WIRELESS NETWORKS

Gentian Jakllari – INP-ENSEEIHT
jakllari@enseeiht.fr

Main Topics

- Architectures for wireless and/or mobile networks
- Research & Engineering challenges
- State of the art and IEEE Standards
- Practice in an open-source simulator: ns-2

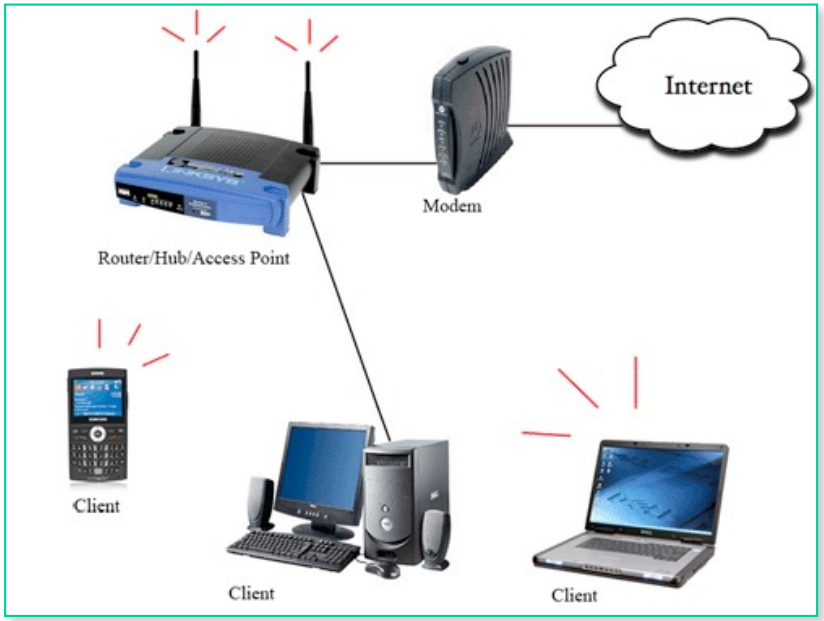
Goals

- Familiarize with the state of the art in wireless and mobile networks
- Get to know the main technologies, standards and bodies that dominate the wireless industry
- Get to know the fundamental engineering challenges
- Get to know the solutions and their limitations
- Practice key concept on an open-source simulator


Wi Fi
™

What is Wi-Fi?

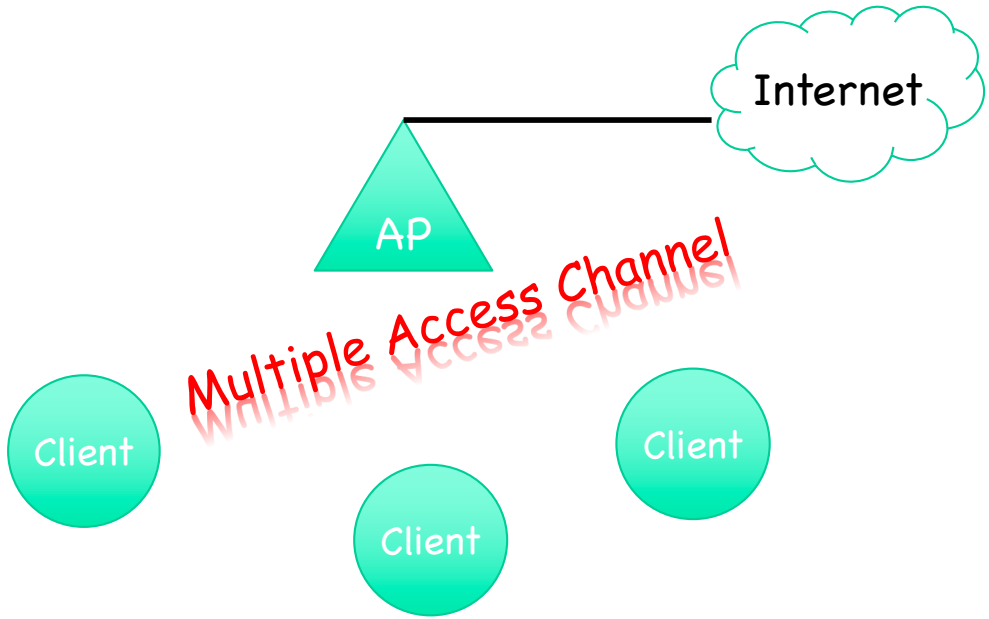
- Wi-Fi is the brand name for products using the IEEE 802.11 family of standards
- Supports two architectures:



WLAN Architecture

Ad-hoc (Peer-to-Peer) Architecture

WLAN Architecture

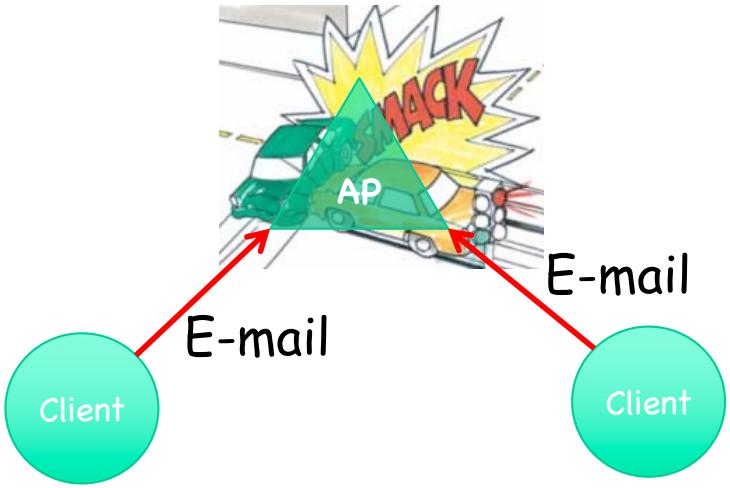


- A special station – Access Point – is static and connected to the internet through wires
- The other stations – the Clients – are free to move and connect to the Internet by giving their data to the Access Point
- All the clients communicate with the AP wirelessly on the same frequency/channel – *multiple access channel*

The Main Engineering Challenge in WLANs

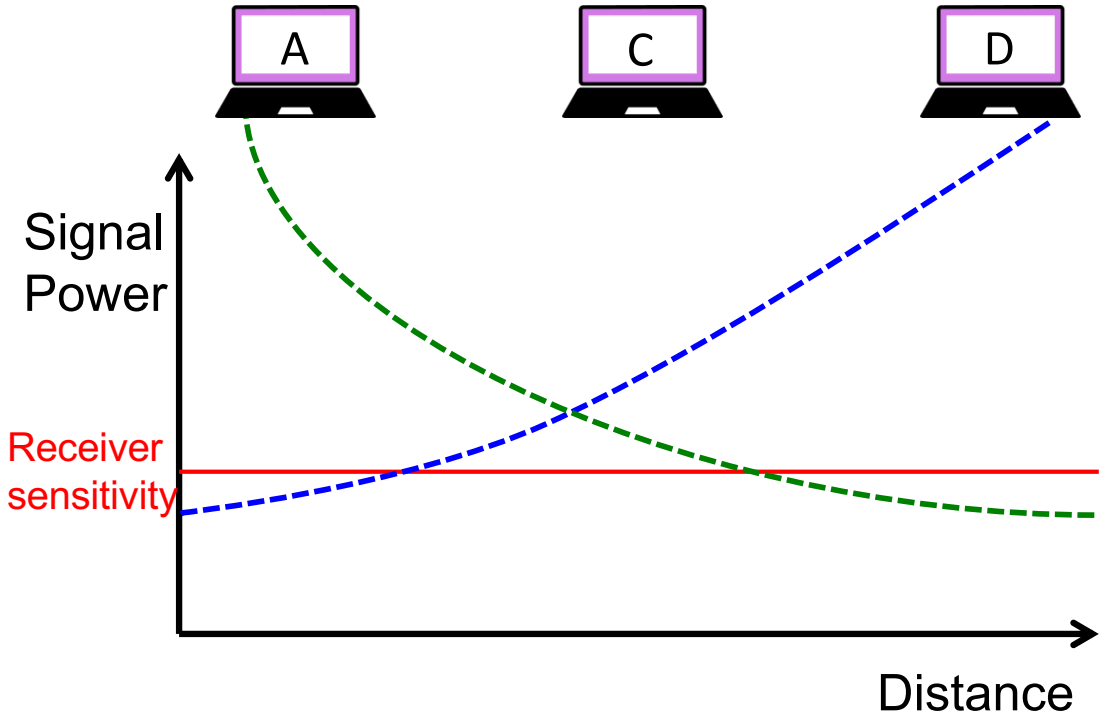
- *How can multiple stations communicate efficiently with a single Access Point while using the same channel?*
  - *Formally defined as the problem of Medium Access Control*

- Why is it challenging?
  1. The Access Point can only hear from one station at a time
     - If two stations transmit to the AP at the same time, the respective transmissions will collide and get destroyed



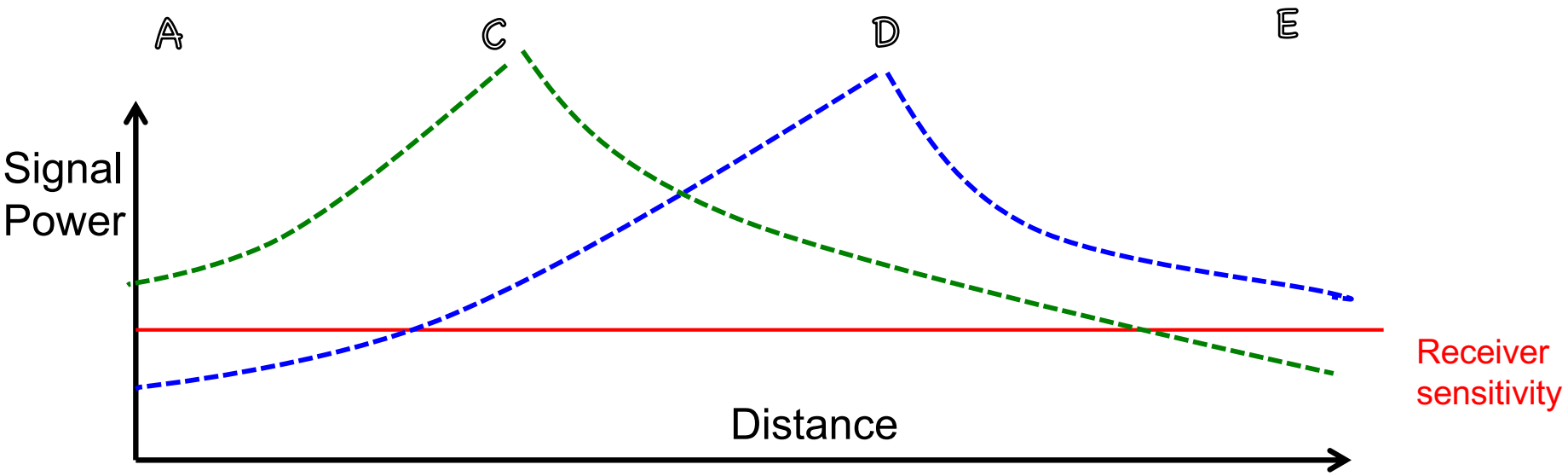
Sharing a wireless medium

Hidden terminal scenario



- Signal power levels not everywhere the same due to *pathloss*
- If A (D) does carrier sensing while D (A) is transmitting it will sense nothing and it will transmit – the wrong decision !

Exposed terminal scenario



- There is no collision at the receivers, A and E
- If C (D) does carrier sensing while D (C) is transmitting it will decide to defer --- the wrong decision (C&D are exposed terminals)

A Quick Conclusion

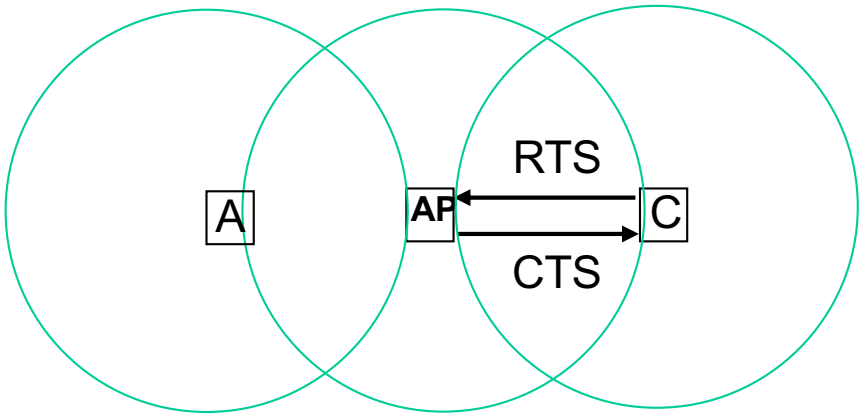
- CSMA in wireless networks:
  - Sometime it tells you to transmit when you should not (hidden terminal)
  - Sometime it tells you to not transmit when you should (exposed terminal)
- CD
  - Physically impossible

The Emergence of MACA, MACAW, & IEEE 802.11

- Wireless MAC proved to be non-trivial
- 1992 - research by Karn (MACA)
- 1994 - research by Bhargavan (MACAW)
- Led to IEEE 802.11 committee
  - The standard was ratified in 1999

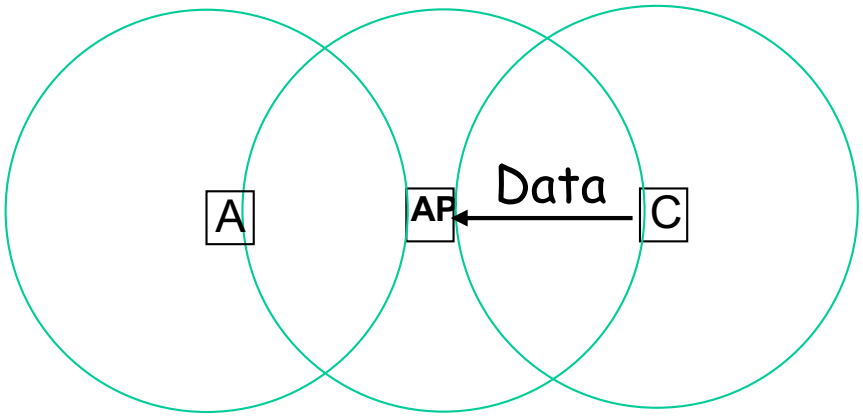
RTS/CTS: Addressing the Hidden Node

- MACA: Multiple Access with Collision Avoidance
- If node ***C*** has data to transmit, it first transmits a Request-to-Send (RTS) to the ***AP***. The RTS includes the duration of the pending data packet.
- The ***AP*** greenlights ***C*** by replying with a Clear-to-Send (CTS).
- ***A*** receives the CTS causing it to defer for the duration of the packet
-



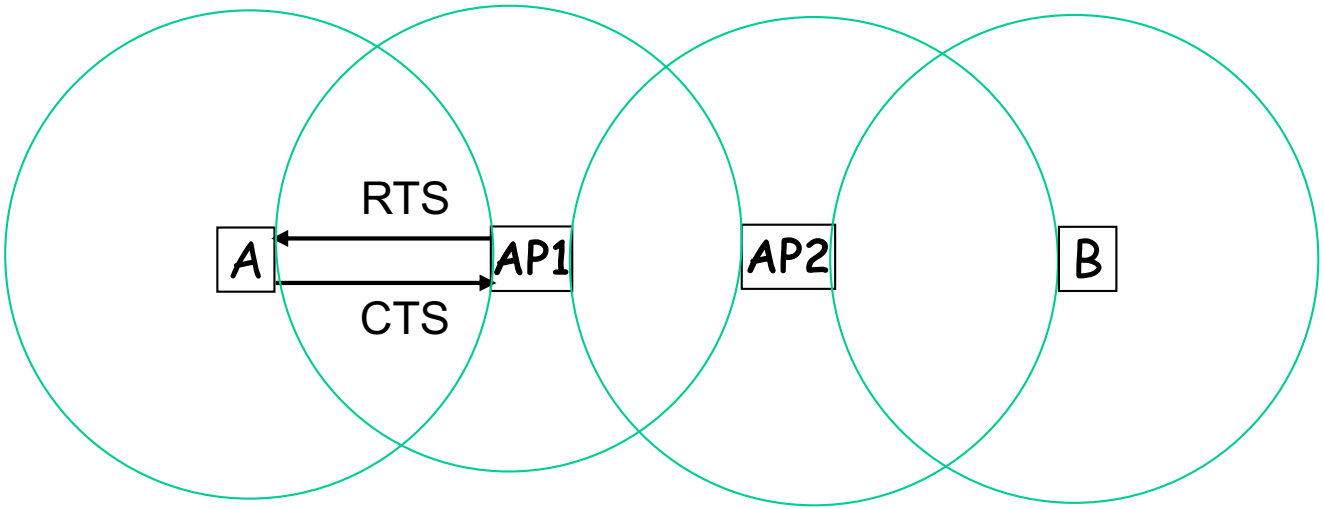
RTS/CTS: Addressing the Hidden Node

- MACA: Multiple Access with Collision Avoidance
- If node ***C*** has data to transmit, it first transmits a Request-to-Send (RTS) to the ***AP***. The RTS includes the duration of the pending data packet.
- The ***AP*** greenlights ***C*** by replying with a Clear-to-Send (CTS).
- ***A*** receives the CTS causing it to defer for the duration of the packet
- ***C*** transmits the data safely to the access point (***AP***)



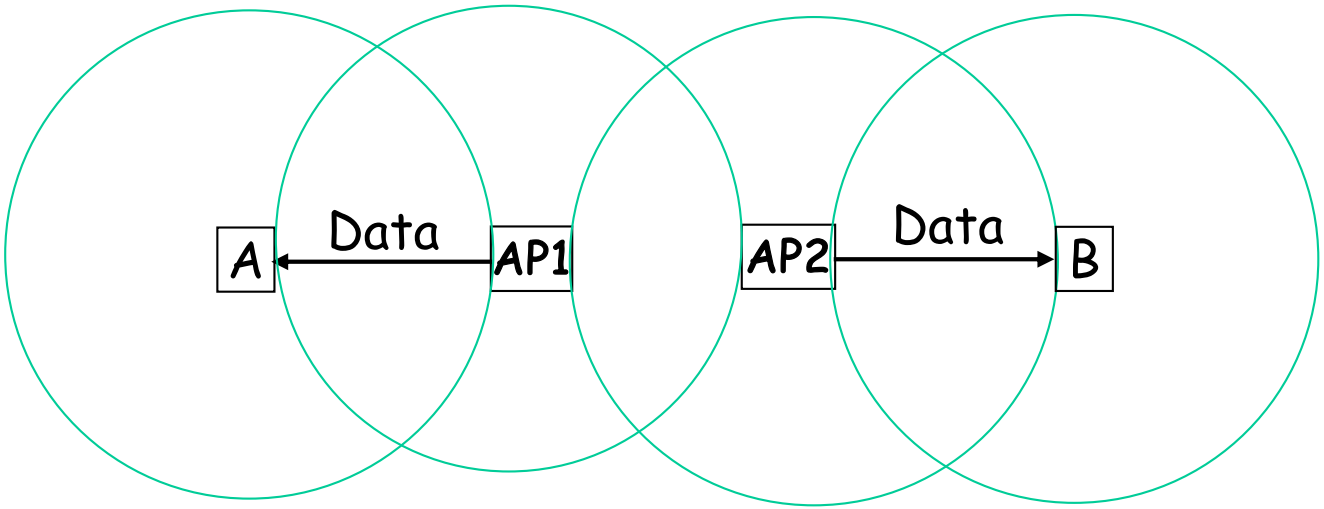
RTS/CTS: Addressing the Exposed Node

- If the ***AP1*** has data to transmit, it first sends a Request-to-Send (RTS) to ***A***.
- ***AP2*** receiving an RTS not addressed at him, defers from transmitting enough for the recipient to send a CTS.
- Node ***A*** transmits a Clear-to-Send (CTS). The CTS echoes the data packet duration ***B*** intends to transmit
-



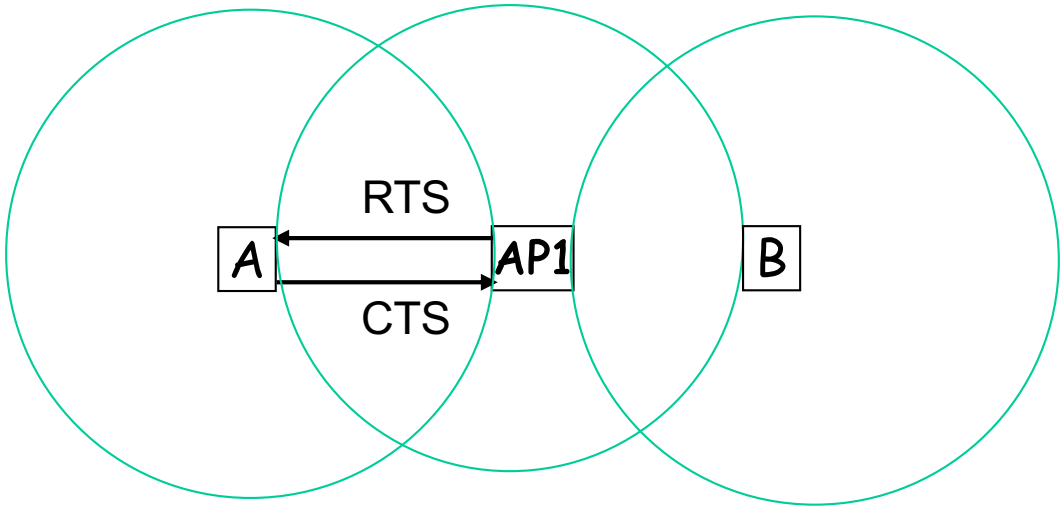
RTS/CTS: Addressing the Exposed Node (?)

