same time, though they can share an antenna
- Several vendors offer dual-mode chips, such as Broadcom, CSR, EM Microelectronics, Nordic Semiconductor, and Texas Instruments.
- Typical dual-mode devices are mobile phones, tablets, and computers

SimpleLink™ Bluetooth dual-mode CC2564

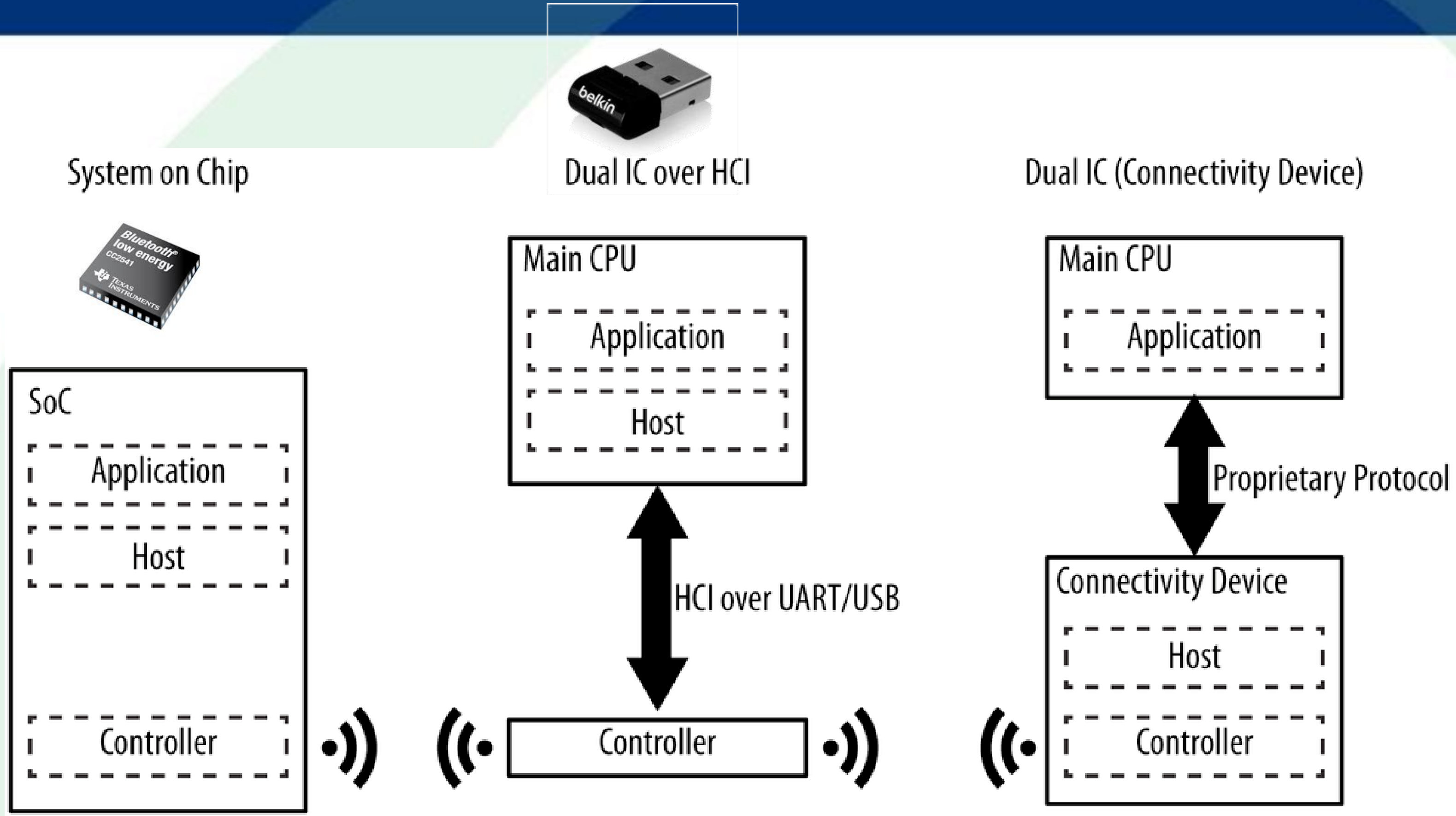
OntarioTech UNIVERSITY

Single mode/ Bluetooth Smart

- Supports **Bluetooth low energy** as the only form of communication
- These devices cannot communicate directly with Bluetooth “classic” devices, but on the other hand they are highly optimized for Bluetooth low energy
- Bluetooth low energy was designed and optimized for use-cases that have a relatively low duty cycle
- For example, a heart rate armband may stay connected for several hours during a long work-out, but it only needs to transmit a few bytes every second



BLE Hardware Configuration



IC: Integrated Circuit
HCI: Host Controller Interface

System on Chip

- **Description:**
  In an SoC configuration, the entire BLE stack (Controller and Host) along with the user application and a main CPU are integrated onto a single silicon chip.
- **Benefits:**
  Highly integrated, compact, low power, often the most cost-effective for mass-produced, dedicated BLE devices.
- **Use Cases:**
  Smart sensors, wearables, disposable medical devices, simple IoT endpoints.