- If the ***AP1*** has data to transmit, it first sends a Request-to-Send (RTS) to ***A***.
- ***AP2*** receiving an RTS not addressed at him, defers from transmitting enough for the recipient to send a CTS.
- Node ***A*** transmits a Clear-to-Send (CTS). The CTS echoes the data packet duration ***B*** intends to transmit
- ***AP2*** receives the RTS but did not receive the CTS- it cannot cause a collision - and therefore it's free to transmit to node ***B***



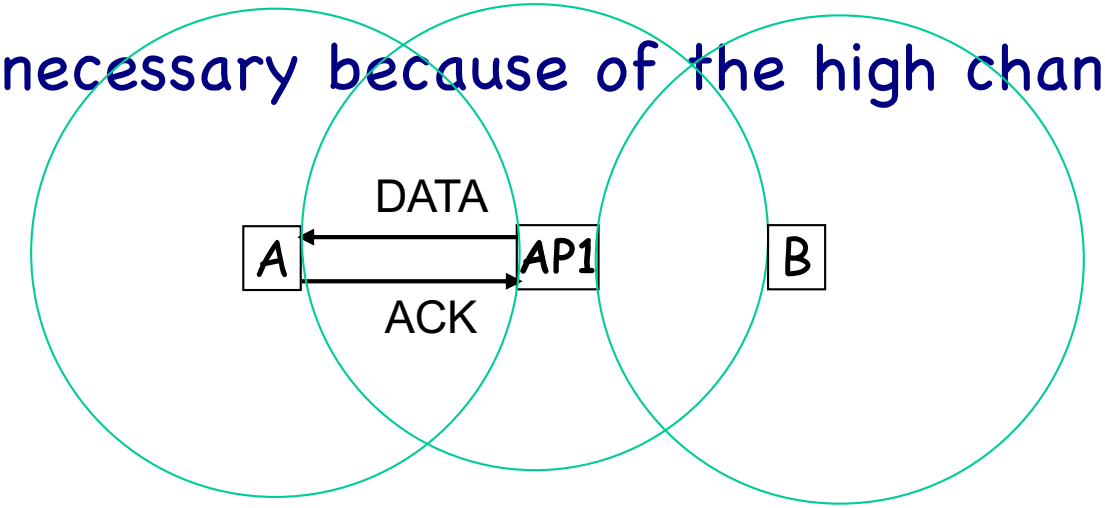
IEEE 802.11 MAC: CSMA/CA

- Combination of carrier sensing with collision avoidance
  - CSMA before transmitting an RTS
- ARQ: Automatic Request Acknowledgment
  - The recipient of data packet sends an ACK to the sender
  - It's necessary because of the high channel errors

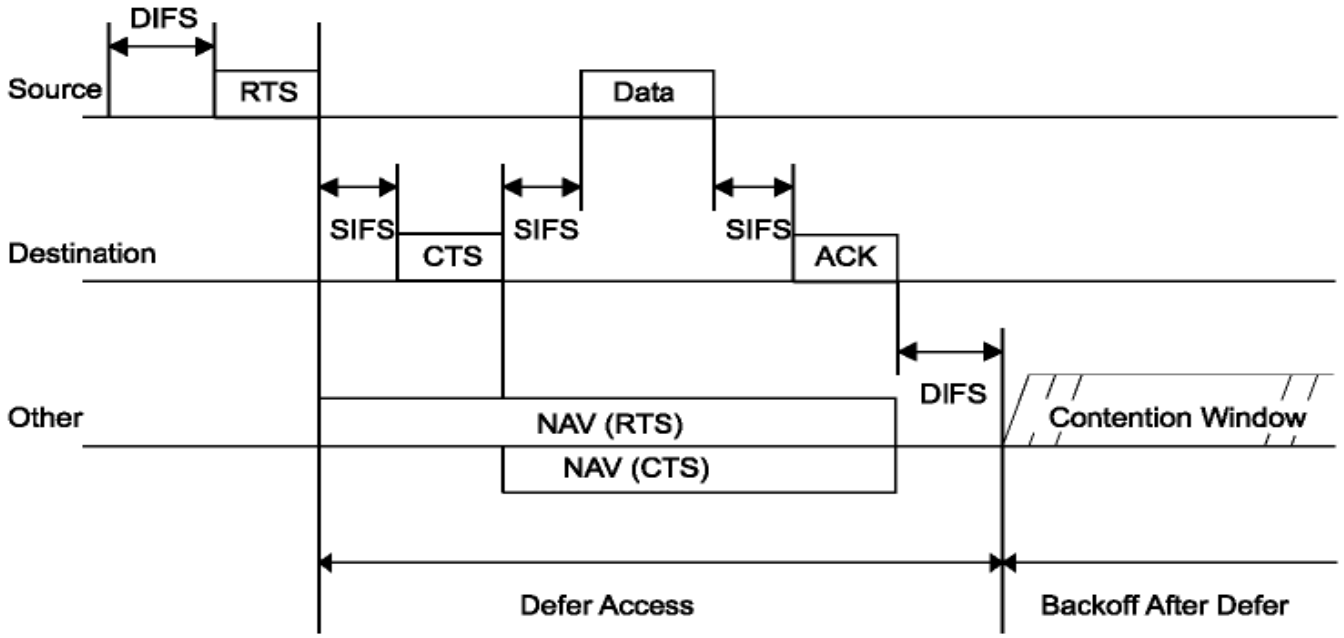


IEEE 802.11 MAC: CSMA/CA

- Combination of carrier sensing with collision avoidance
  - CSMA before transmitting an RTS
- ARQ: Automatic Request Acknowledgment
  - The recipient of data packet sends an ACK to the sender
  - It's necessary because of the high channel errors



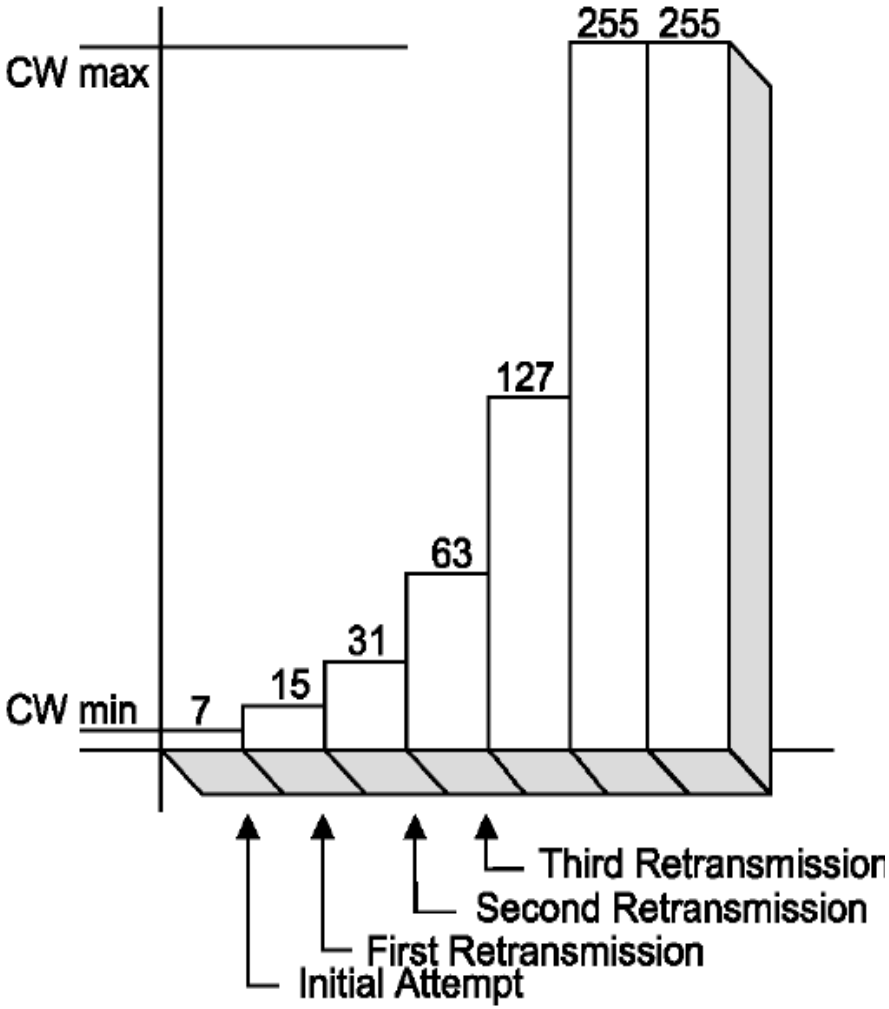
Frame Spacing in IEEE 802.11



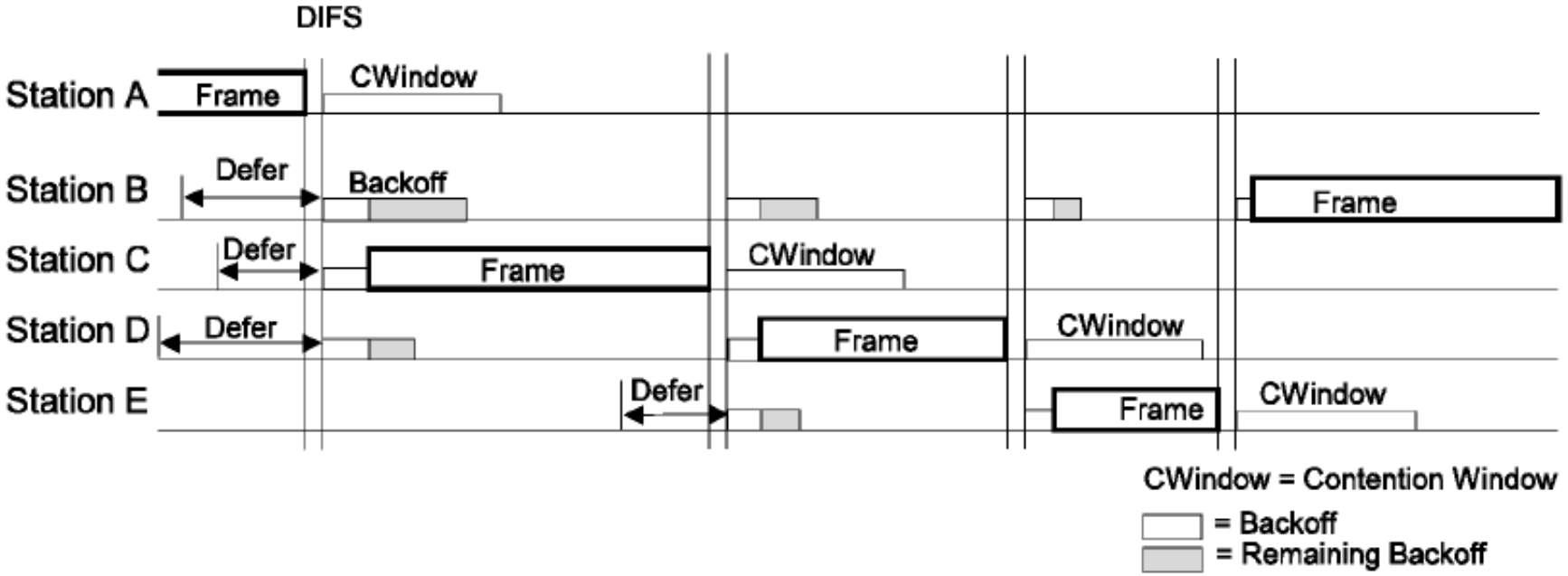
- **SIFS** (Short Interframe Space)= RxRFDelay + RxPLCPDelay + MACProcessingDelay + RxTxTurnaroundTime
- **SlotTime** = aCCATime + aRxTxTurnaroundTime + aAirPropagationTime + aMACProcessingDelay
- **DIFS**(DCF Interframe Space) = SIFS + 2 x SlotTime

Backoff in IEEE 802.11

- Backoff Time = Random() x SlotTime
- Random() = Pseudo-random integer drawn from [0,CW], where CWmin ≤ CW ≤ Cwmax
- The CW is initialized at CWMin and is doubled every time there is no CTS for an RTS or there is no ACK for a DATA -> Why?
- Once it reaches CWmax the CW does not increase anymore
- The CW is reset to CWmin after receiving a CTS or ACK or a packet is dropped.



Example of a Backoff Race



1. The node drawing the smallest Contention Window wins the race and gets to transmit first

QOS IN IEEE 802.11 (WI-FI)

What is QoS ?

- Quality of service is the ability to:

1. Provide ***different*** priorities to different applications, users, or data flows

   or

2. To ***guarantee*** a certain level of performance to a data flow

QoS in IEEE 802.11

- Can IEEE 802.11:

1. Provide different priorities to different applications, users, or data flows ?

2. Guarantee a certain level of performance to a data flow ?

QoS in IEEE 802.11

- Can IEEE 802.11:

1. Provide different priorities to different applications, users, or data flows ?

- Answer: YES
- Different nodes could use different backoff counters

2. Guarantee a certain level of performance to a data flow ?

- Answer: NO
- It is possible, though highly unlikely, that a node never wins the backoff race -> guarantees are off the table

IEEE 802.11e: QoS Amendment

- An approved amendment that defines QoS enhancements through modifications to the MAC layer

- DCF -> Enhanced distributed channel access (EDCA)
  - Use shorter CW counter for higher priority traffic

  - Transmit opportunity (TXOP): a node winning the backoff race is free to transmit frames continuously for a up to a TXOP period
    - No need for a backoff between packet transmissions

IEEE 802.11e Parameters

AC	CWmin	CWmax	AIFSN	Max TXOP
Background (AC_BK)	31	1023	7	0
Best Effort (AC_BE)	31	1023	3	0
Video (AC_VI)	15	31	2	3.008ms
Voice (AC_VO)	7	15	2	1.504ms
Legacy DCF	15	1023	2	0

IEEE 802.11 RATE CONTROL

Wi-Fi PHY

802.11 protocol	Release[1]	Freq. (GHz)	Bandwidth (MHz)	Data rate per stream (Mbit/s)[2]	Allowable MIMO streams	Modulation
—	Jun 1997	2.4	20	1, 2	1	DSSS, FHSS
a	Sep 1999	5	20	6, 9, 12, 18, 24, 36, 48, 54	1	OFDM
		3.7[A]				
b	Sep 1999	2.4	20	1, 2, 5.5, 11	1	DSSS
g	Jun 2003	2.4	20	6, 9, 12, 18, 24, 36, 48, 54	1	OFDM, DSSS
n	Oct 2009	2.4/5	20	7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 65, 72.2[B]	4	OFDM
			40	15, 30, 45, 60, 90, 120, 135, 150[B]		
ac	Dec 2012	5	20	up to 87.6[4]	8	
			40	up to 200[4]		
			80	up to 433.3[4]		
			160	up to 866.7[4]		
ad	~Feb 2014	2.4/5/60		up to 6912 (6.75Gb/s)[5]		

Questions on the PHY

1. Why so many rates?

2. Which rate do the stations use?

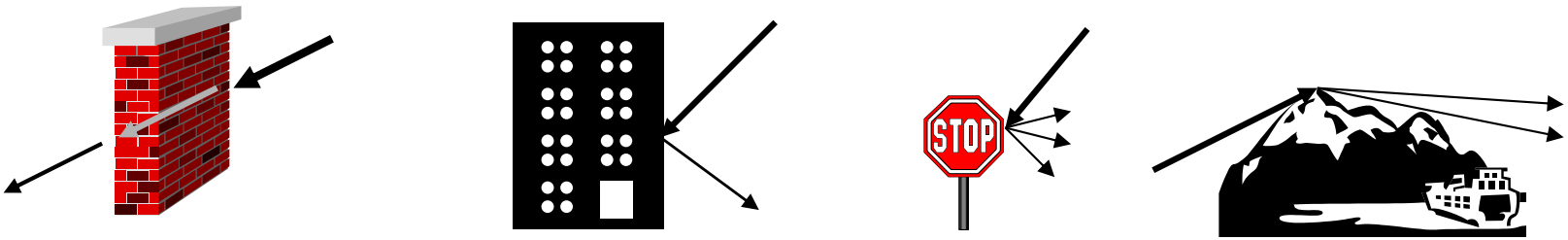
Radio Channel

- The radio channel is different
  - Extremely harsh environment compared to "wired" or guided media
  - Channel is time variant because of
    - Movement of people changes reflection
    - Switching off and on of interference
    - Movement of mobile terminals changes the distance
    - Sensitivity to a variety of other factors like "Fading" and "Multipath"

Radio Channel

- Path loss
- Interference
- Shadowing
- Multipath – receiving multiple reflections of the original signal that will interfere with each other
- Interference from other operators on the same frequency (microwaves use the same frequency as wi-fi)

- All of these can change fast when the nodes are mobile!



shadowing reflection scattering diffraction

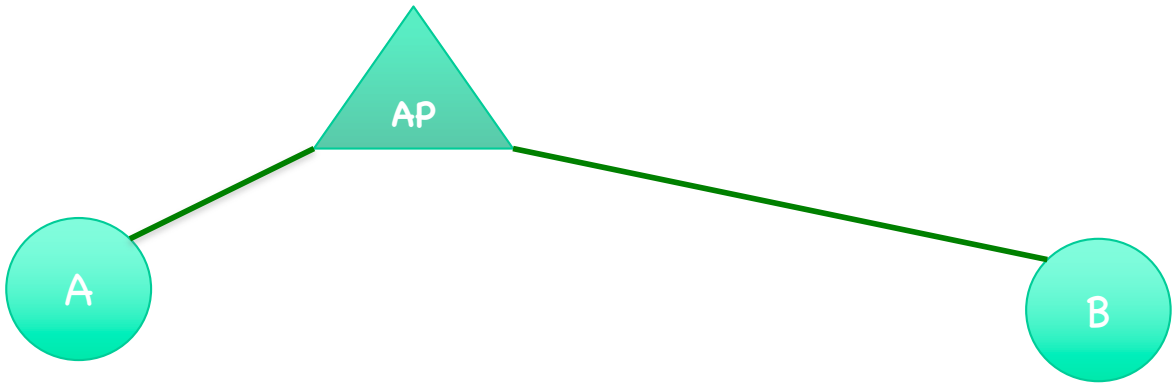
Why so many rates in Wi-Fi

- The quality of the signal between any pair of Wi-Fi stations can vary greatly
  - Distance
  - Environment (shadowing, multipath)

- The better the signal the higher a rate can be used

Rate Control

- Problem: The access point wants to transmit to A and B.
  - What rate should it use with A?
  - What rate should it use with B?
- Why is it challenging
  - The AP does not know the conditions of the links to A and B
  - A and B may be moving, which would change the channel conditions and therefore the right bit-rate to use



Auto Rate Fallback (ARF)

- When the ARF algorithm starts for a new destination, it selects the initial bit-rate to be the highest possible bit-rate.

- ARF adjusts the bit-rate for the destination based on the following criteria:
  - Move to the next lowest bit-rate if a packet was dropped (i.e., never ACKed)
  - Move to the next highest bit-rate if 10 successive transmissions have occurred without any retransmissions.
  - Otherwise, continue at the current bit-rate.

- Weakness: only reacts to packet drops not retries!

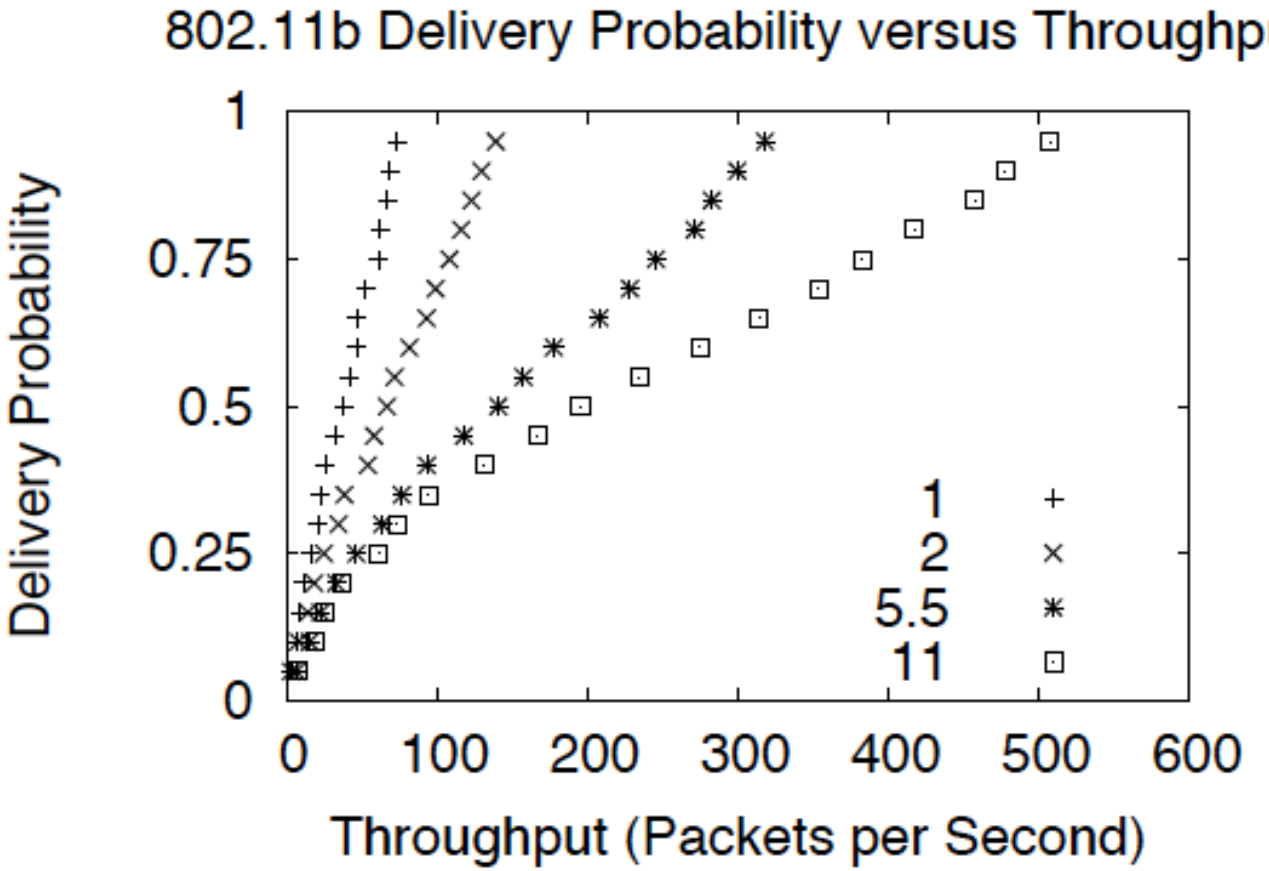
Onoe

- The first open source bit-rate selection algorithm designed to work with 802.11b, 802.11g, and 802.11a devices
- Tries to find the highest bit-rate that has less than 50% loss rate
- For each individual destination, the Onoe algorithm keeps track of the current bit-rate for the link and the number of credits that bit-rate has accumulated
  - It only keeps track of these credits for the current bit-rate and increments the credit if it is performing with very little packet loss
- Once a bit-rate has accumulated a threshold value of credits, Onoe will increase the bit-rate
- If a few error conditions occur, the credits will be reset and the bit-rate will decrease

Onoe

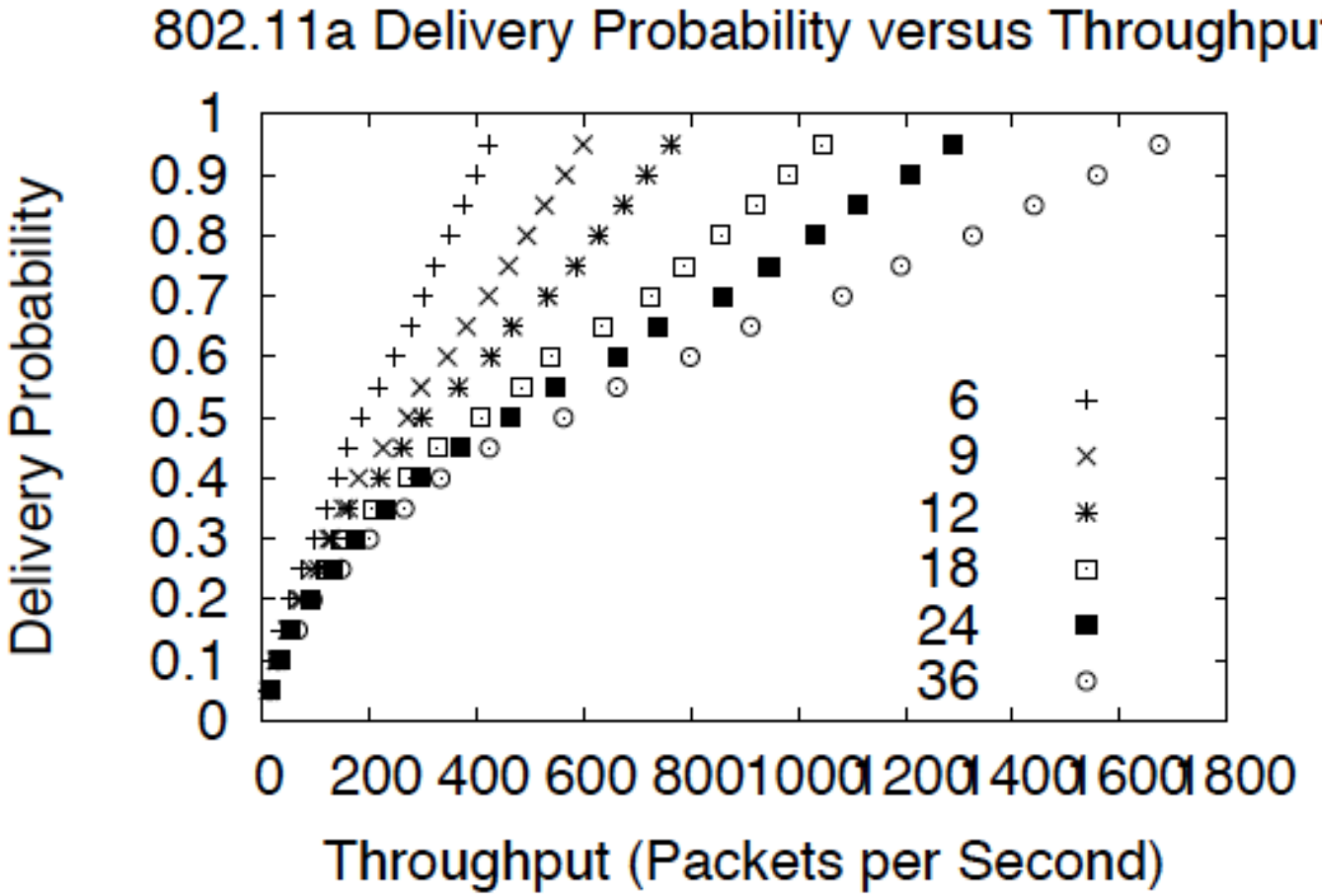
- Initially, set the rate to a destination to the highest. It also sets the number of credits for that bit-rate to 0.
- Periodically (1/sec by defaul) perform the following for every destination
  - Move to the next lower rate if:
    - No packets have succeeded
    - If 10 or more packets have been sent and the average number of retries per packet was greater than one
  - If the current bit-rate has 10 or more credits, increase the bit-rate
    - If more than 10% of the packets needed a retry, decrement the number of credits (minimum 0)
    - If less than 10% of the packets needed a retry, increment the number of credits
  - Otherwise continue at the current rate.

Onoe Performance



- Customized for 802.11b where the throughput of the next lowest bit-rate is usually half the current bit-rate

Onoe Performance



- It won't work well for 802.11a: perfect 24 Mbps is better than 70% 36 Mbps

Onoe Performance

- Onoe is conservative: once it decides a bit-rate will not work, it will not attempt to step up again until at least 10 seconds have gone by
- It can take time to stabilize: It will only step down one bit-rate during each period
  - It can take a few seconds before the Onoe algorithm can send packets if it starts at a bit-rate that is too high for a given link.

Receiver Based Auto-Rate (RBAR)

- Chooses the bit-rate based on SNR measurements at the receiver
- When a receiver gets an RTS packet, it calculates the highest bit-rate that would achieve a BER less than 10^{-5} based on the SNR of the RTS packet
- The receiver piggybacks on the CTS packet the rate the sender should use to send the data packet
- Weakness: It may not be possible to compute the best rate based on the SNR

Opportunistic Auto-Rate (OAR)

- The intuition behind OAR is that channel coherence times typically exceed multiple packet transmission times
  - By taking advantage of high link qualities when they appear, channel throughput can be increased
- OAR uses the RTS/CTS exchange for rate control purposes (like RBAR)
- Grant each sender the same amount of time in the CTS as the transmission time of a packet at the base rate
  - The sender can send multiple packets at a high bit-rate in the same time that one transmission would take at a lower bit-rate.

SAMPLERATE

Design Principles

- A bit-rate selection algorithm cannot conclude that higher bit-rates will perform poorly just because lower bit-rates perform poorly
- The bit-rate that achieves the most throughput may suffer from a significant amount of loss. Algorithms that only use bit-rates with high delivery probability may not find the bit-rate that achieves the highest throughput
- Link conditions may change. Failing to react to changes in link conditions could result in needlessly low throughput
- A bit-rate selection algorithm that constantly measured the throughput of every bit-rate would likely achieve low throughput.

General Approach

- SampleRate sends data at the bit-rate that has the smallest predicted average packet transmission time, including time required to recover from losses
  - Predicts the estimated packet transmission time by averaging the transmission times of previous packet transmissions at a particular bit-rate
- Periodically send packets at a bit-rate other than the current bit-rate to gather information about other bit-rates
- Reduce the number of bit-rates to sample by eliminating bit-rates that could never send at higher rates than the current bit-rate that is being used.

SampleRate

- Start at the highest possible rate
- Stop using a bit-rate after four successive drops
- Every tenth data packet pick a random bit-rate from the set of bit-rates that *may* do better than the current one and sends the packet using that bit-rate instead of the current one
- A bit-rate is not eligible to be sampled if
  - Four recent successive packets at that bit-rate have been unacknowledged or
  - Its lossless transmission time (without any retries) is greater than the average transmission time of the current bit-rate
  - Calculate the average transmission time over packets that were sent within the last 10 seconds

SampleRate: Example in Practice

Destination	Bit-rate	Tries	Packets Ack'ed	Succ. Fails	Total TX Time	Avg TX Time	Lossless TX Time
00:05:4e:46:97:28	11	16	0	4	250404	∞	1873
00:05:4e:46:97:28	5.5	100	100	0	297600	2976	2976
00:05:4e:46:97:28	2	0	0	0	0	-	6834
00:05:4e:46:97:28	1	0	0	0	0	-	12995
00:0e:84:97:07:50	11	28	14	0	52654	3761	1873
00:0e:84:97:07:50	5.5	50	46	0	148814	3235	2976
00:0e:84:97:07:50	2	0	0	0	0	-	6834
00:0e:84:97:07:50	1	0	0	0	0	-	12995

- Destination **00:05:4e:46:97:28** has the properties that 11 megabits delivers no packets and all packets sent at 5.5 megabits are acknowledged successfully without retries
- The first few packets sent on this link at 11 megabits failed, and once the Successive-Failures column reached 4 it stopped sending packets at 11 megabits. It then proceeded to send 100 packets at 5.5 megabits
- Packets were never sent at 1 or 2 megabits because their lossless transmission time is higher than the average transmission time for 5.5 megabits

SampleRate: Example in Practice

Destination	Bit-rate	Tries	Packets Ack'ed	Succ. Fails	Total TX Time	Avg TX Time	Lossless TX Time
00:05:4e:46:97:28	11	16	0	4	250404	∞	1873
00:05:4e:46:97:28	5.5	100	100	0	297600	2976	2976
00:05:4e:46:97:28	2	0	0	0	0	-	6834
00:05:4e:46:97:28	1	0	0	0	0	-	12995
00:0e:84:97:07:50	11	28	14	0	52654	3761	1873
00:0e:84:97:07:50	5.5	50	46	0	148814	3235	2976
00:0e:84:97:07:50	2	0	0	0	0	-	6834
00:0e:84:97:07:50	1	0	0	0	0	-	12995

- Destination **00:0e:84:97:07:50** has the properties that packets sent at 11 megabits require a retry before being acknowledged, and packets sent at 5.5 megabits require no retries 90% of the time and one retry otherwise.
- SampleRate starts at 11 megabits and then sends the 10th packet at 5.5 megabits
- After the first 5.5 megabit packet required no retries, SampleRate determined that 5.5 megabits is the better rate
- It still sent at 11 megabits once every 10 packets to see if performance has improved at that bit-rate.

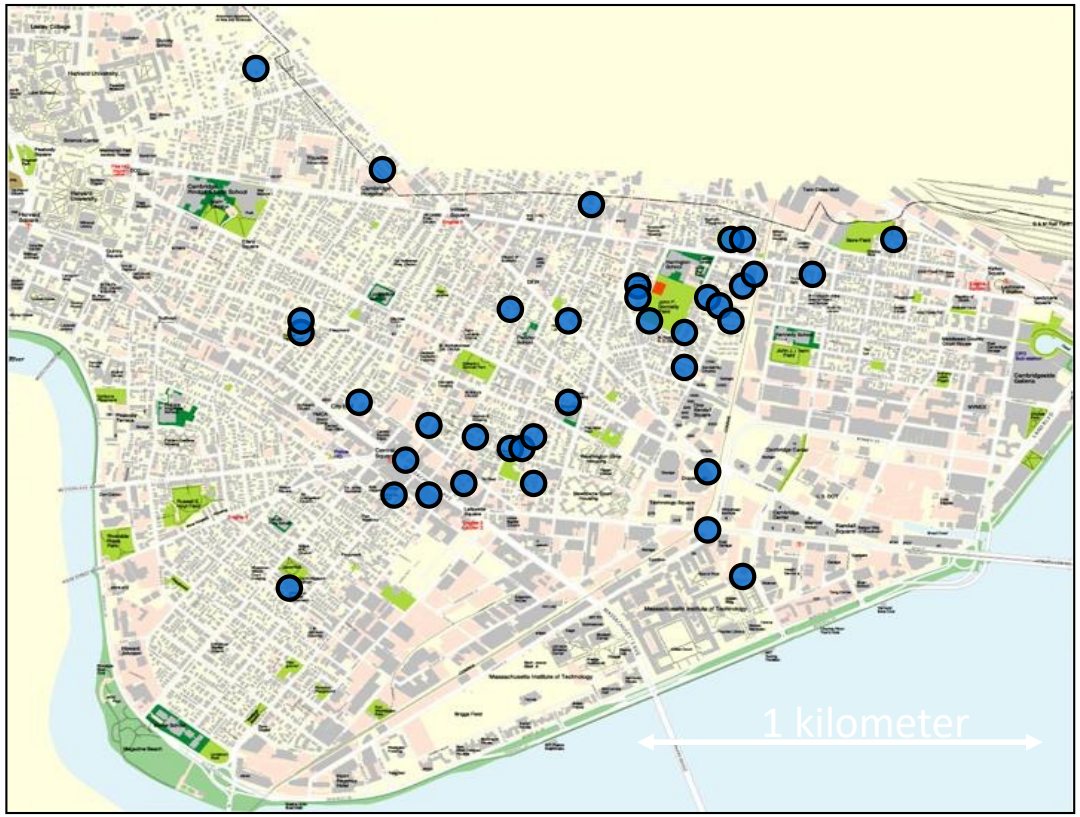
Computing the Transmission Time

Attempt	Average Back-off
1	155
2	315
3	635
4	1275
5	2555
6	5115
7	5115
8	5115

$$tx\_time(b, r, n) = difs + backoff(r) + (r + 1) * (sifs + ack + header + (n * 8/b)$$

b: bit-rate, r: number of retries, n: packet size

Evaluation – RoofNet Project

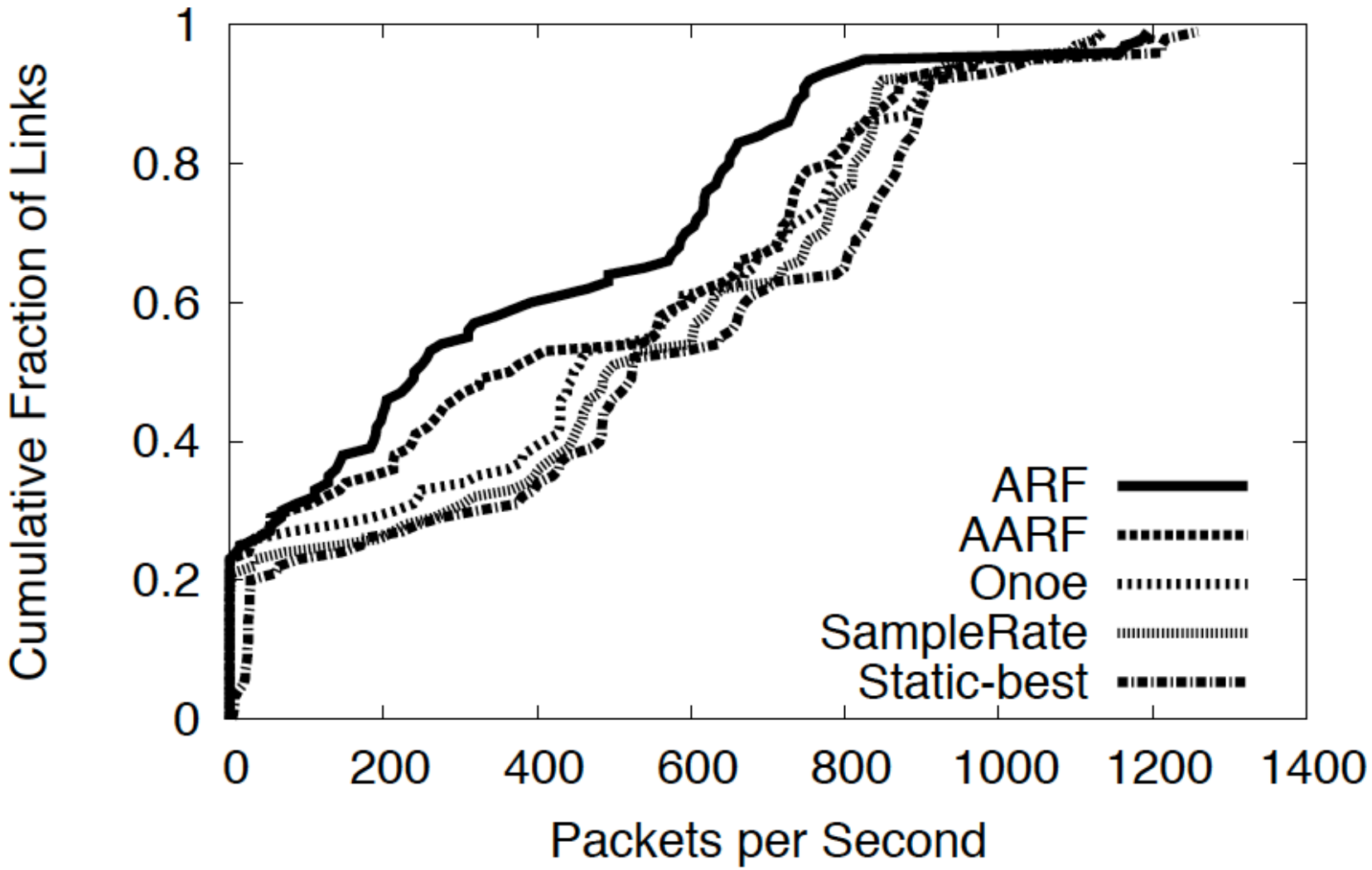


- 38 nodes distributed over 6 km^2
- Each node consists of a PC with an 802.11b card connected to an omni antenna mounted on the roof
  - Intersil Prism 2.5 chip-set
- The cards use 802.11b channel 3 with transmission power +23dBm(200mV)
- The antenna provides 8dBi of gain with 20-degree -3dB vertical beam-width

Evaluation – RoofNet Project

- Each bit-rate selection algorithm runs for 30 sec on every link on the testbed
- During the 30 sec the transmitters sends 1500-byte unicast packets as fast as it could
- To have a basis of comparison, the unicast throughput for each bit-rate was measured over every link
- The maximum throughput achieved for all the bit-rates on a link is referred to as best-static throughput

Evaluation – RoofNet Project



Rate Control

- All the rates control algorithms are for unicast traffic
  - They all require feedback from the receiver
- What about the broadcast traffic?

TP: ns-2 Exercise: Contention Window

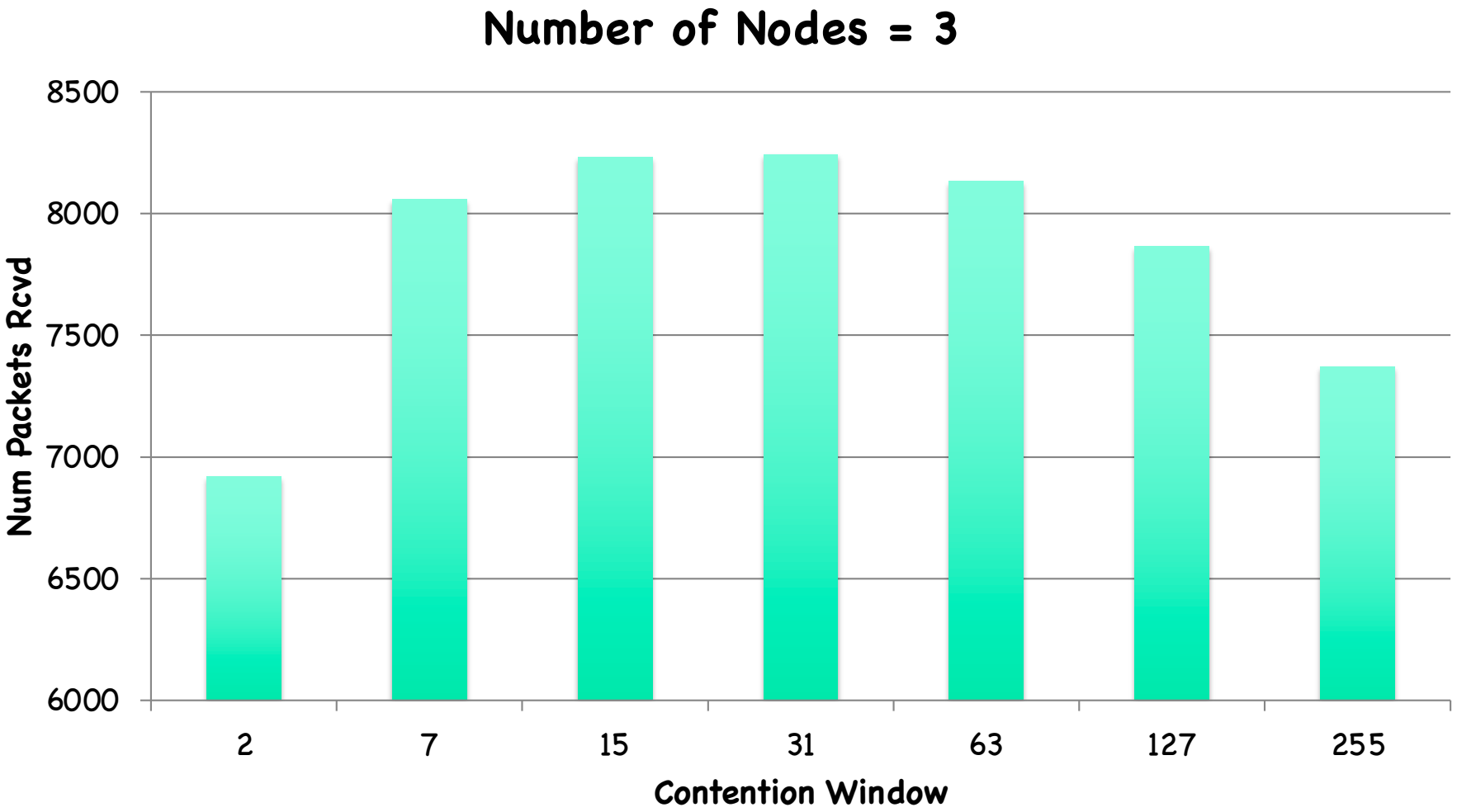
- Goal: Illustrate the relationship between the contention window, the number of interfering nodes and the network performance
- Use the provided script, *cwsim.tcl*, to run simulations for different-size clique networks
- Usage: ns cwsim.tcl -numNodes <number of nodes>
- For contention window cw in {2, 7, 15, 31, 63, 127, 255}
  - Edit cwsim.tcl and do:
  - Mac/802_11 set CWMin_ cw
  - Mac/802_11 set CWMax_ cw
  - For number of nodes n in {3, 10, 50}
    - ns cwsim.tcl -numNodes n
    - Measure the number of packets successfully transmitted (grep "^r" cwsim.tr | grep "AGT"|wc - l") and save it

ns-2 Exercise

- For every topology size create a graph where
  - x is the congestion window
  - y is the number of packets successfully transmitted during the respective ns experiment

Number of Nodes = 3	
CWMin_= CMMax_	Packets Received
2	6920
7	8058
15	8231
31	8243
63	8132
127	7865
255	7372

ns-2 Example Graph



ns-2 Experiment Analysis

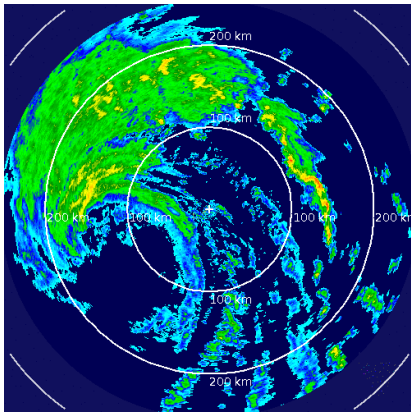
- Answer the following questions:

1. What trends do you observe? Do you observe any optimal CW size for each network population?

2. Can you predict what would happen if you tried to run with larger topologies?

WiFi-based Indoor Localization

LOCALIZATION BASED ON FINGERPRINTING



RADAR:

An In-Building RF-based User Location and Tracking System

Main Goal

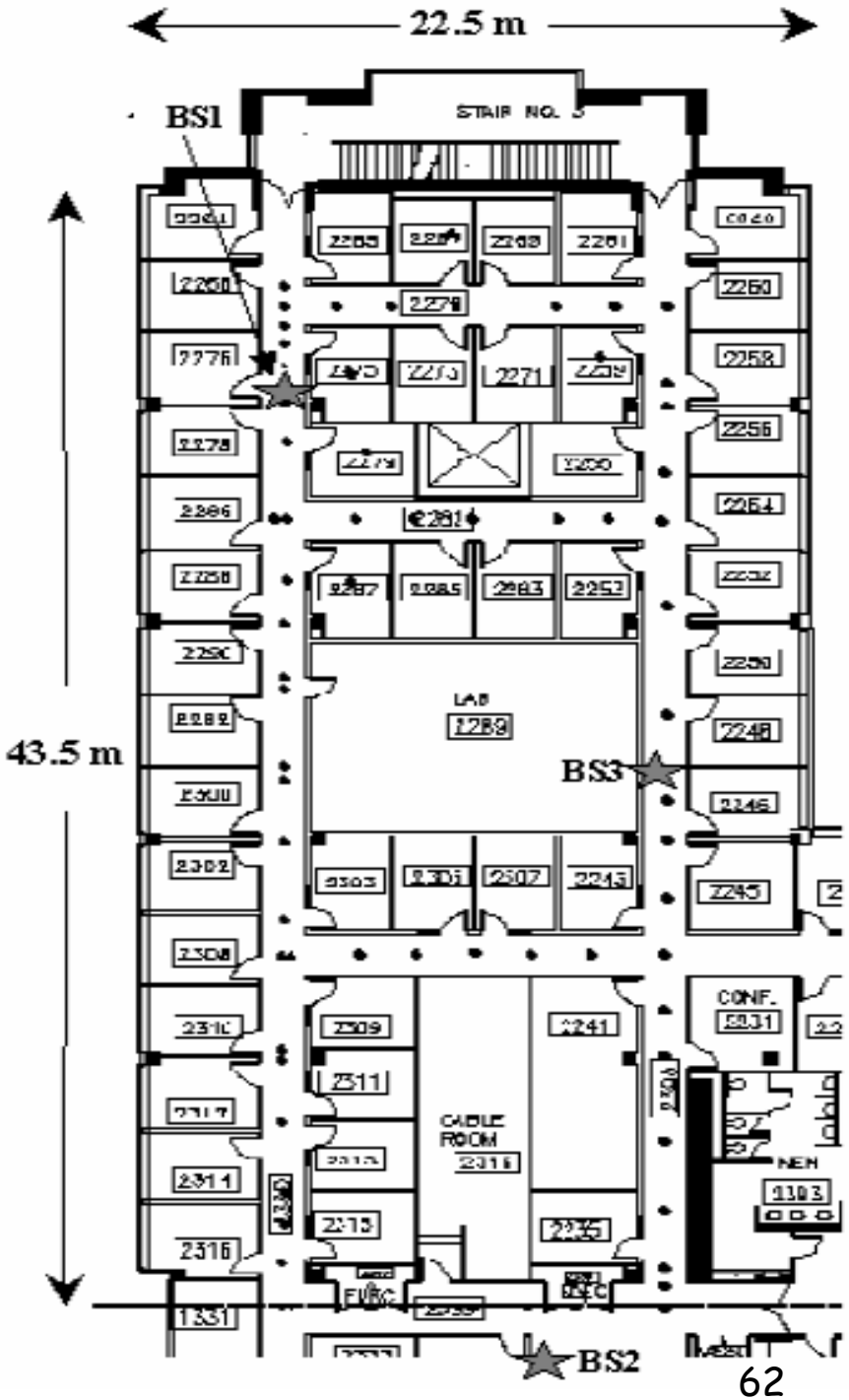
- Leverage the existing infrastructure of an indoor RF wireless LAN to build applications that take advantage of location information

Main Premise

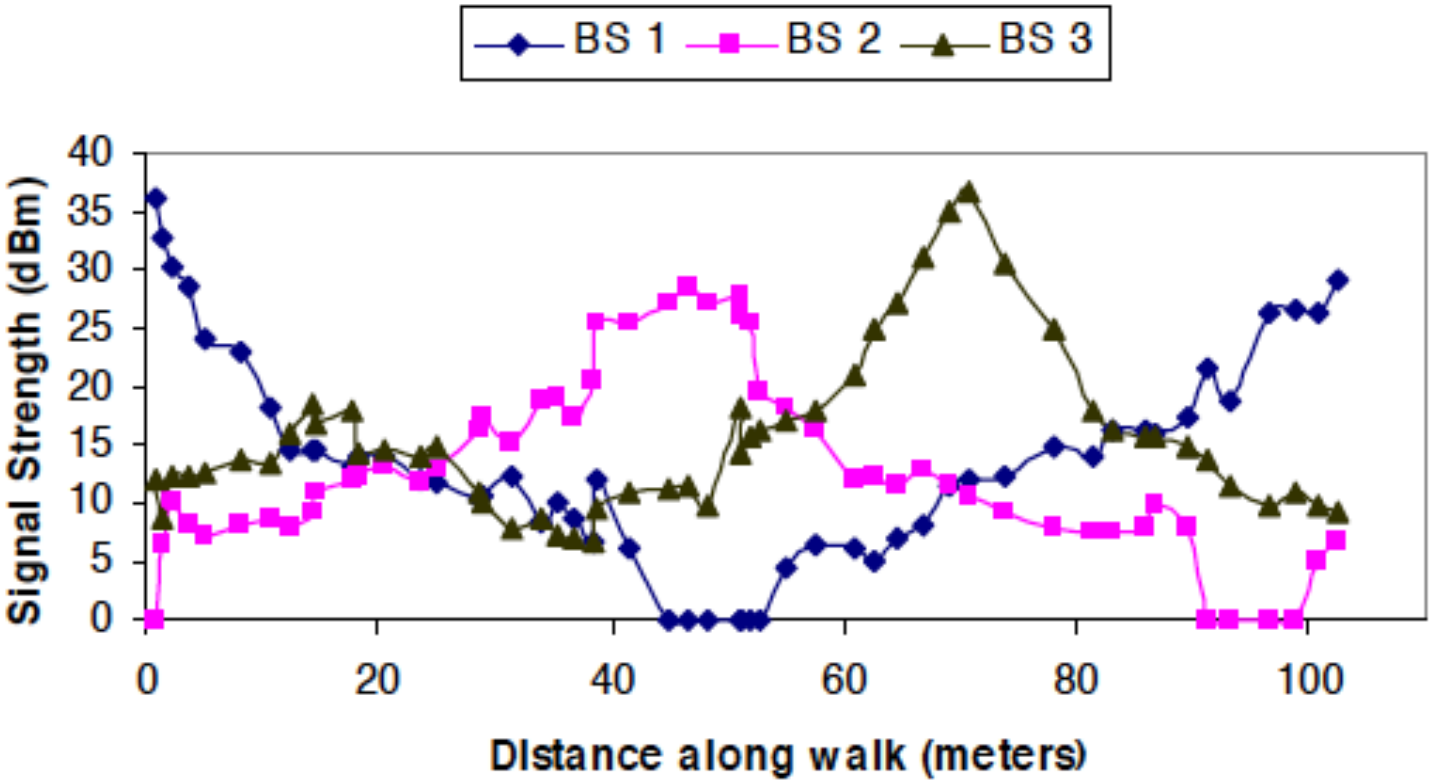
- There is a correlation between signal strength and location/distance

Experimental Testbed

- Black Dots = locations where signal strength info was collected
- Large Stars = Base Stations (BS)



How good an indicator of location is signal strength?