OntarioTech UNIVERSITY

Dual IC over HCI (Host Controller Interface)

Connection:

The Main CPU communicates with the Controller via a standardized

interface called **HCI (Host Controller Interface)**.

Benefits: Allows for a powerful general-purpose CPU to handle complex applications while offloading the time-critical, real-time BLE radio operations to a dedicated Controller IC. Provides flexibility in CPU choice.

Use Cases:

Computers (where a Bluetooth dongle is a common example), more complex embedded systems where the main application processor is separate from the BLE radio.

OntarioTech UNIVERSITY

Dual IC (Connectivity Device)

Connection:
The Main CPU communicates with the Connectivity Device using a "Proprietary Protocol." This means the interface between the Main CPU and the Connectivity Device is not a standard HCI, but rather a custom communication protocol defined by the manufacturer of the Connectivity Device.

Benefits:

- Simpler for the Main CPU as it doesn't need to implement the BLE Host stack itself.
- The Connectivity Device handles most of the BLE complexity.
- Good for rapid prototyping or applications where the main CPU has limited resources but needs BLE functionality.

- **Use Cases:**
  - Embedded systems, modules used in consumer electronics where the manufacturer provides a simplified API.

BLE Data Throughput

- The modulation rate of the Bluetooth Low Energy radio is set by the specification at a constant 1Mbps.
- But there are many limitations that can affect the throughput
- For example, Nordic's radio hardware and BLE stack impose the following data throughput limitations:
  - The nRF51822 can transmit up to **six data packets** per connection interval (limited by the IC).
  - Each outgoing data packet can contain up to **20 bytes** of user data (set by the specification unless higher packet sizes are negotiated).
  - Connection interval is between 7.5 ms and 4s

Operating Range

- Bluetooth devices are divided into three power classes; the only difference between them is the transmission power levels used

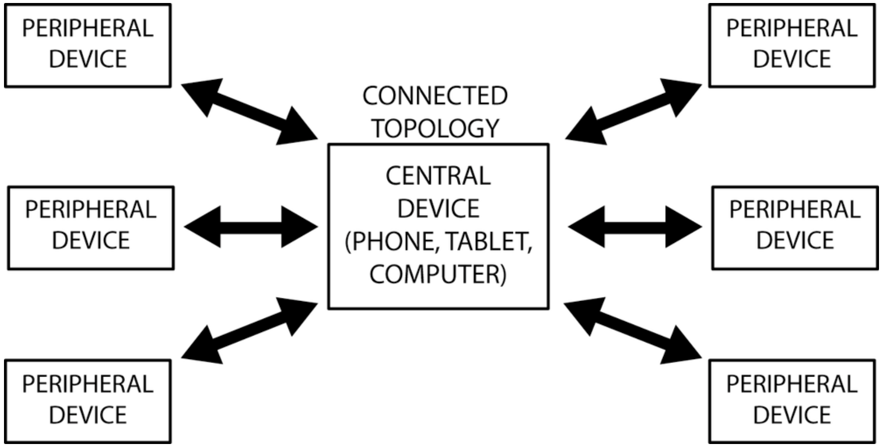
Power Class	Transmission Power Level (mW)	Advertised Range (m)
1	100	100
2	2.5	10
3	1	1

- Almost all Bluetooth-enabled cell phones, headsets, laptops, and other consumer-level Bluetooth devices are class 2 devices

OntarioTech UNIVERSITY

Broadcasting and Observing

- Using connectionless *broadcasting*, you can send data out to any scanning device or receiver in listening range.



Broacaster
Sends nonconnectable *advertising* packets periodically to anyone willing to receive them.

Observer
Repeatedly scans preset frequencies to receive any broadcasting nonconnectable advertising packets.

Connections

- Connections involve two separate roles:

Central (master)	***Peripheral (slave)***
Repeatedly scans the preset frequencies for connectable advertising packets and, when suitable, initiates a connection	Sends connectable advertising packets periodically and accepts incoming connections