- User walks along the outer hallway of the floor in a counter-clockwise direction
- The walk begins and terminates close to BS1

Signal strength correlates well with distance

General Approach

- Key Idea: Map signal strengths to physical locations (Radio Fingerprinting)
- Inputs: Building geometry
- Training (Off-line) Phase: Construct a **Radio Map**
  - <location, Signal Strength (ss)> records in a database
- Operating Phase:
  - Extract SS from base station beacons
  - Transmit a location request to AP with SS as input
  - Find Radio Map entry that best matches the measured SS

Radio Map Construction (Off-Line)

Empirical Method

- Base Station emit beacons periodically: Measure SS based on beacons at various locations
- Record SS along with corresponding coordinates
  - User orientation needs to be included too
  - Tuples of the form $(x,y,z,d,s_1,\ldots,s_n)$
- Accurate but laborious

Mathematical Method

- Compute SS using a simple propagation model
  - Factor in path loss and wall attenuation
- More convenient but less accurate

Empirical Radio Map Construction

Measurement Based Map Construction

- Synchronize clocks on mobile host & base station
- Mobile hosts (it could have been the base station) broadcasts UDP packets
- Data are collected from 70 locations and 4 directions
- Each base station records SS at (t, x, y, d)
  - Time stamp (t)
  - Direction, d, user is facing (north, south, east, west)
  - User indicates location by clicking map on floor
- The data is combined in a common database where every entry is of the form (x,y,d,ss_i)
  - $i \in \{1,2,3\}$ corresponding to the three base stations

Operating Phase

- Input: The Radio Map and an observed signal strength (ss)
- Output: Location (x,y)
- Challenge: searching the radio map database for the best fit given a signal strength value

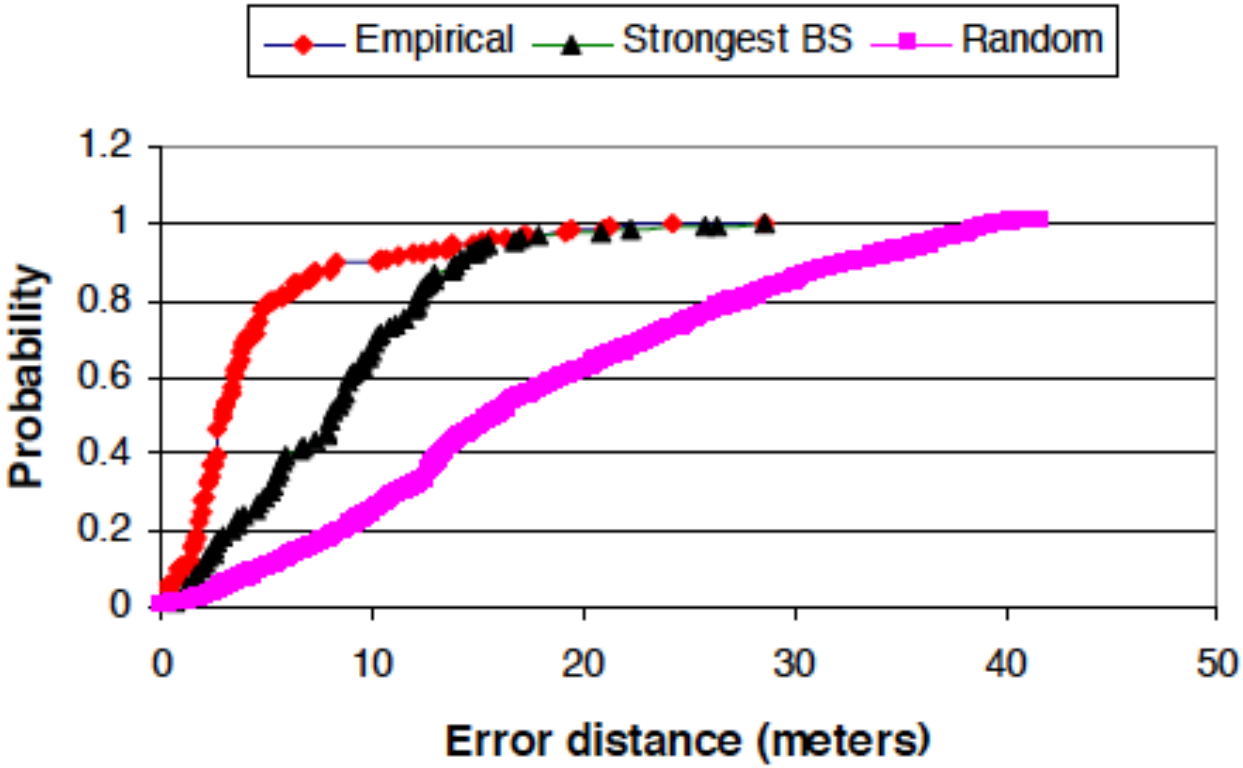
Mapping Signal Strength to Location

- Need a metric and a search methodology to compare multiple locations and pick the one that best matches the observed signal strength
- Approach: nearest neighbor(s) in signal space (NNSS)
  - Compute the distance (in signal space) between the observed set of SS measurements, (ss_1,ss_2,ss_3), and the SS, (ss'_1,ss'_2,ss'_3), at a fixed set of locations recorded in the radio map
  - Use the Euclidean distance measure $\text{sqrt}((ss_1-ss'_1)^2+(ss_2-ss'_2)^2+(ss_3-ss'_3)^2)$
  - Search linearly the radio map database and return the (x,y,d) for which the Euclidian is minimized

Performance Evaluation

- Select at random one of the 70 locations in the radio map database
  - Remove it from the database
- Try to locate it using the rest of the entries in the database
- Compare with two simplistic schemes to quantify how worthwhile the increased sophistication of Radar is
  - Random Selection of a point in the radio map
  - Strongest Base Station Selection: same location as the base station with the strongest signal

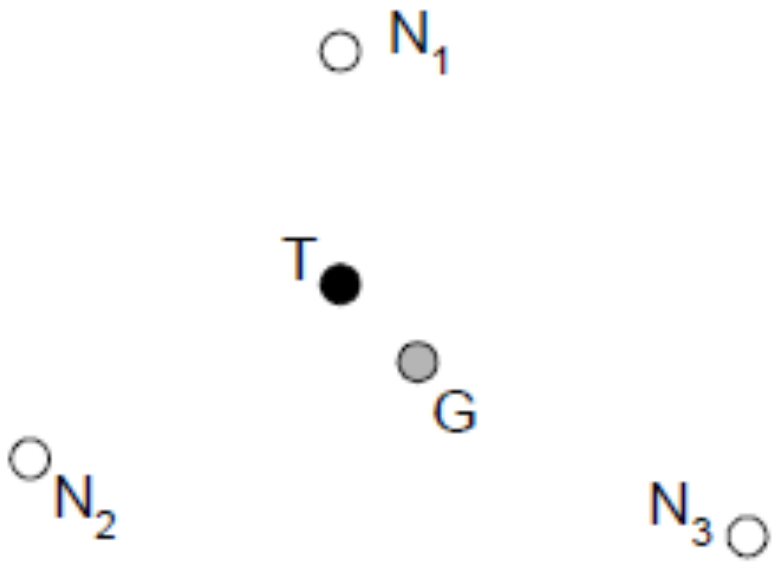
Location Estimate Error



Median error distance is 2.94 meters

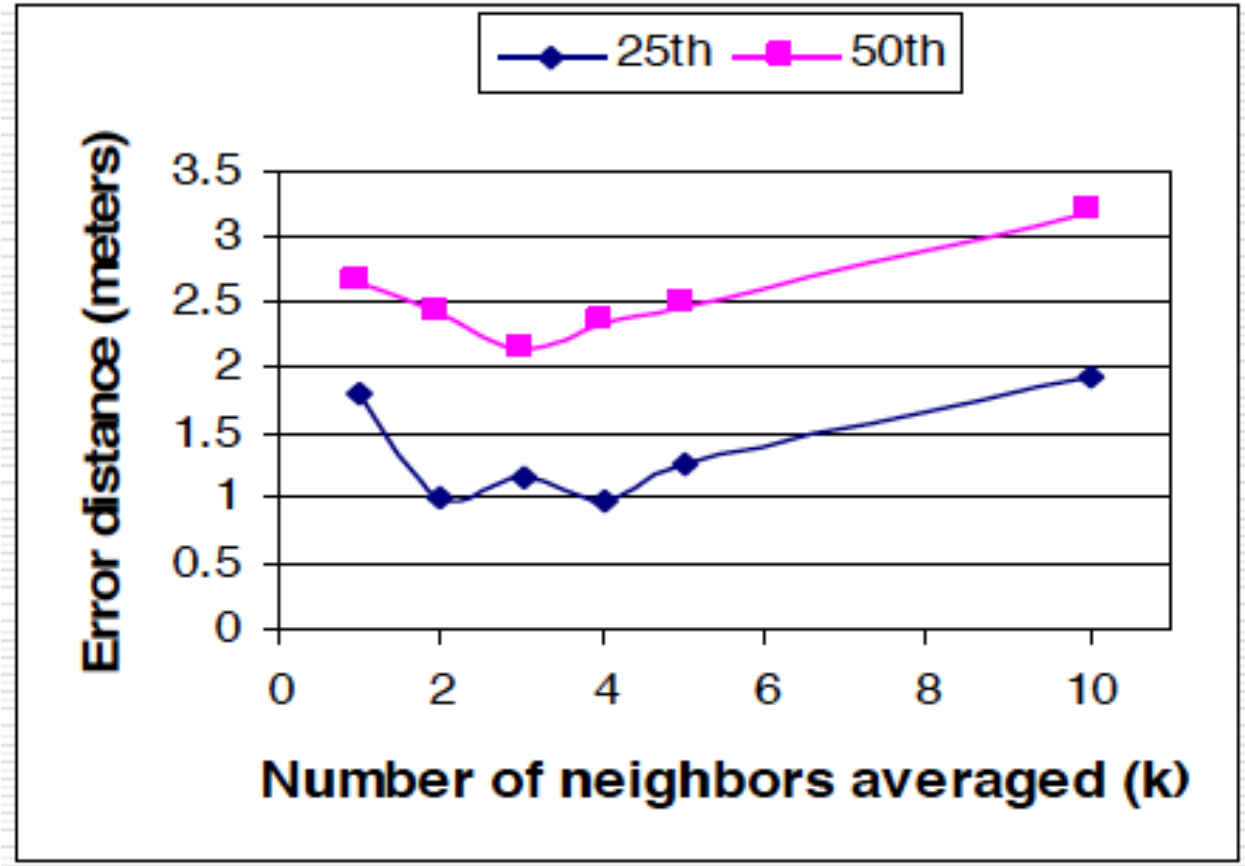
Multiple Nearest Neighbors

- Do not limit to just nearest data point (neighbor)
  - Average the coordinates of k nearest neighbors



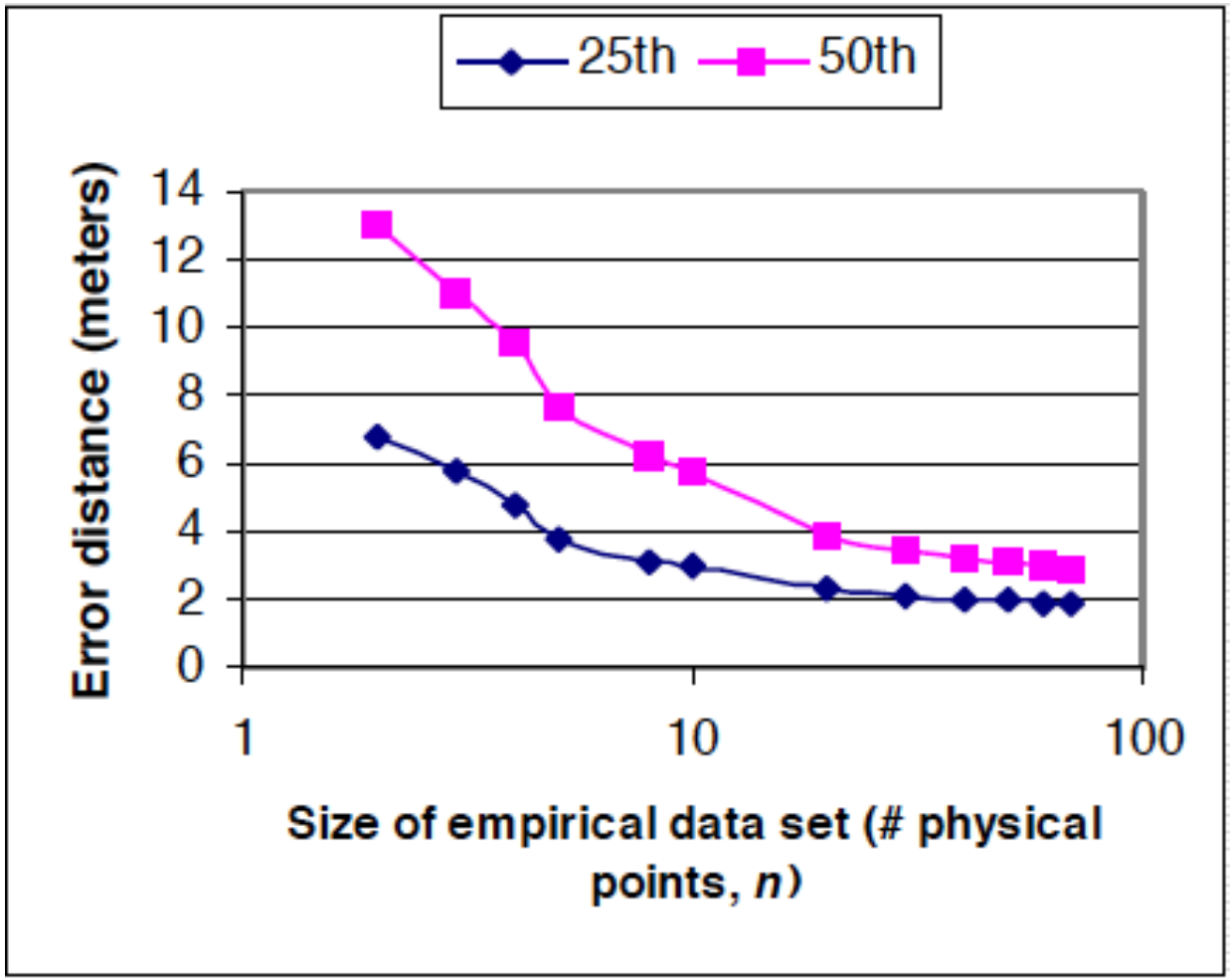
T: true location, G: guess

Performance with Averaging



Median error distance is 2.13 meters when averaging is done over 3 neighbors

How Extensive Does the Radio Map Have to Be?



Diminishing returns as the number of physical points mapped increases

Tracking a Mobile User

- Reduce the problem of tracking the mobile user to a sequence of location determination problems for a (nearly stationary) user
- 4 SS samples/second
- Use a sliding window of 10 samples to compute the mean signal strength on a continuous basis
- The median error distance observed was 3.5 meters, about 19% worse than that for a stationary user

Summary of the Empirical Method

- The empirical method is able to estimate user location with a high degree of accuracy
  - The median error distance is 2 to 3 meters, about the size of a typical office room
  - Much of the accuracy can be achieved with an empirical data set of about 40 physical points and about 3 real-time signal strength samples
- Long time to gather all the empirical data
- If BS moves, have to recollect all the data

LOCALIZATION BASED ON TIME OF FLIGHT (TOF)

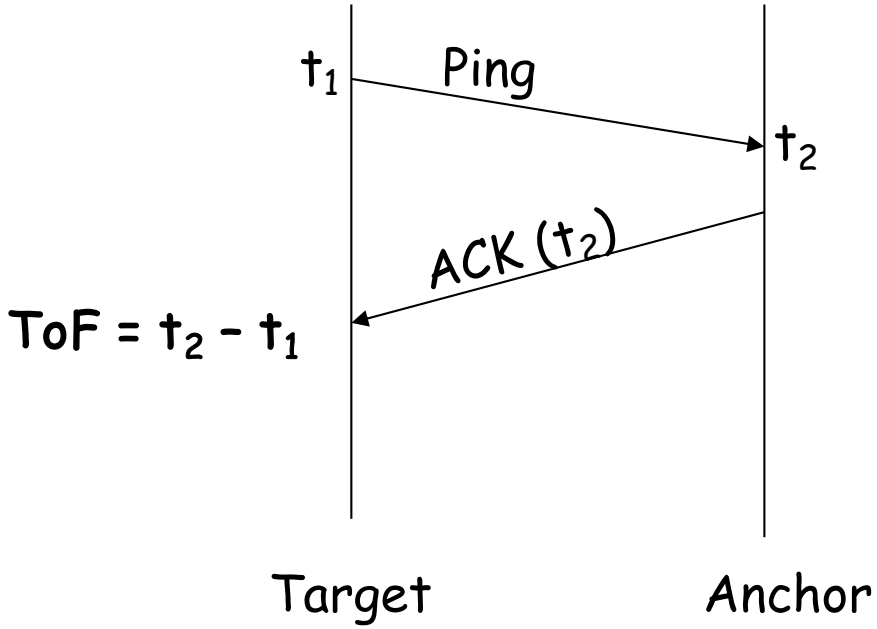
ToF-Based Localization

- The localization problem is reduced to a problem of computing the distance between the target a node and a set of anchor points whose coordinates are known

- Computing the distance between two devices
  - Equivalent to computing how long it takes for a wireless signals to travel the direct path between the devices – the Time of Flight (ToF)

Computing Time of Flight (ToF)

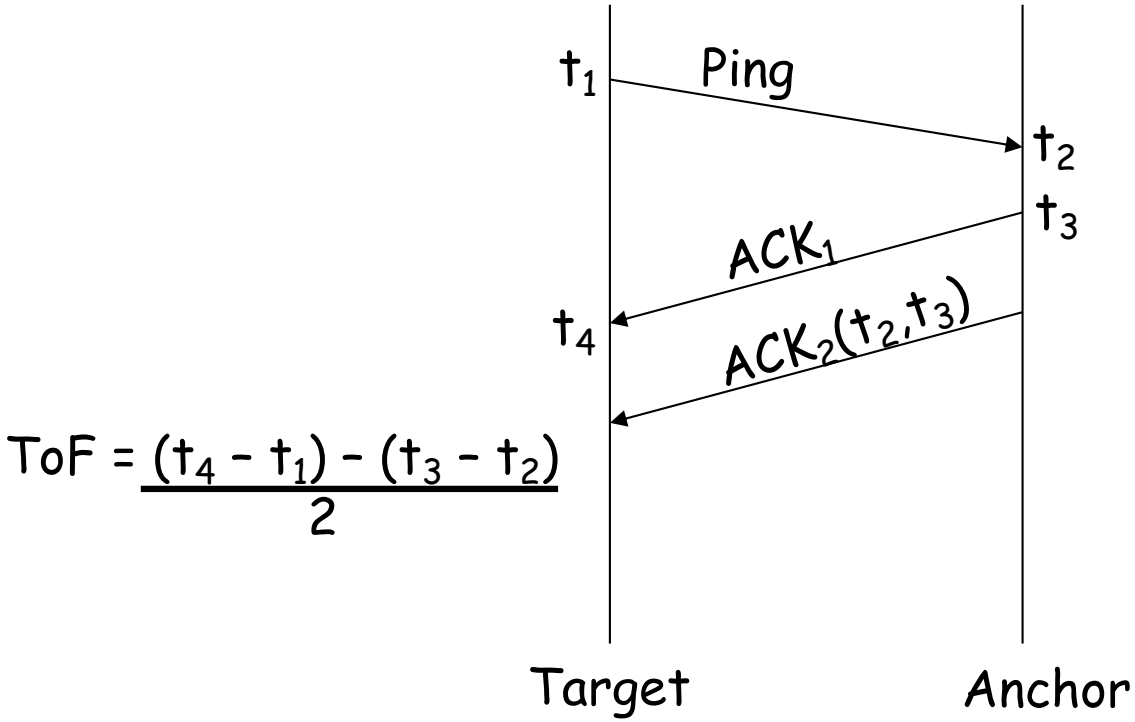
- Simplest approach



Any challenges ?

Computing Time of Flight (ToF)

- Two-way ranging (TWR)



Why is ACK_2 necessary?

WIFI FTM (IEEE 802.11AC)

IEEE Fine Time Measurement

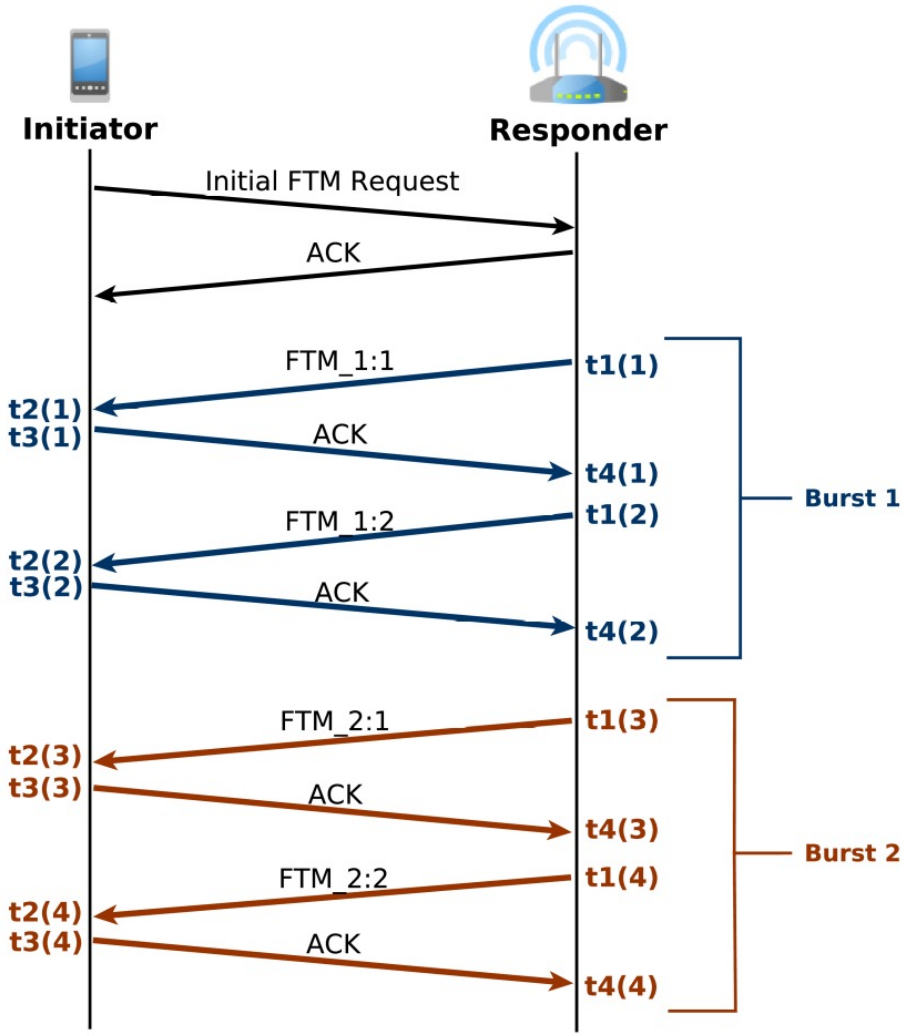
- Standardized as part of the IEEE 802.11-2016
  - Included as part of the 802.11mc amendment
- It enables a WiFi station to compute the distance to an access point in range without having to associate to the particular access point
  - It promises meter-level accuracy
- A standardized and native firmware implementation using clocks with picosecond resolution
- Supported by major WiFi manufacturers and it is adopted by the Android operating system
  - Google Pixel 2 and 3 phones, for example, are 802.11mc-compliant

IEEE Fine Time Measurement

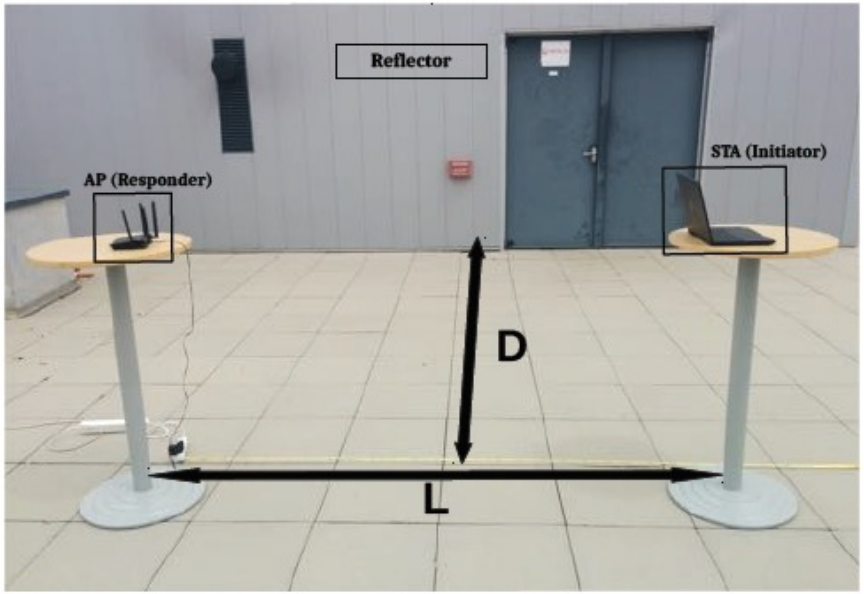
- The process starts with a WiFi station (called *initiator*) which scans for access points supporting FTM
- If an FTM-capable access point is detected, the initiator sends to the latter an FTM request frame
- Upon the reception that request, the access point can choose to ignore it, or to become a *responder*.
- The two stations start a series of (FTM, ACK) packet exchanges, called burst, allowing the initiator to estimate the round trip time (RTT) with the responder
- An FTM burst consists of the responder sending multiple FTM packets, which are all acknowledged by the initiator
  - Both stations capture the timestamps at which the burst packets are sent and received

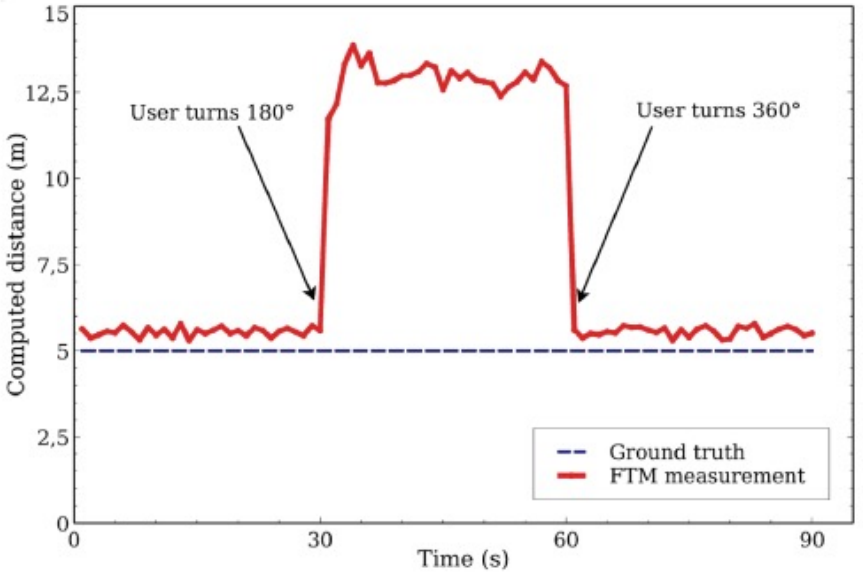
IEEE Fine Time Measurement

$$RTT = \frac{1}{N}\sum_{i=1}^{N}\left(t_4(i) - t_1(i)\right)$$

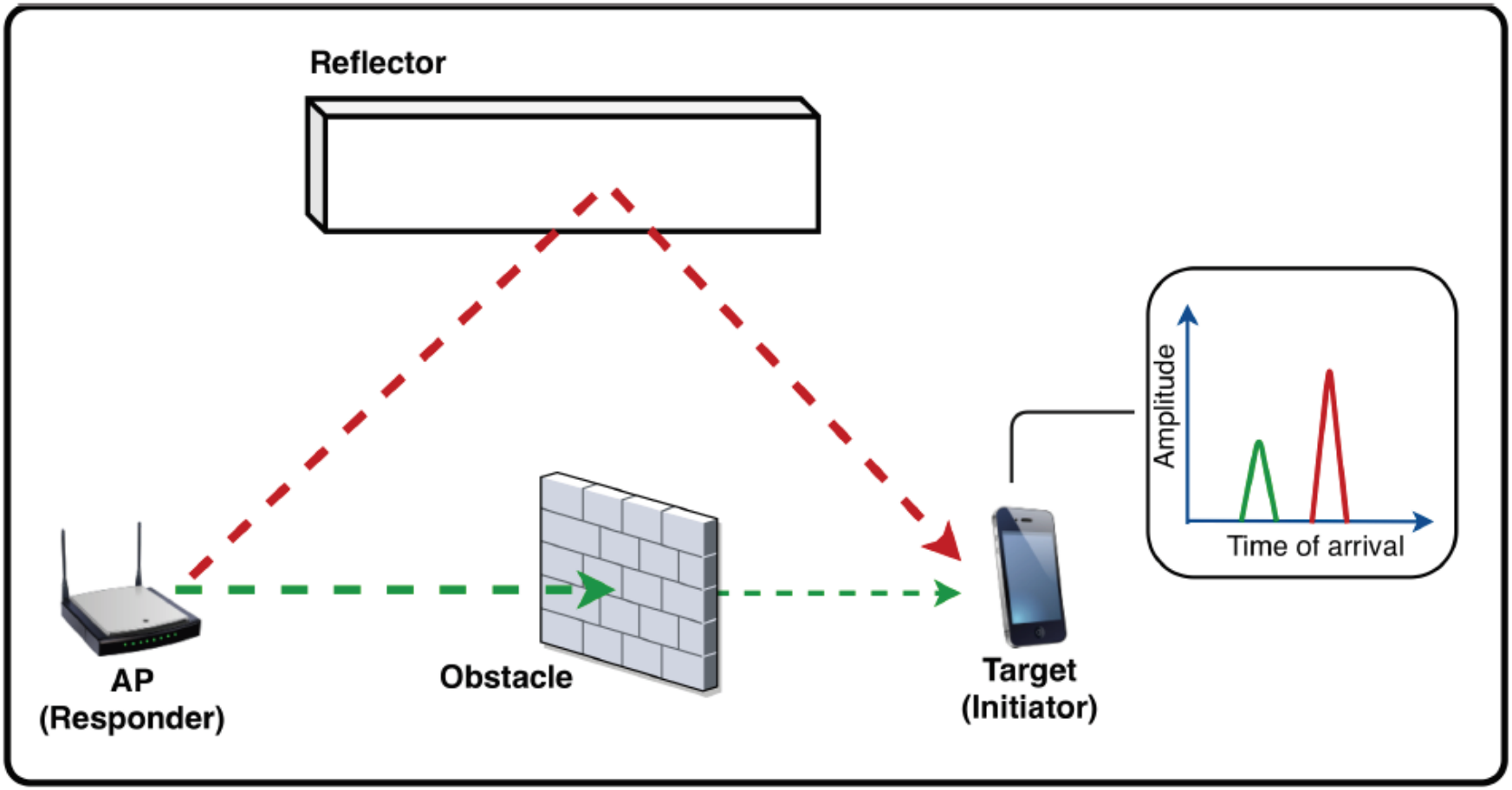


Does it work?


Reflector
AP (Responder)
STA (Initiator)
D
L

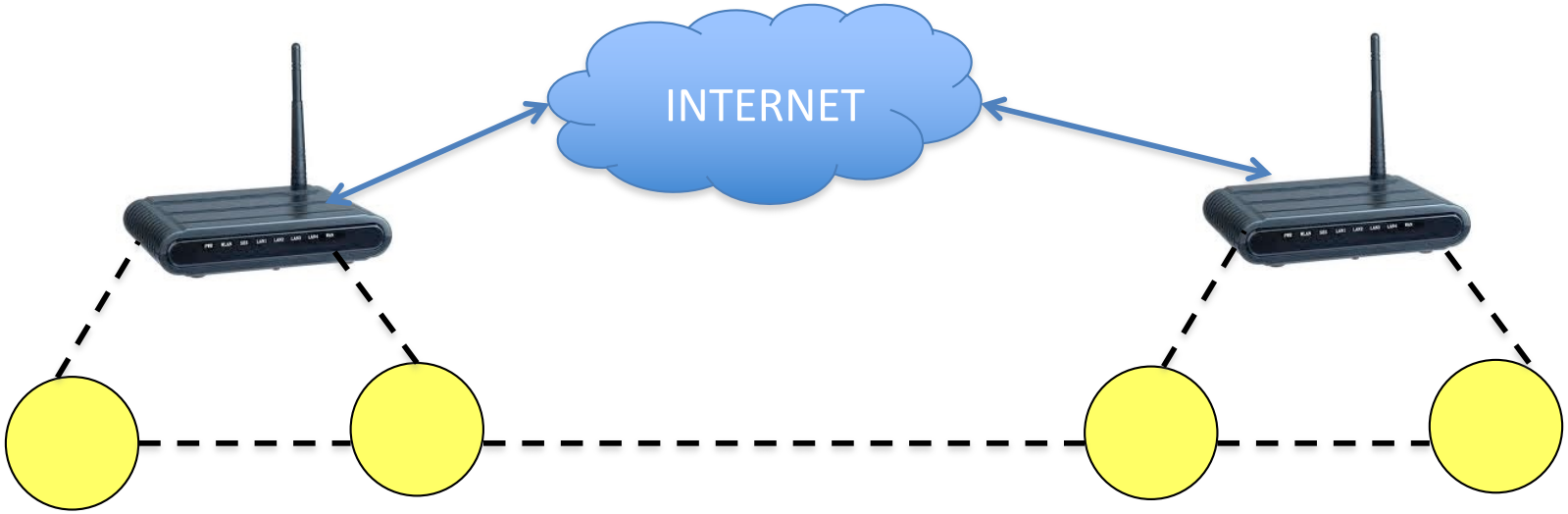

15
12,5
10
7,5
5
2,5
0
Computed distance (m)
User turns 180°
User turns 360°
Ground truth
FTM measurement
0
30
60
90
Time (s)

What happened?



Wi-Fi Mobile Ad-Hoc Architecture (MANET)

Wireless Ad Hoc Networks

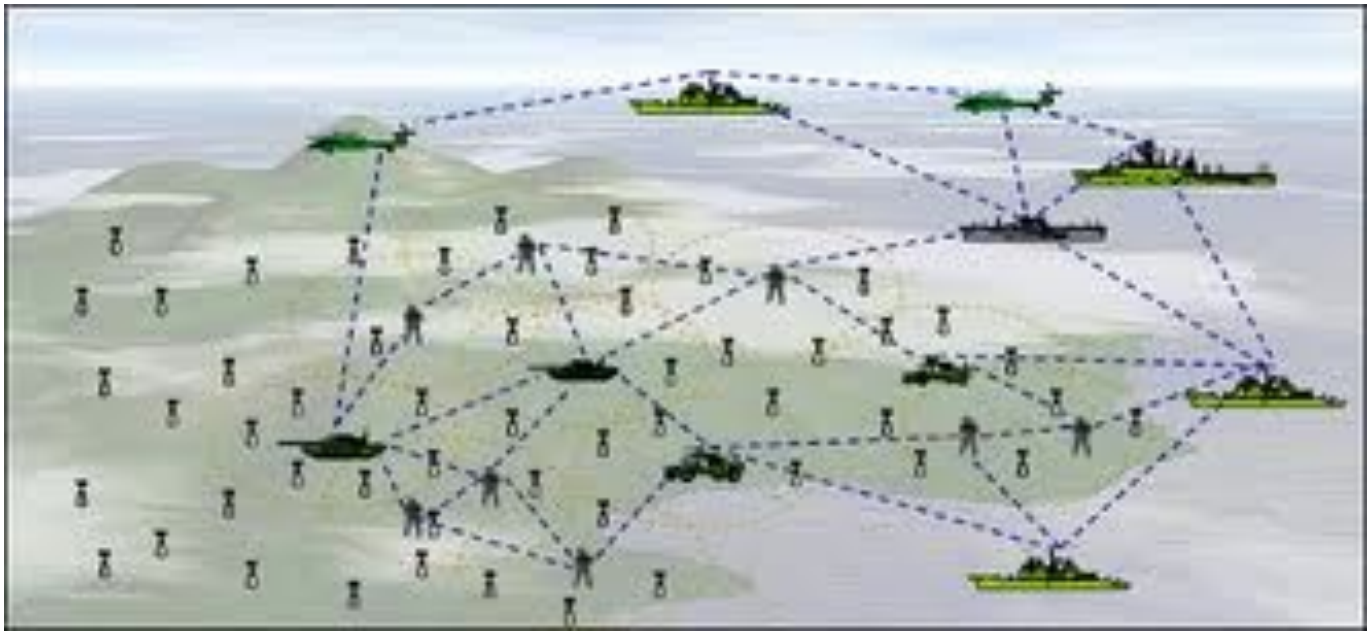


- Computers equipped with wireless communication devices create dynamically a network without prior infrastructure or coordination
- Every node is also a ***router***

Why Ad Hoc Networks ?

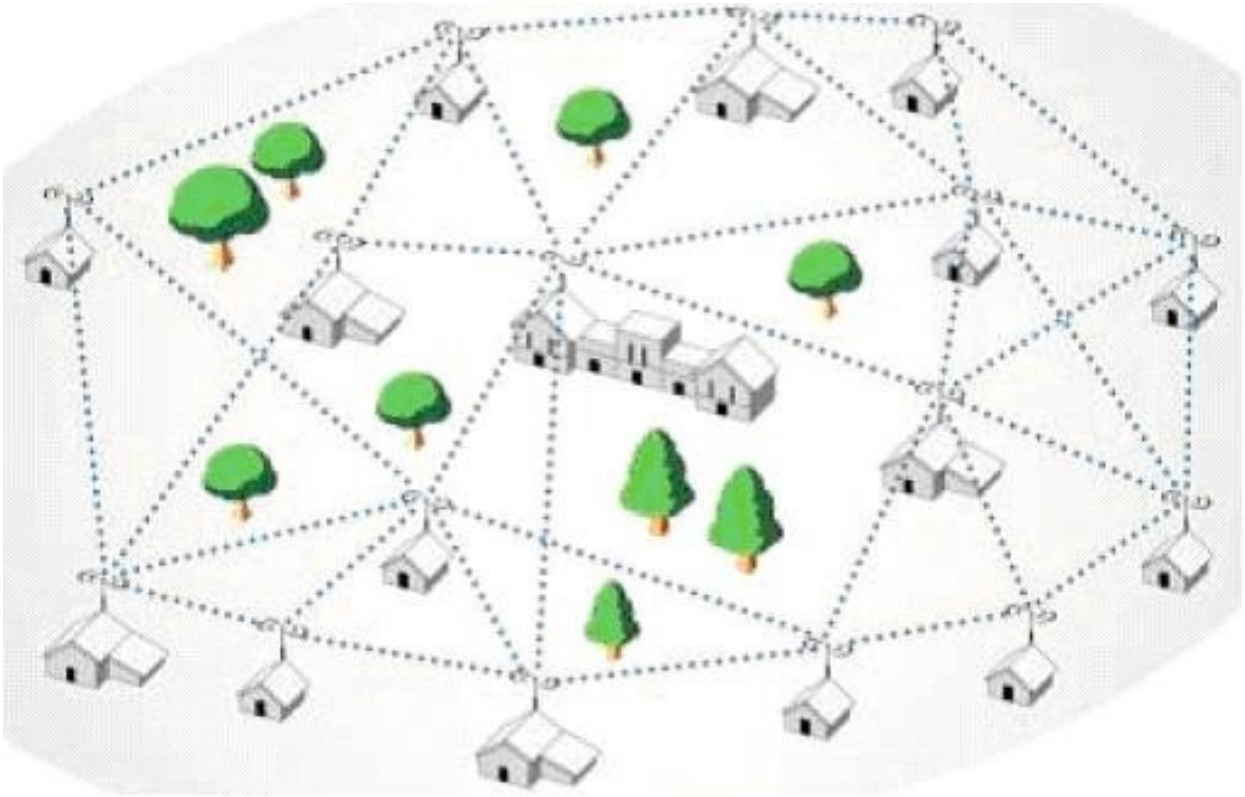
- Ease of deployment
- Speed of deployment
- Decreased dependence on infrastructure

Military Applications



- Enable voice and data communication in the battlefield where there is no Orange/SFR/Wi-Fi
- DARPA packet radio network (1973-1987) the first ad hoc network

Civilian Applications



- Enable communications in community networks
- Remote, developing regions (one laptop/child)
- The communication devices are static: mesh networks

Research Challenges

- MAC
- *Routing*
- Scalability
- Security
- Etc …

Many Variations

- Fully Symmetric Environment
  - all nodes have identical capabilities and responsibilities
- Asymmetric Capabilities
  - transmission ranges and radios may differ
  - battery life at different nodes may differ
  - processing capacity may be different at different nodes
  - speed of movement
- Asymmetric Responsibilities
  - only some nodes may route packets
  - some nodes may act as leaders of nearby nodes (e.g., cluster head)

Many Variations

- Traffic characteristics may differ in different ad hoc networks
  - bit rate
  - timeliness constraints
  - reliability requirements
  - unicast / multicast / geocast
  - host-based addressing / content-based addressing / capability-based addressing
- May co-exist (and co-operate) with an infrastructure-based network

Many Variations

- Mobility patterns may be different
  - people sitting at an airport lounge
  - New York taxi cabs
  - kids playing
  - military movements
  - personal area network
- Mobility characteristics
  - speed
  - predictability
    - direction of movement
    - pattern of movement
  - uniformity (or lack thereof) of mobility characteristics among different nodes

Engineering Challenges

- MAC
- Routing
- Scalability
- Security

Challenges

- Broadcast nature of the wireless medium
  - Hidden terminal problem
- Packet losses due to transmission errors
- Mobility-induced route changes
- Mobility-induced packet losses
- Battery constraints
- Potentially frequent network partitions
- Ease of snooping on wireless transmissions (security hazard)

Unicast Routing
in
Mobile Ad Hoc Networks

Why is Routing in MANET different ?

- Host mobility
  - link failure/repair due to mobility may have different characteristics than those due to other causes
- Rate of link failure/repair may be high when nodes move fast
- New performance criteria may be used
  - route stability despite mobility
  - energy consumption

Unicast Routing Protocols

- Many protocols have been proposed
- Some have been invented specifically for MANET
- Others are adapted from previously proposed protocols for wired networks
- No single protocol works well in all environments
  - some attempts made to develop adaptive protocols

Routing Protocols

- Proactive protocols
  - Determine routes independent of traffic pattern
  - Traditional link-state and distance-vector routing protocols are proactive
- Reactive protocols
  - Maintain routes only if needed
- Hybrid protocols

Trade-Off

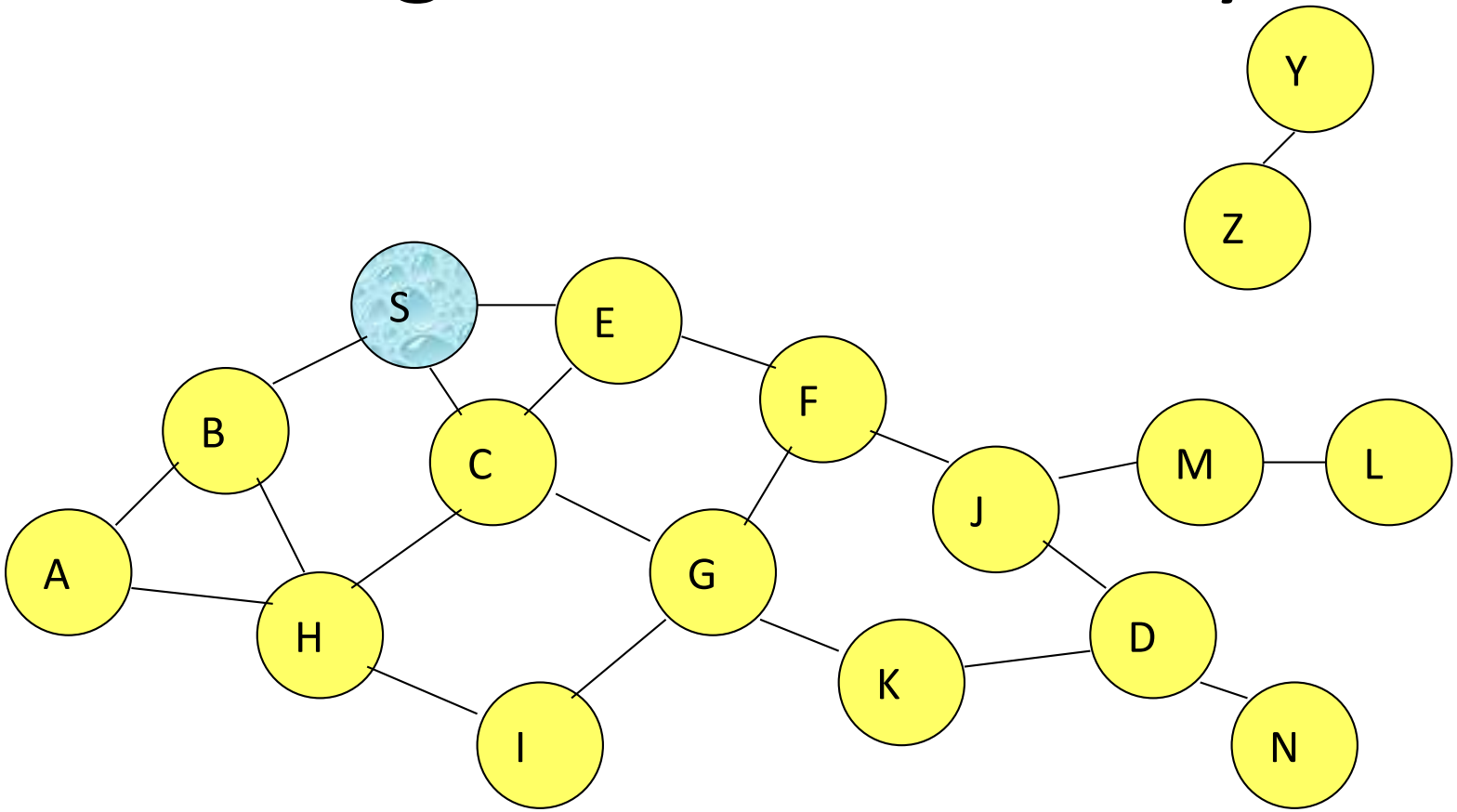
- Latency of route discovery
  - Proactive protocols may have lower latency since routes are maintained at all times
  - Reactive protocols may have higher latency because a route from X to Y will be found only when X attempts to send to Y
- Overhead of route discovery/maintenance
  - Reactive protocols may have lower overhead since routes are determined only if needed
  - Proactive protocols can (but not necessarily) result in higher overhead due to continuous route updating
- Which approach achieves a better trade-off depends on the traffic and mobility patterns

Overview of Unicast Routing Protocols

Flooding for Data Delivery

- Sender S broadcasts data packet P to all its neighbors
- Each node receiving P forwards P to its neighbors
- Sequence numbers used to avoid the possibility of forwarding the same packet more than once
- Packet P reaches destination D provided that D is reachable from sender S
- Node D does not forward the packet