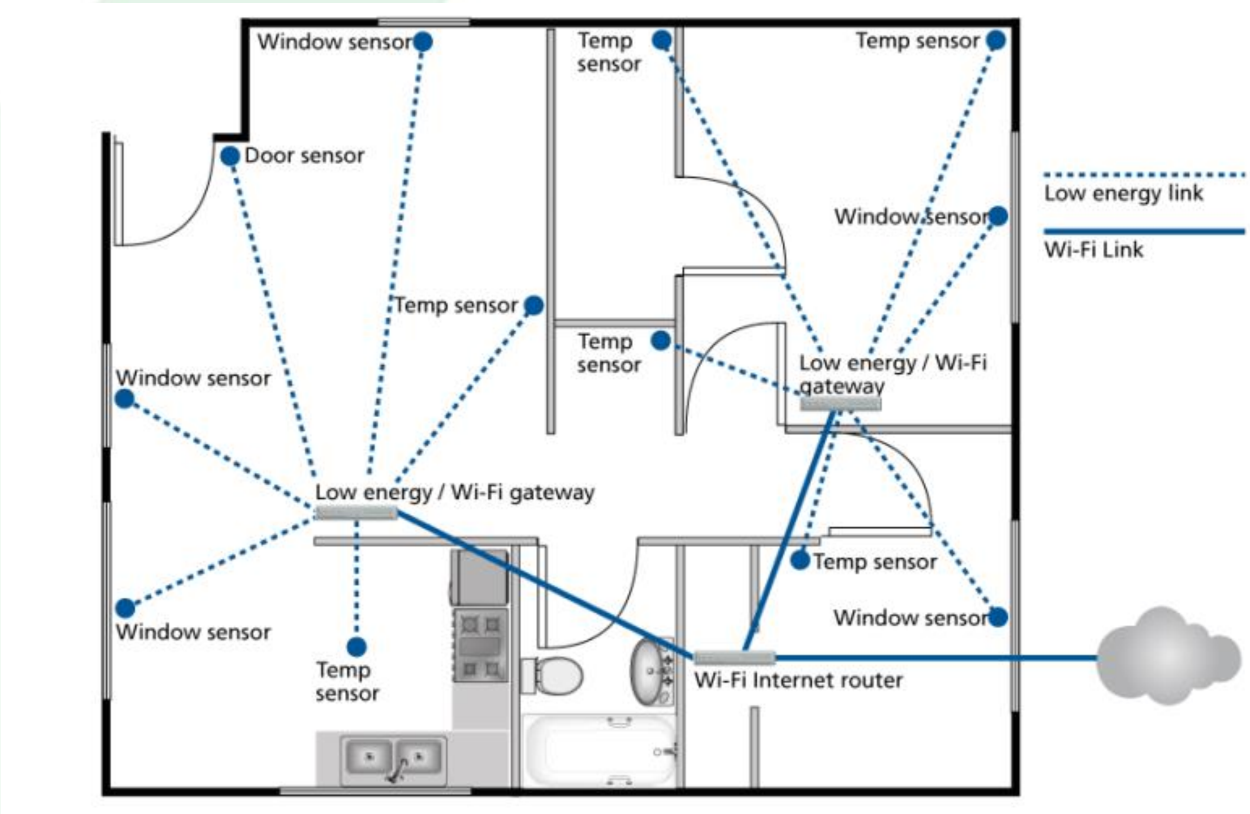
BLE 4.1 Specifications:

A device can act as a central & a peripheral at the same time

A central can be connected to multiple peripherals

A peripheral can be connected to multiple centrals

Extending the Coverage of BLE



Example of how to extend the Bluetooth low energy range via gateways.

Extending the Coverage of BLE (cont'd)

- The upstream link can be a cable (Ethernet) or a wireless link (Wi-Fi or Classic Bluetooth) and the downstream can be a Bluetooth low energy link.

- In the examples, Wi-Fi upstream links are used.

- Since the upstream connection in all the examples above is based on Internet protocols, the IP protocol contains all the necessary mechanisms to support traffic routing to cloud services and in some cases also between the local Bluetooth low energy devices (e.g. when IPv6 over Bluetooth low energy is used)

OntarioTech UNIVERSITY

Limitations – Things Bluetooth Can't Do

- Announce the presence of another device
- Detect when a remote device is inquiring for nearby devices
- Determine the Bluetooth address of an inquiring device
- Estimate the distance to a remote Bluetooth device
- Broadcast messages

OntarioTech UNIVERSITY

Bluetooth Programming with Python

Device discovery	`discover_devices()`
Name lookup	`lookup_name( address )`
SDP connect SDP search	`find_service( uuid )`
Establish an outgoing connection	`s = BluetoothSocket( protocol )` `s.connect( ( address, port ) )`

Establish an incoming connection	`s = BluetoothSocket( protocol )`
	`s.bind( ( address, port ) )`
	`s.listen( backlog )`
	`s.accept()`

Advertise an SDP service	`advertise_service( s, name, ... )`
Transfer data	`s.send( data )` `s.recv( numbytes )`
Disconnect	`s.close()`

PyBluez Quick Reference

References

- Getting Started with Bluetooth Low Energy, Book by Kevin Townsend, Carles Cufí, Akiba, Robert Davidson, May 2014, O'Reilly Media, Inc. – Chapters 1-4