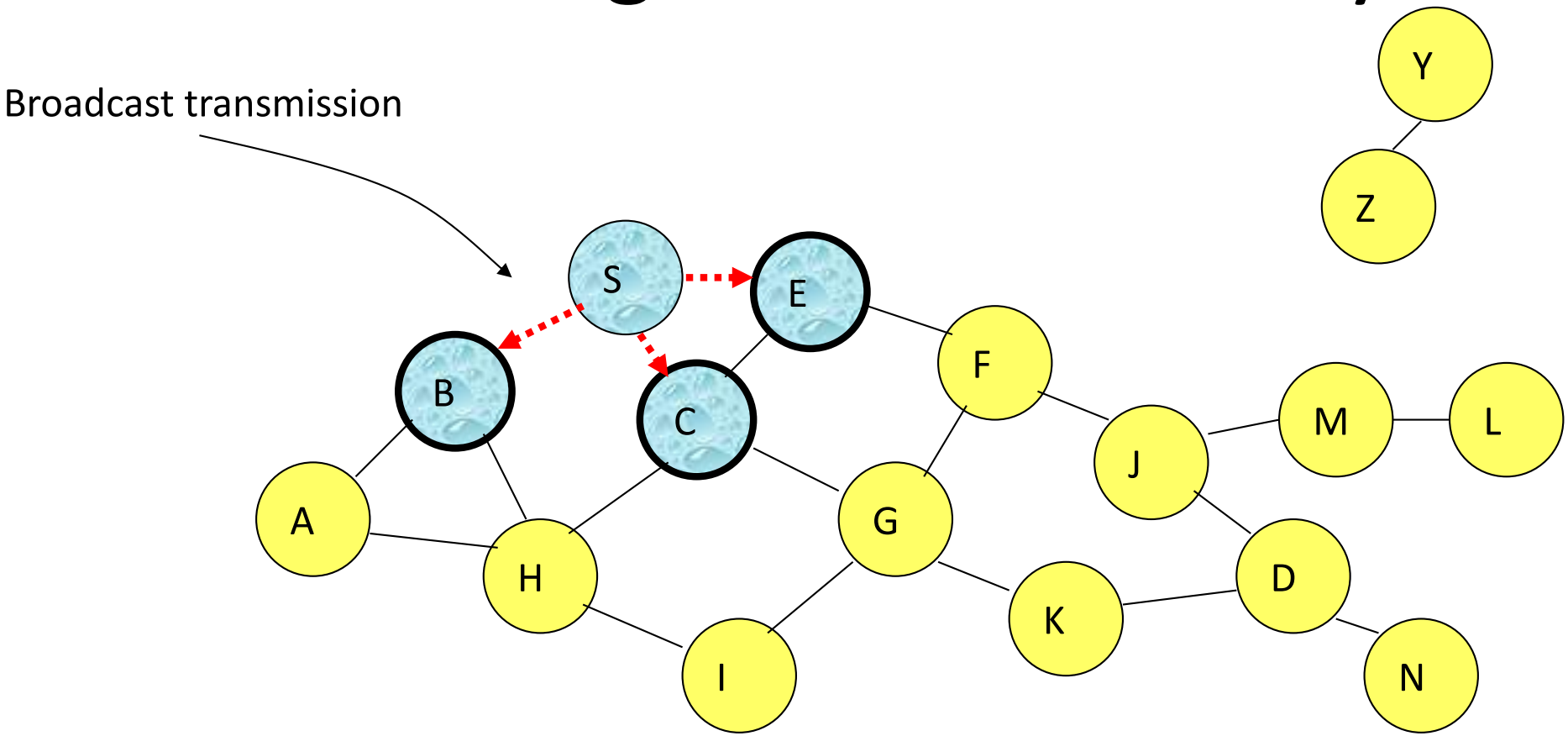
Flooding for Data Delivery



Represents a node that has received packet P

Represents that connected nodes are within each other's transmission range

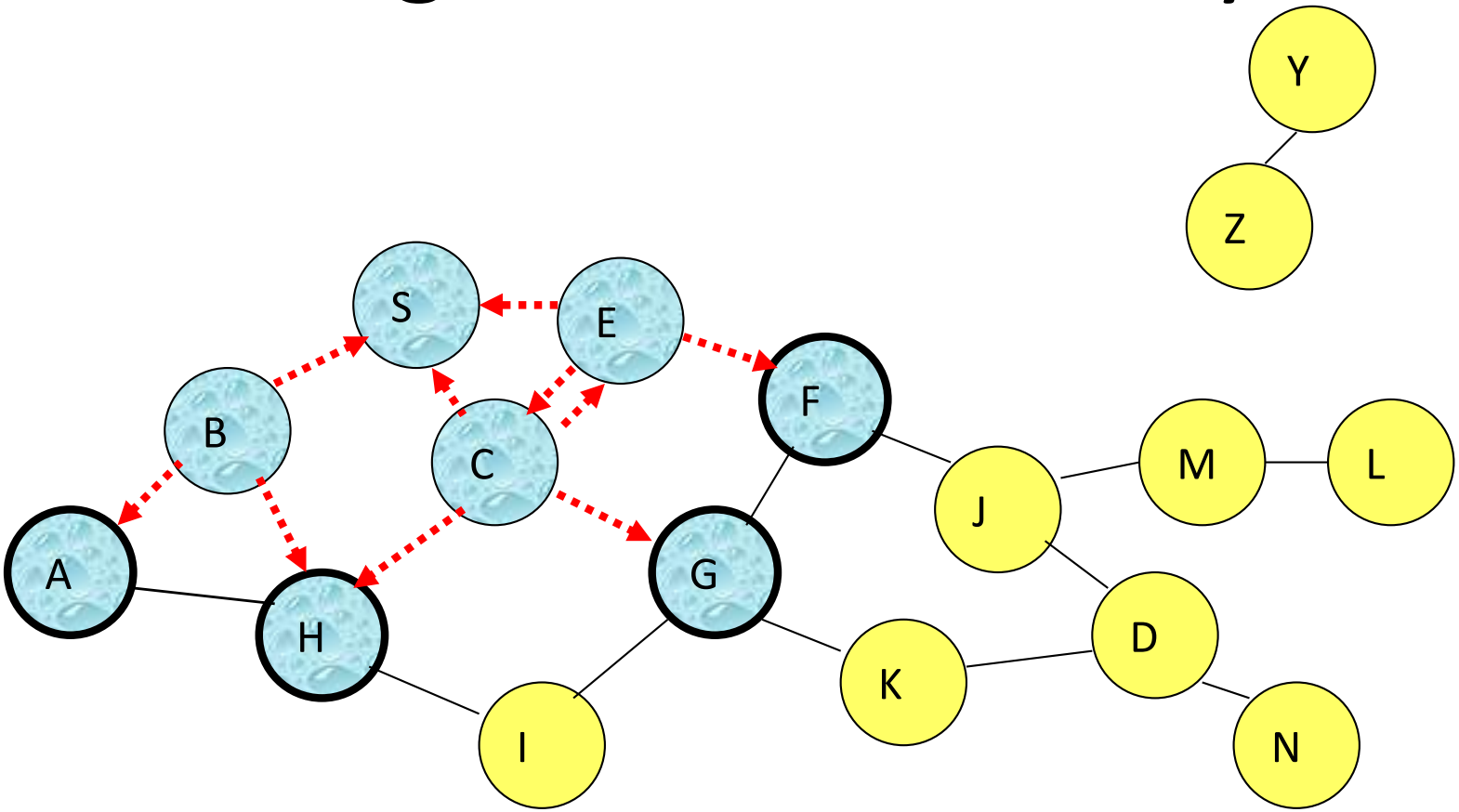
Flooding for Data Delivery



Represents a node that receives packet P for the first time

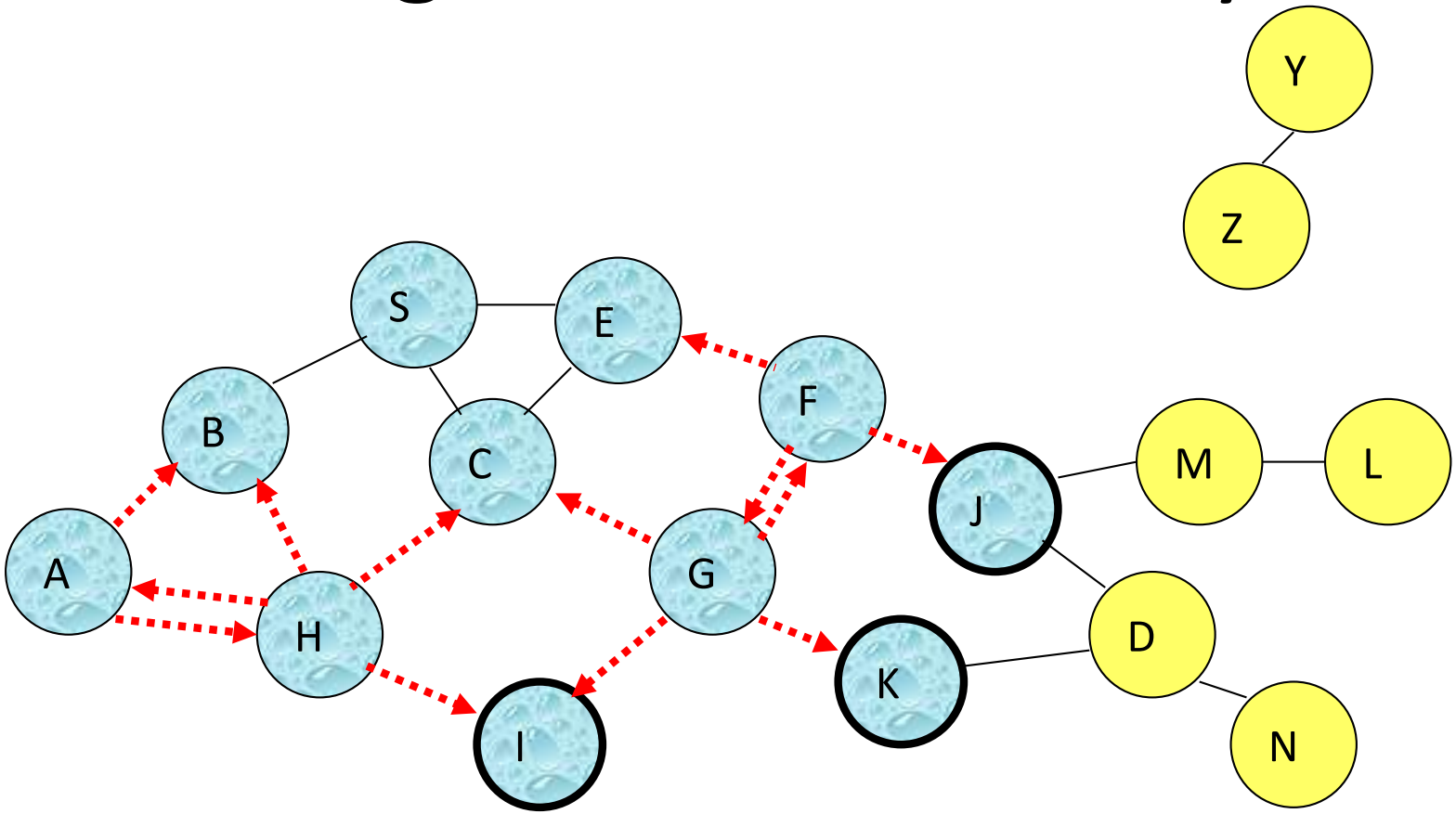
Represents transmission of packet P

Flooding for Data Delivery



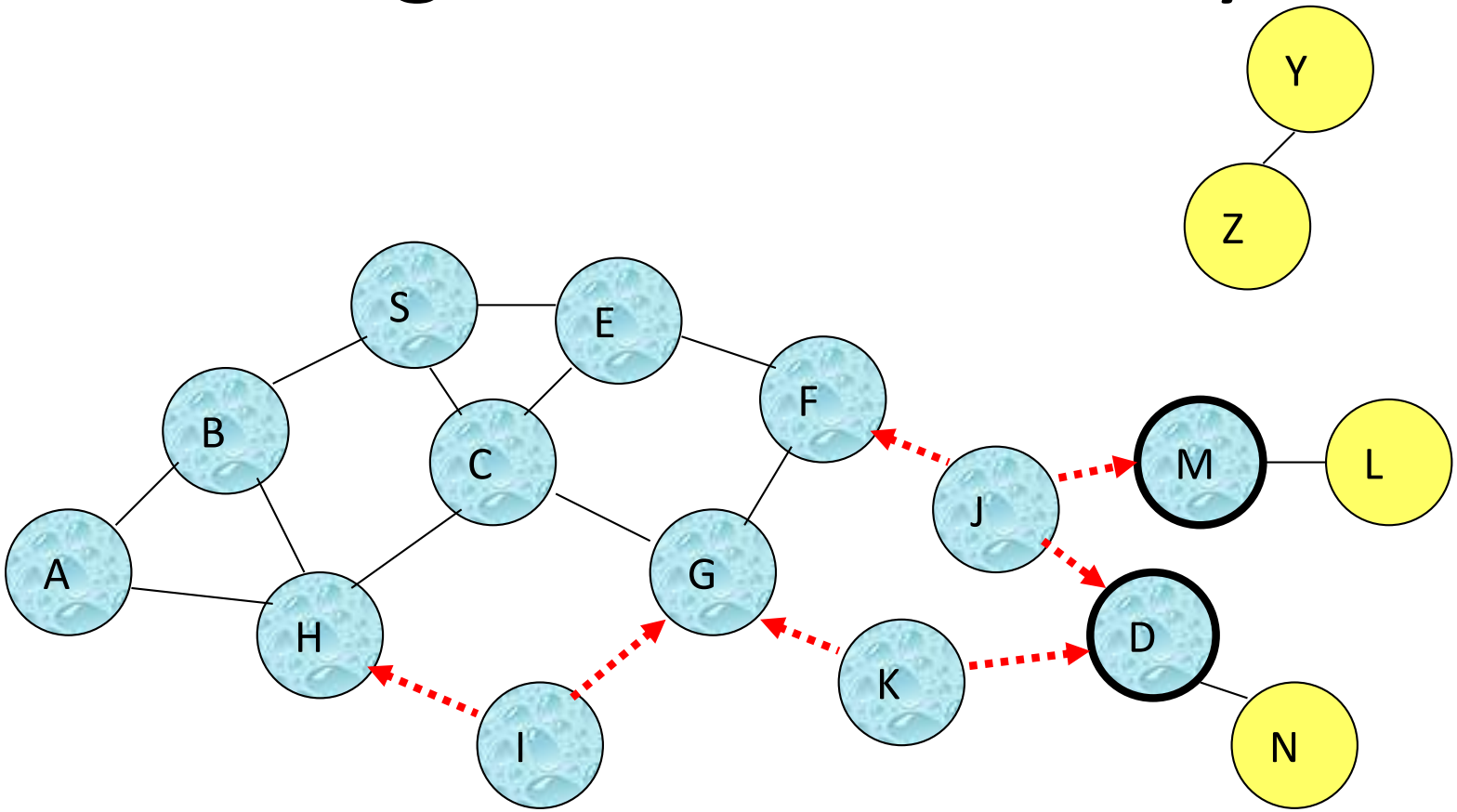
- Node H receives packet P from two neighbors: potential for collision

Flooding for Data Delivery



- Node C receives packet P from G and H, but does not forward it again, because node C has already forwarded packet P once

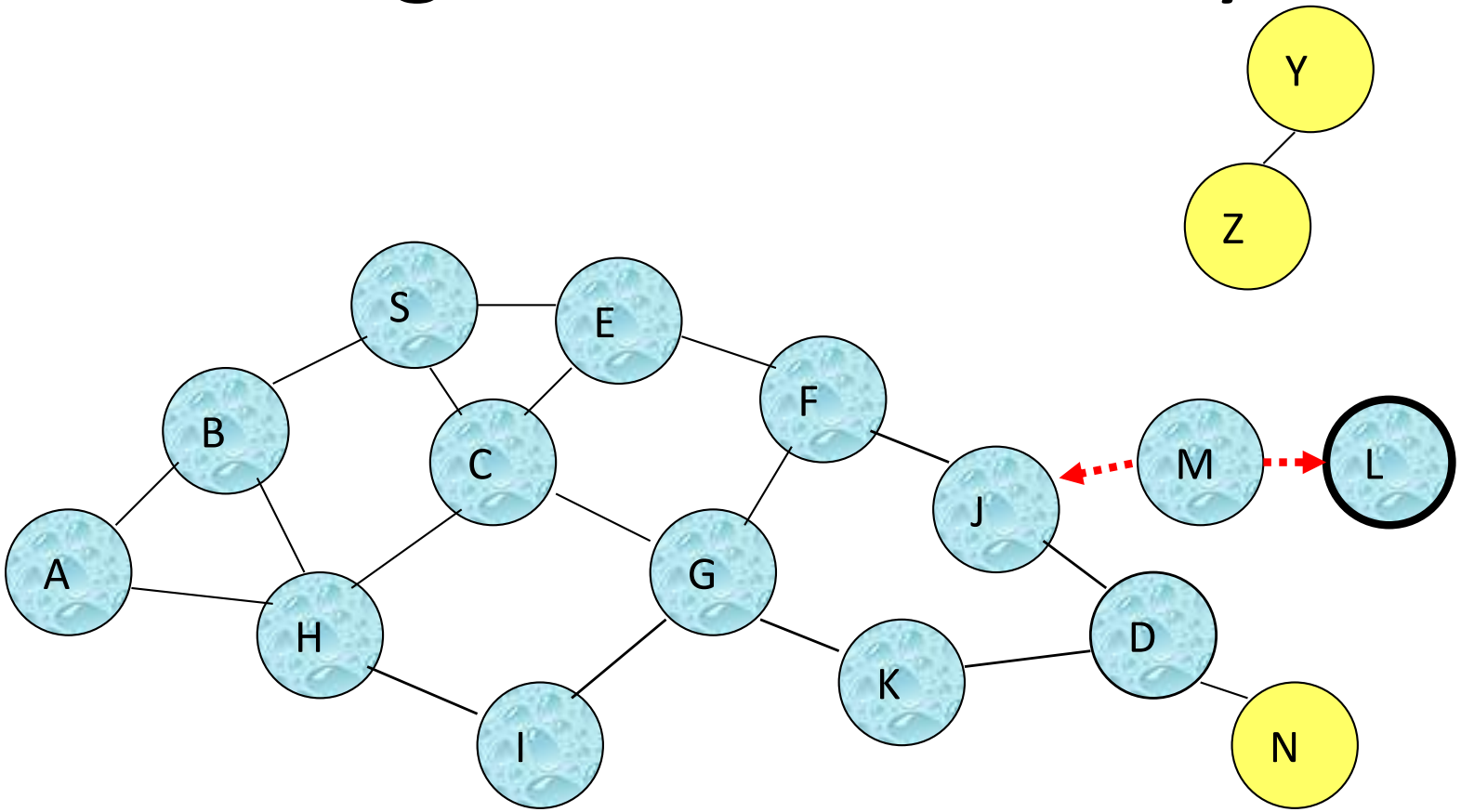
Flooding for Data Delivery



- Nodes J and K both broadcast packet P to node D
- Since nodes J and K are hidden from each other, their transmissions may collide

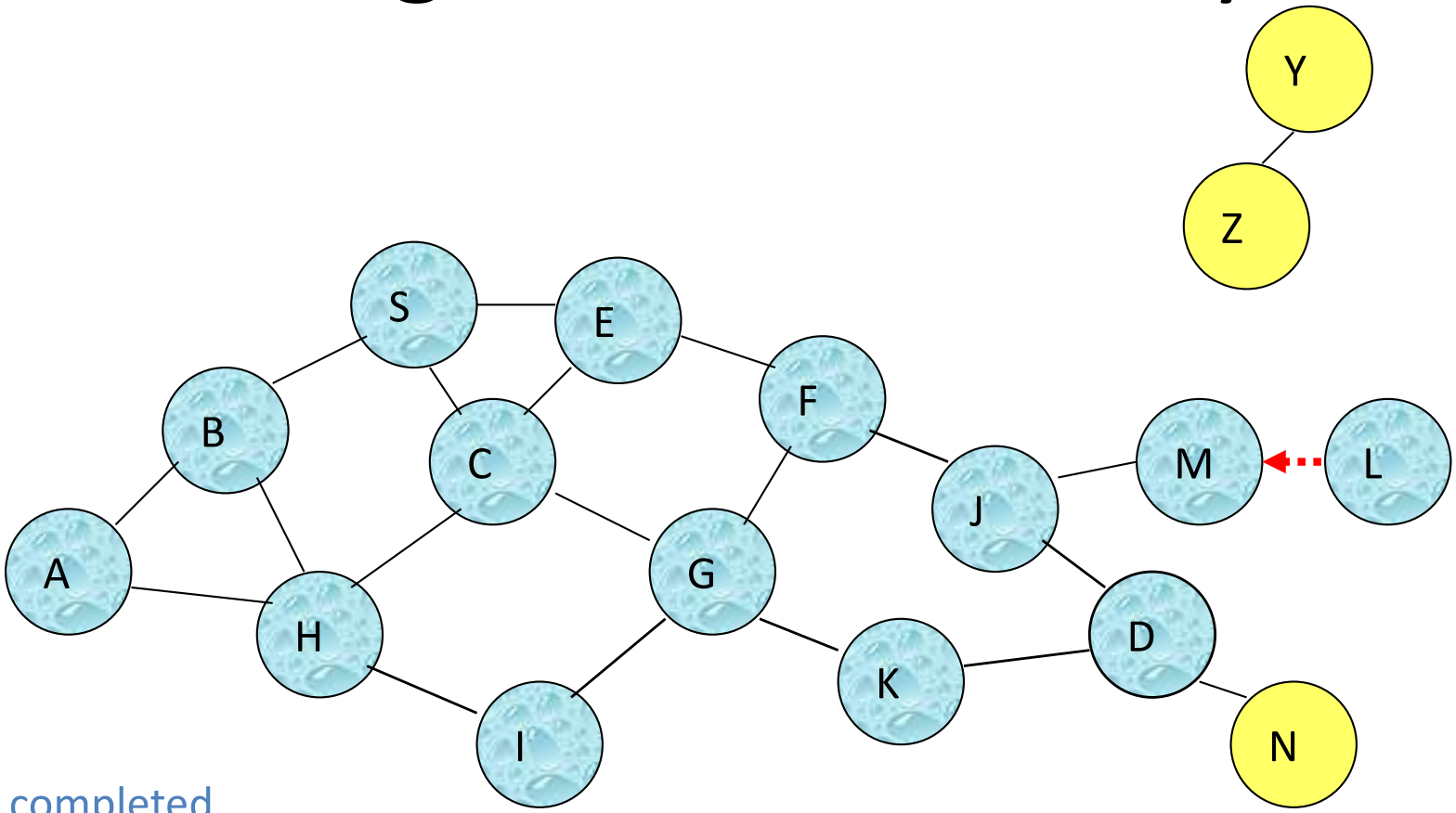
  => Packet P may not be delivered to node D at all, despite the use of flooding

Flooding for Data Delivery



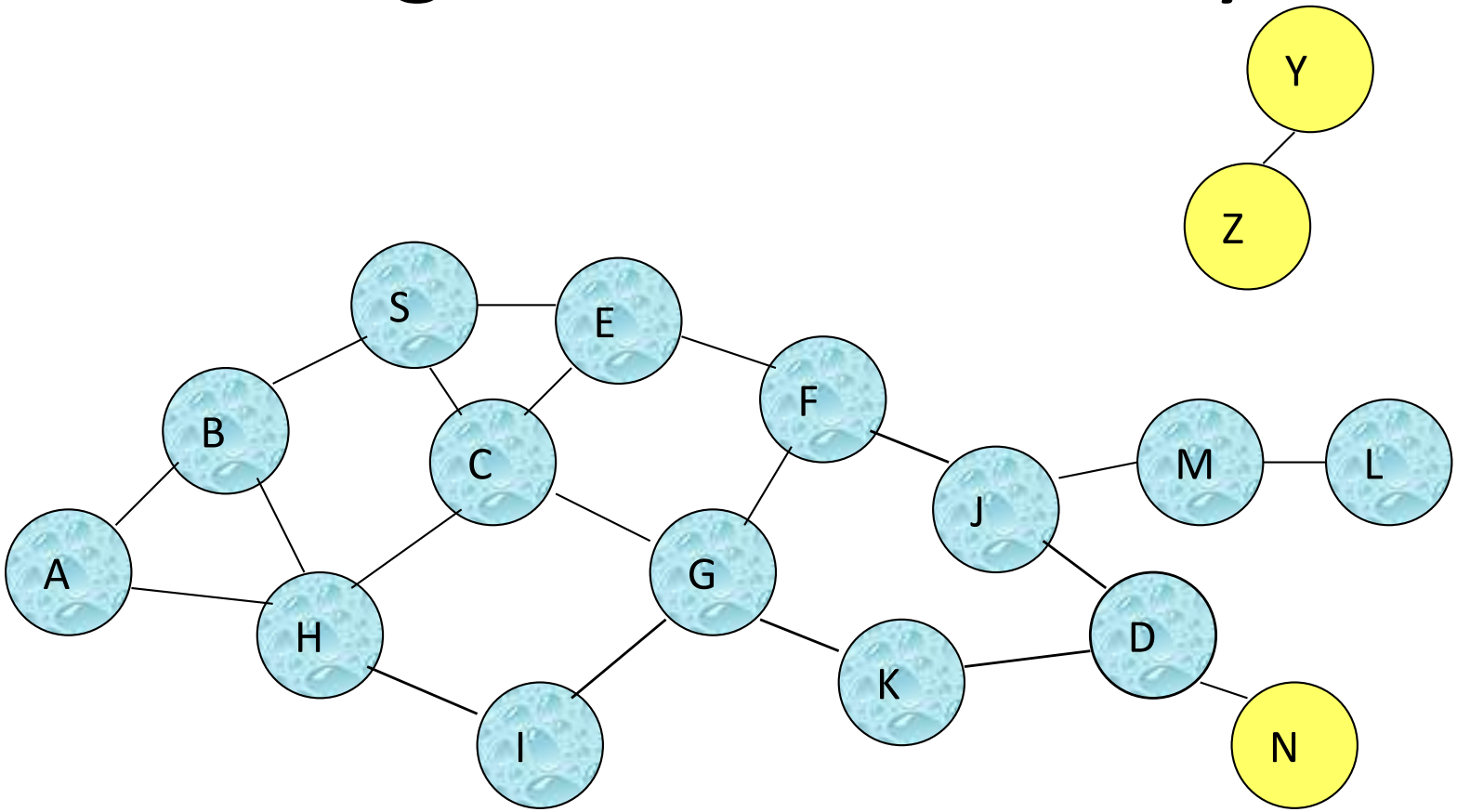
- Node D does not forward packet P, because node D is the intended destination of packet P

Flooding for Data Delivery



- Flooding completed
- Nodes unreachable from S do not receive packet P (e.g., node Z)
- Nodes for which all paths from S go through the destination D also do not receive packet P (example: node N)

Flooding for Data Delivery



- Flooding may deliver packets to too many nodes (in the worst case, all nodes reachable from sender may receive the packet)

Flooding for Data Delivery: Advantages

- Simplicity
- May be more efficient than other protocols when rate of information transmission is low enough that the overhead of explicit route discovery/maintenance incurred by other protocols is relatively higher
  - this scenario may occur, for instance, when nodes transmit small data packets relatively infrequently, and many topology changes occur between consecutive packet transmissions
- Potentially higher reliability of data delivery
  - Because packets may be delivered to the destination on multiple paths

Flooding for Data Delivery: Disadvantages

- Potentially, very high overhead
  - Data packets may be delivered to too many nodes who do not need to receive them
- Potentially lower reliability of data delivery
  - Flooding uses broadcasting -- hard to implement reliable broadcast delivery without significantly increasing overhead
    - Broadcasting in IEEE 802.11 MAC is unreliable
  - In our example, nodes J and K may transmit to node D simultaneously, resulting in loss of the packet
    - in this case, destination would not receive the packet at all

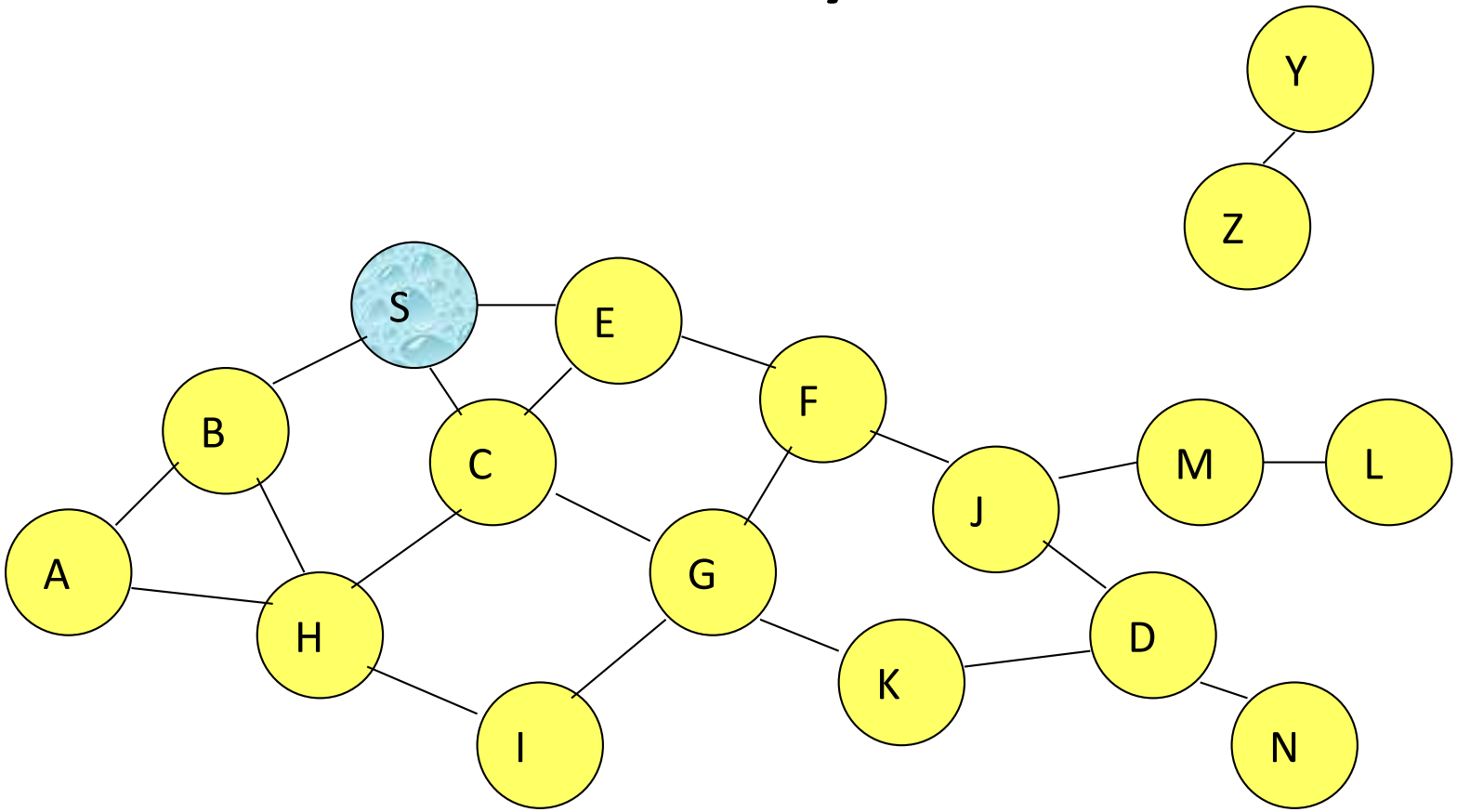
Flooding of Control Packets

- Many protocols perform (potentially *limited*) flooding of control packets, instead of data packets
- The control packets are used to discover routes
- Discovered routes are subsequently used to send data packet(s)
- Overhead of control packet flooding is amortized over data packets transmitted between consecutive control packet floods

Dynamic Source Routing (DSR) [Johnson96]

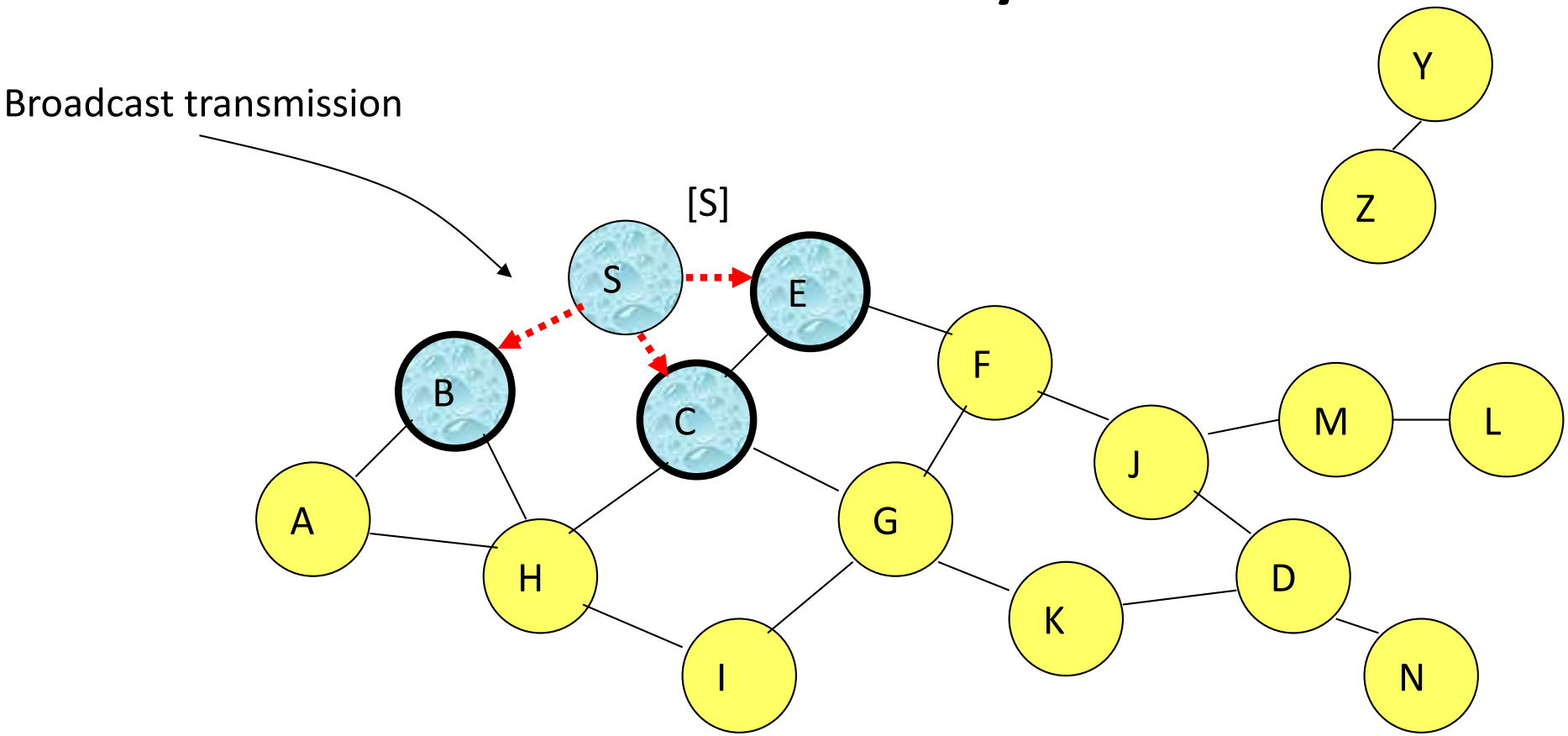
- When node S wants to send a packet to node D, but does not know a route to D, node S initiates a route discovery
- Source node S floods Route Request (RREQ)
- Each node appends own identifier when forwarding RREQ

Route Discovery in DSR



Represents a node that has received RREQ for D from S

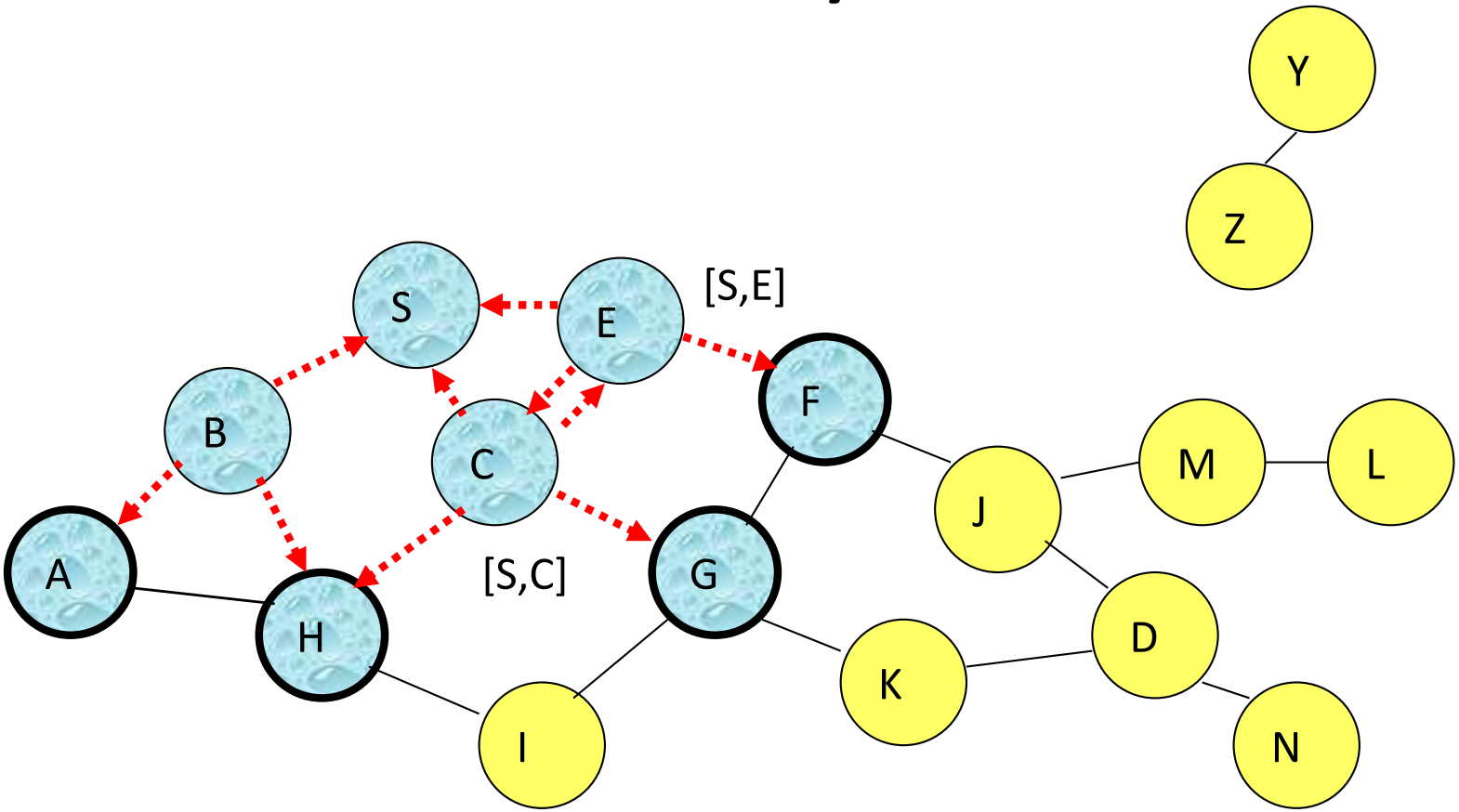
Route Discovery in DSR



Represents transmission of RREQ

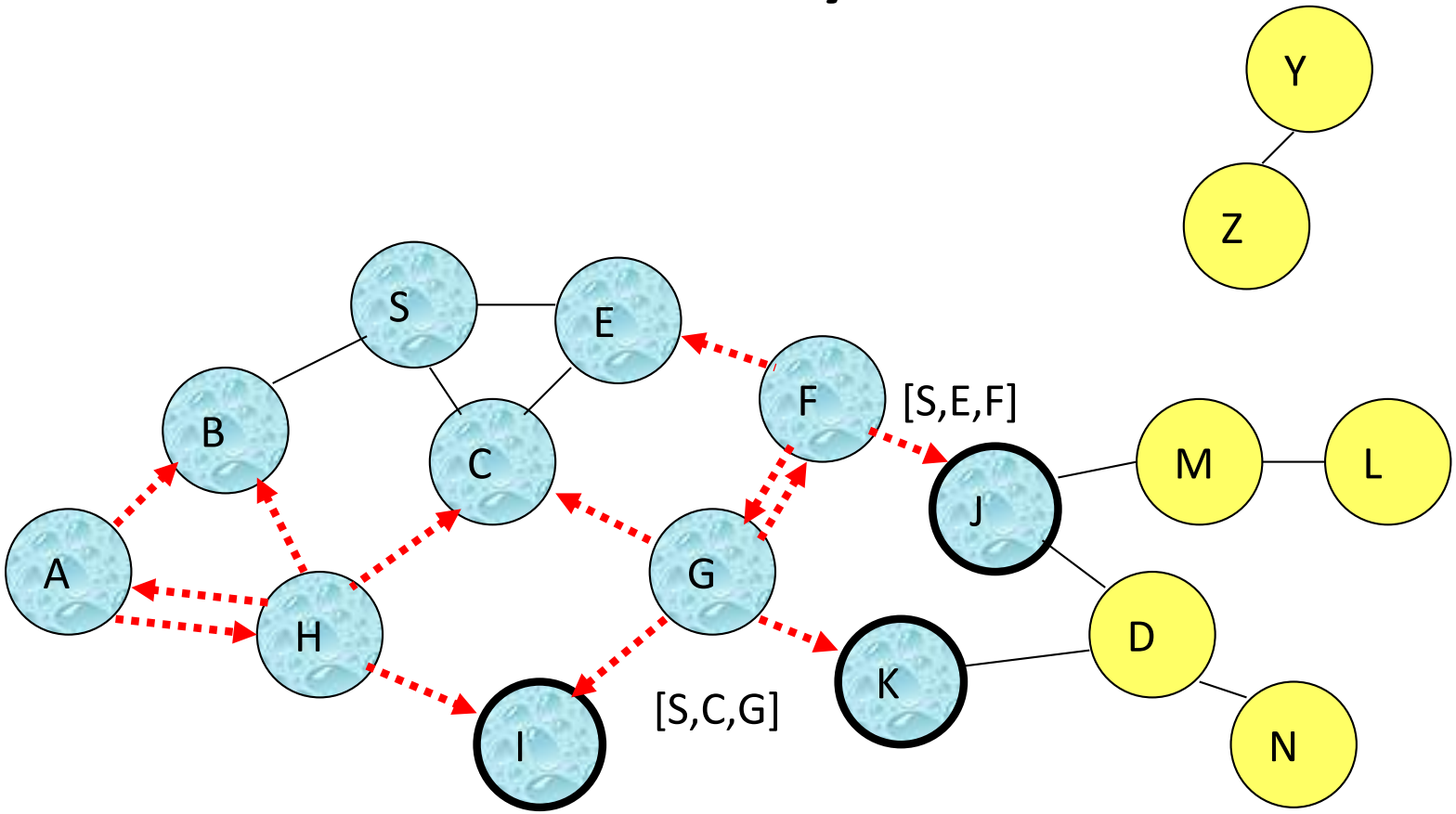
[X,Y] Represents list of identifiers appended to RREQ

Route Discovery in DSR



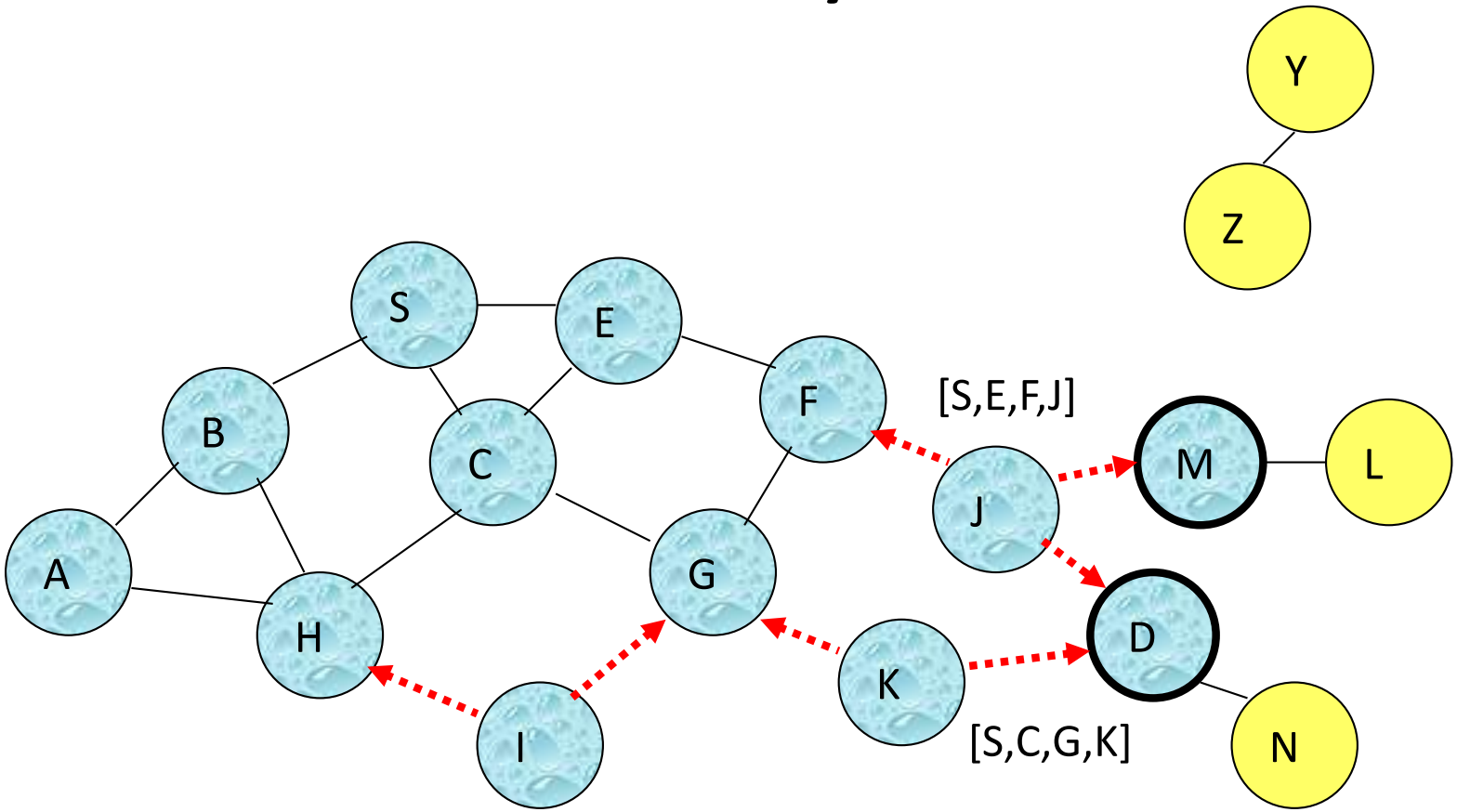
- Node H receives packet RREQ from two neighbors: potential for collision

Route Discovery in DSR



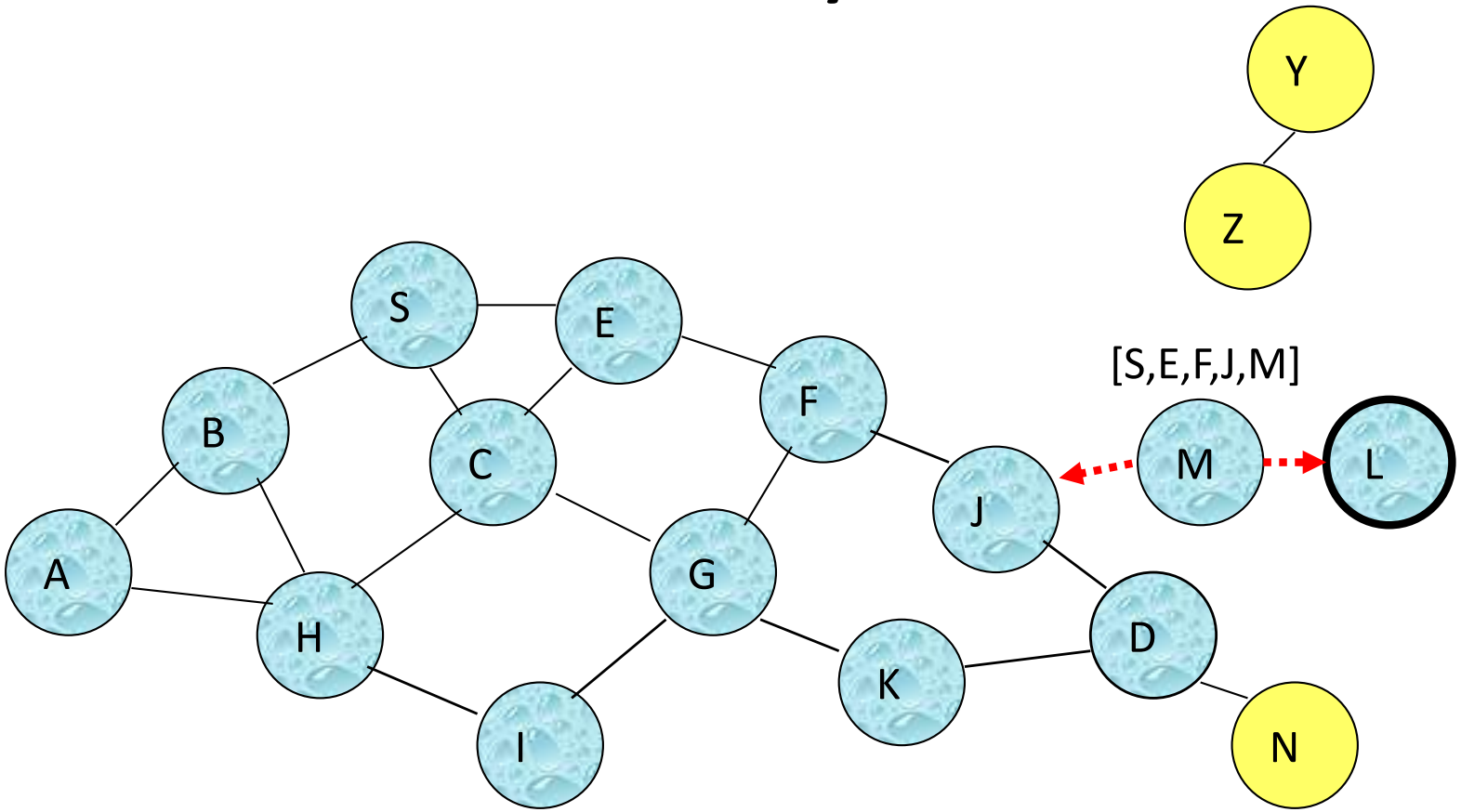
- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

Route Discovery in DSR



- Nodes J and K both broadcast RREQ to node D
- Since nodes J and K are hidden from each other, their transmissions may collide

Route Discovery in DSR

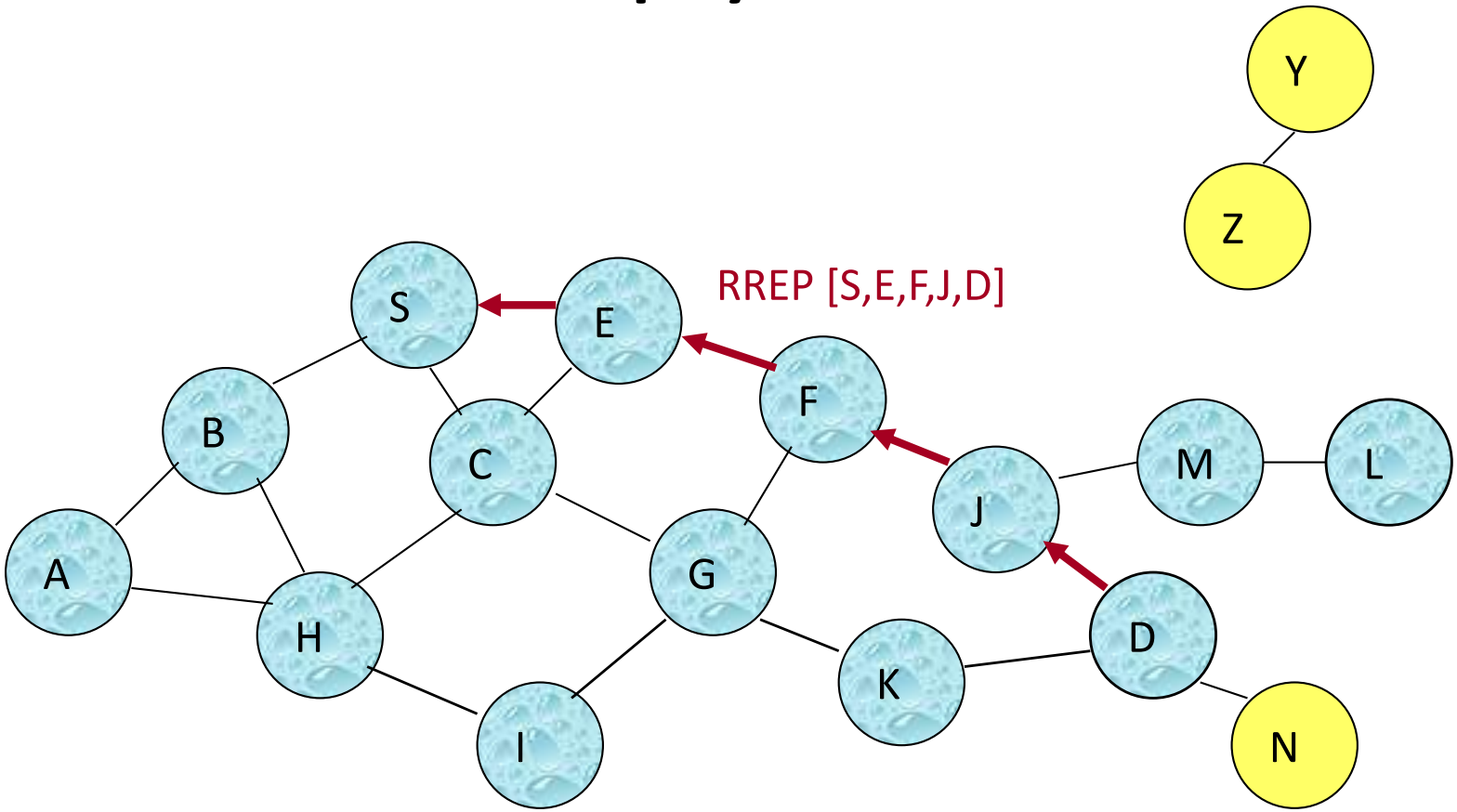


- Node D does not forward RREQ, because node D is the intended target of the route discovery

Route Discovery in DSR

- Destination D on receiving the first RREQ, sends a Route Reply (RREP)
- RREP is sent on a route obtained by reversing the route appended to received RREQ
- RREP includes the route from S to D on which RREQ was received by node D

Route Reply in DSR



Represents RREP control message

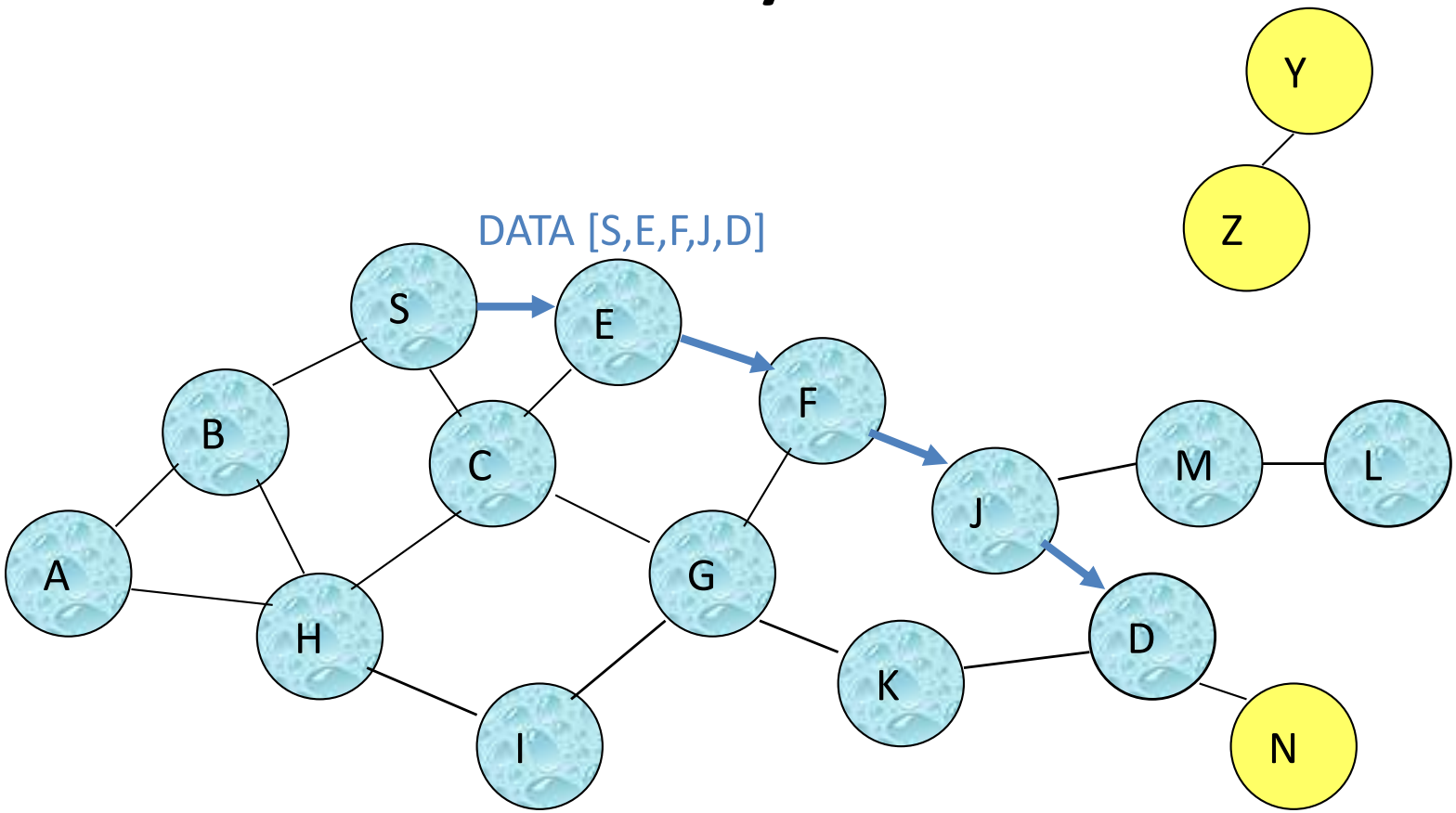
Route Reply in DSR

- Route Reply can be sent by reversing the route in Route Request (RREQ) only if links are guaranteed to be bi-directional
  - To ensure this, RREQ should be forwarded only if it received on a link that is known to be bi-directional
- If unidirectional (asymmetric) links are allowed, then RREP may need a route discovery for S from node D
  - Unless node D already knows a route to node S
  - If a route discovery is initiated by D for a route to S, then the Route Reply is piggybacked on the Route Request from D.
- If IEEE 802.11 MAC is used to send data, then links have to be bi-directional (since Ack is used)

Dynamic Source Routing (DSR)

- Node S on receiving RREP, caches the route included in the RREP
- When node S sends a data packet to D, the entire route is included in the packet header
  - hence the name source routing
- Intermediate nodes use the source route included in a packet to determine to whom a packet should be forwarded

Data Delivery in DSR



Packet header size grows with route length

Dynamic Source Routing: Advantages

- Routes maintained only between nodes who need to communicate
  - reduces overhead of route maintenance
- Route caching can further reduce route discovery overhead
- A single route discovery may yield many routes to the destination, due to intermediate nodes replying from local caches

Dynamic Source Routing: Disadvantages

- Packet header size grows with route length due to source routing
- Flood of route requests may potentially reach all nodes in the network
- Care must be taken to avoid collisions between route requests propagated by neighboring nodes
  - insertion of random delays before forwarding RREQ
- Increased contention if too many route replies come back due to nodes replying using their local cache
  - Route Reply *Storm* problem
  - Reply storm may be eased by preventing a node from sending RREP if it hears another RREP with a shorter route

Dynamic Source Routing: Disadvantages

- An intermediate node may send Route Reply using a stale cached route, thus polluting other caches
- This problem can be eased if some mechanism to purge (potentially) invalid cached routes is incorporated.
- For some proposals for cache invalidation, see [Hu00Mobicom]
  - Static timeouts
  - Adaptive timeouts based on link stability

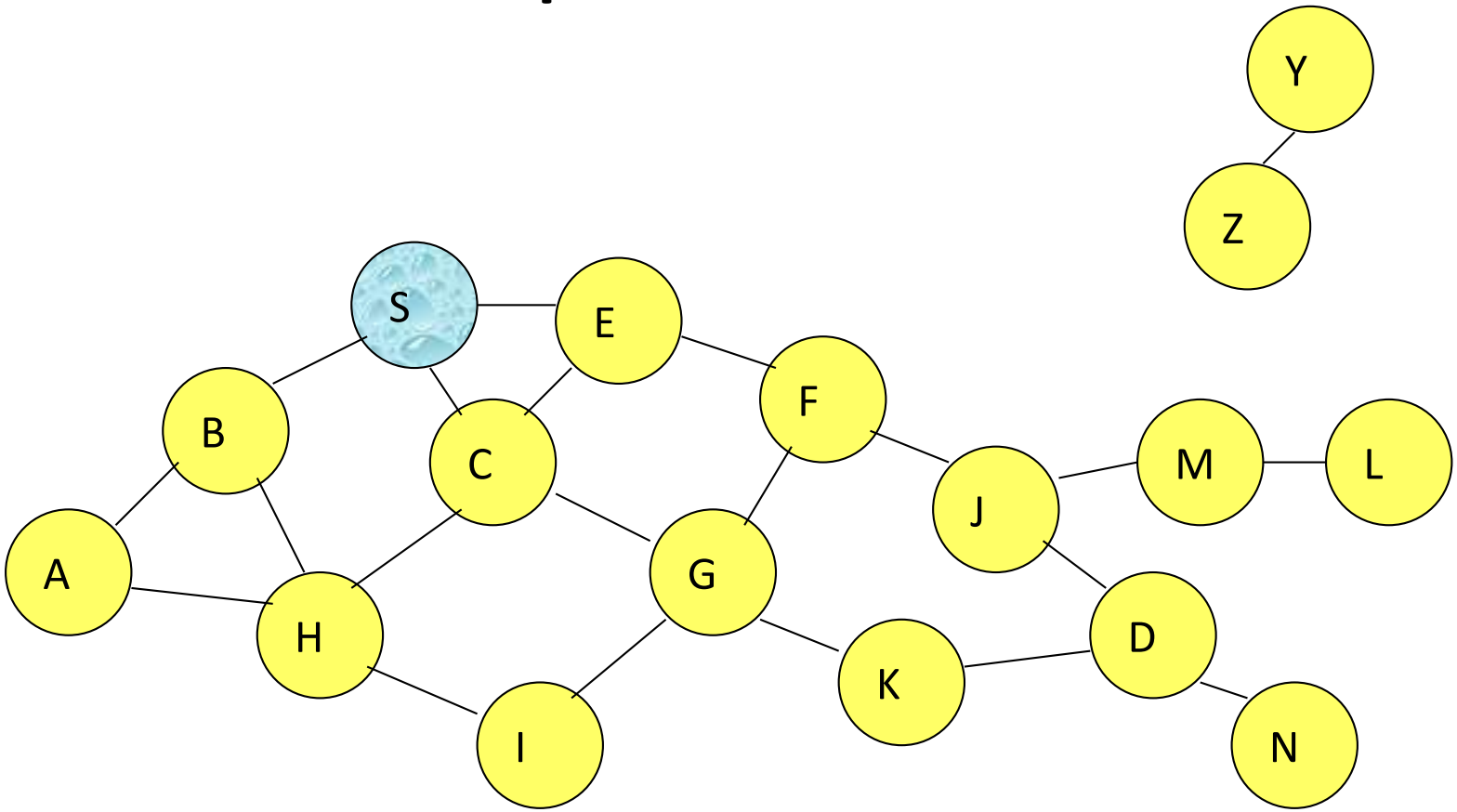
Ad Hoc On-Demand Distance Vector Routing (AODV)

- DSR includes source routes in packet headers
- Resulting large headers can sometimes degrade performance
  - particularly when data contents of a packet are small
- AODV attempts to improve on DSR by maintaining routing tables at the nodes, so that data packets do not have to contain routes
- AODV retains the desirable feature of DSR that routes are maintained only between nodes which need to communicate

AODV

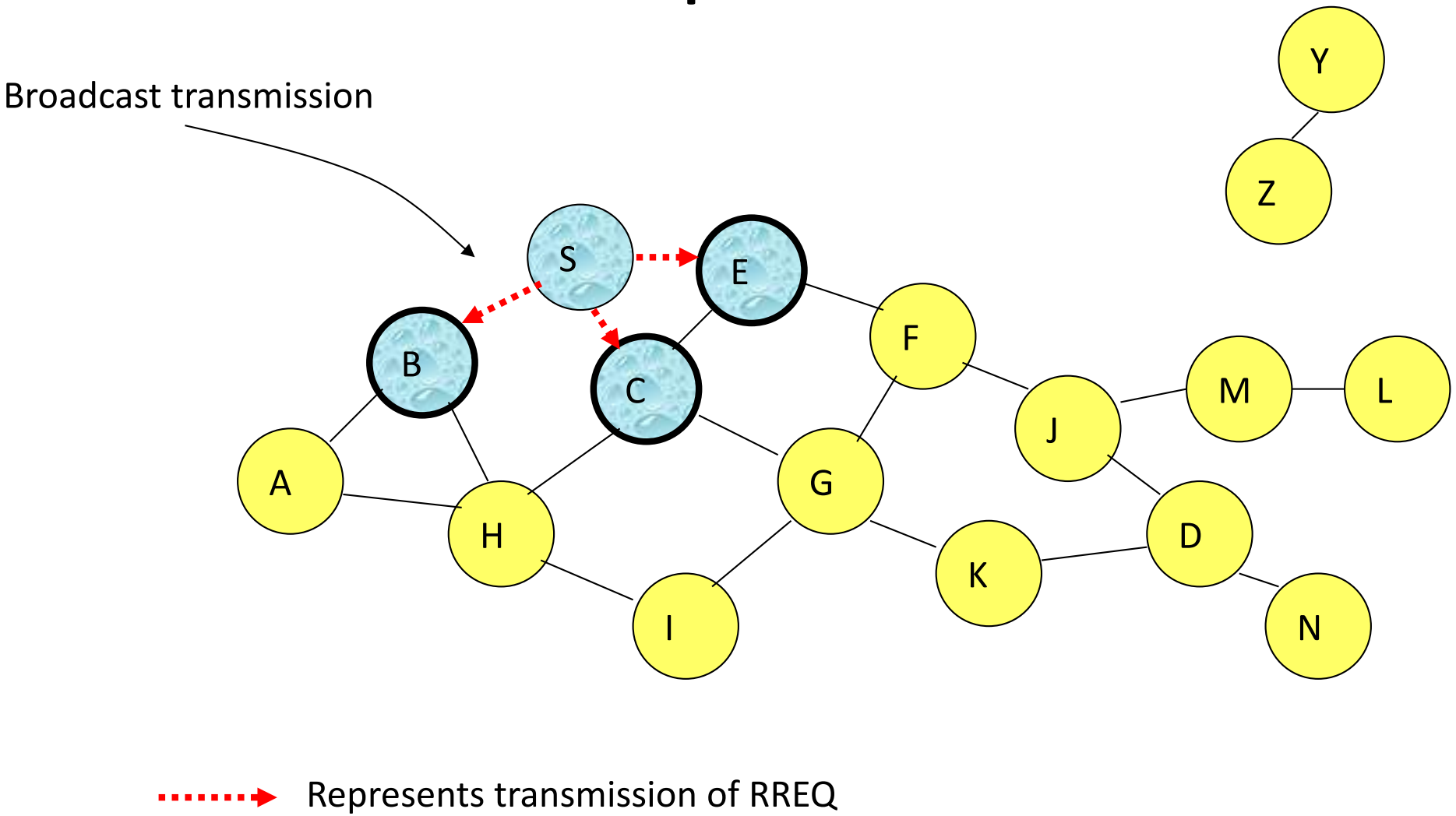
- Route Requests (RREQ) are forwarded in a manner similar to DSR
- When a node re-broadcasts a Route Request, it sets up a reverse path pointing towards the source
  - AODV assumes symmetric (bi-directional) links
- When the intended destination receives a Route Request, it replies by sending a Route Reply
- Route Reply travels along the reverse path set-up when Route Request is forwarded

Route Requests in AODV

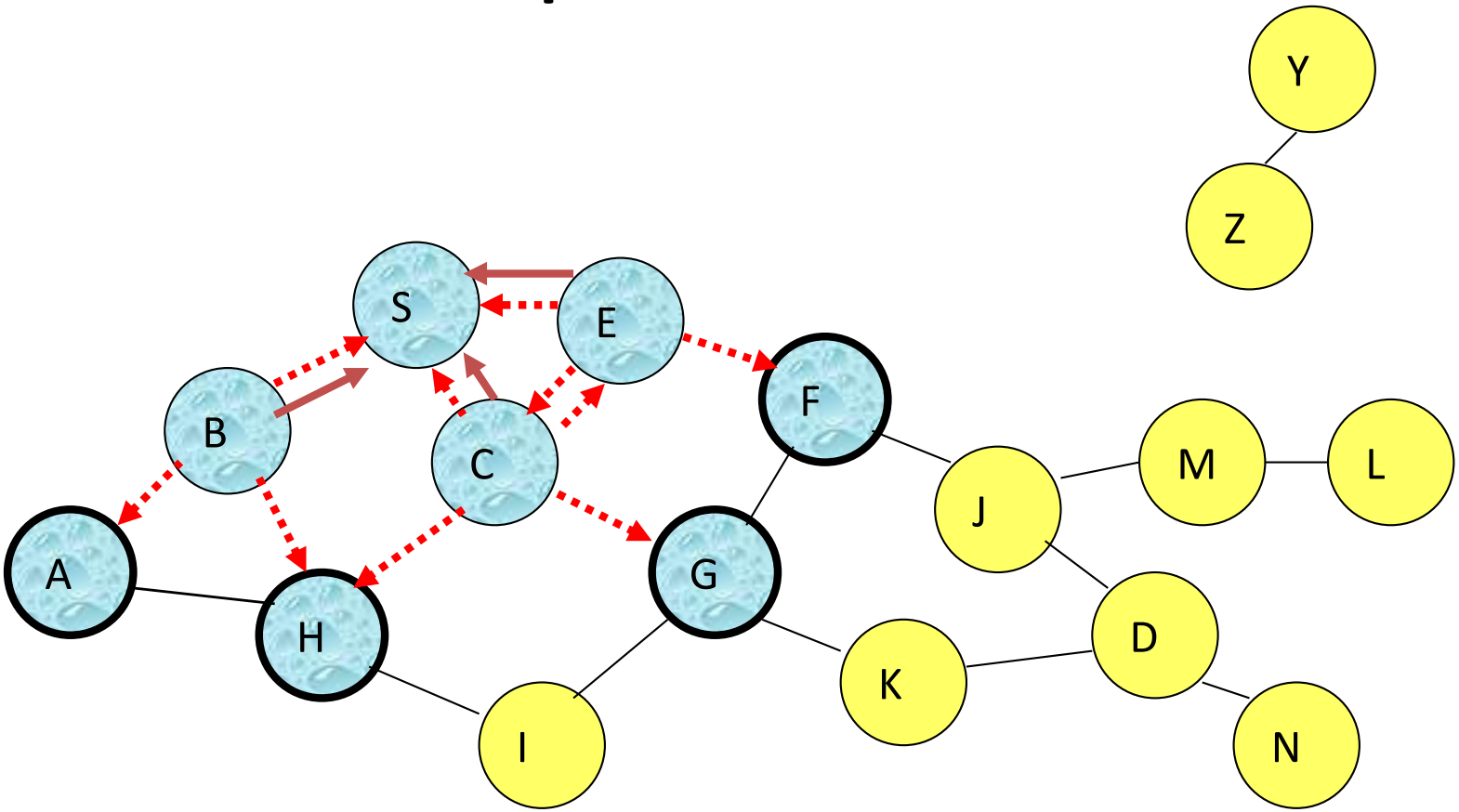


Represents a node that has received RREQ for D from S

Route Requests in AODV

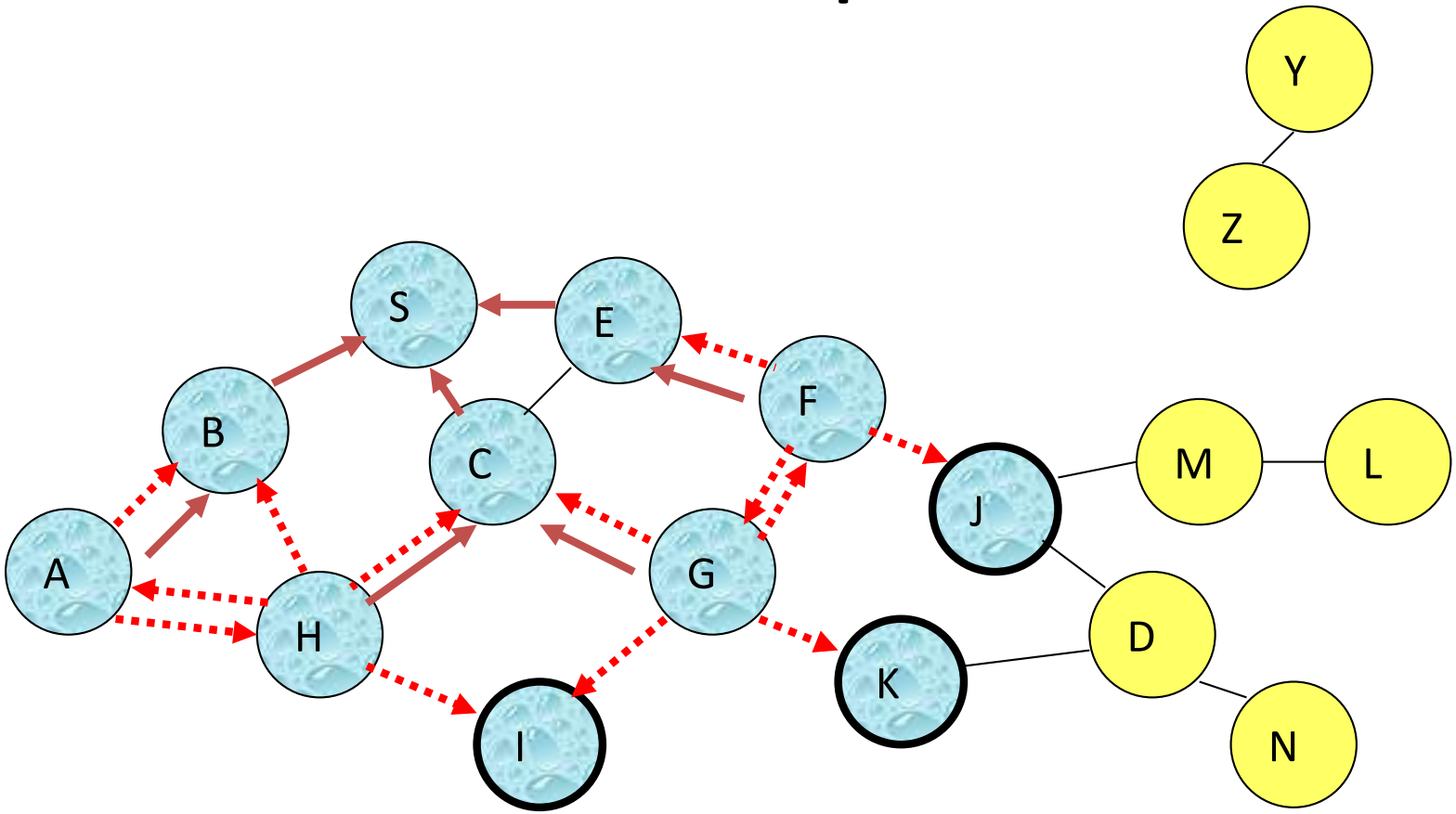


Route Requests in AODV



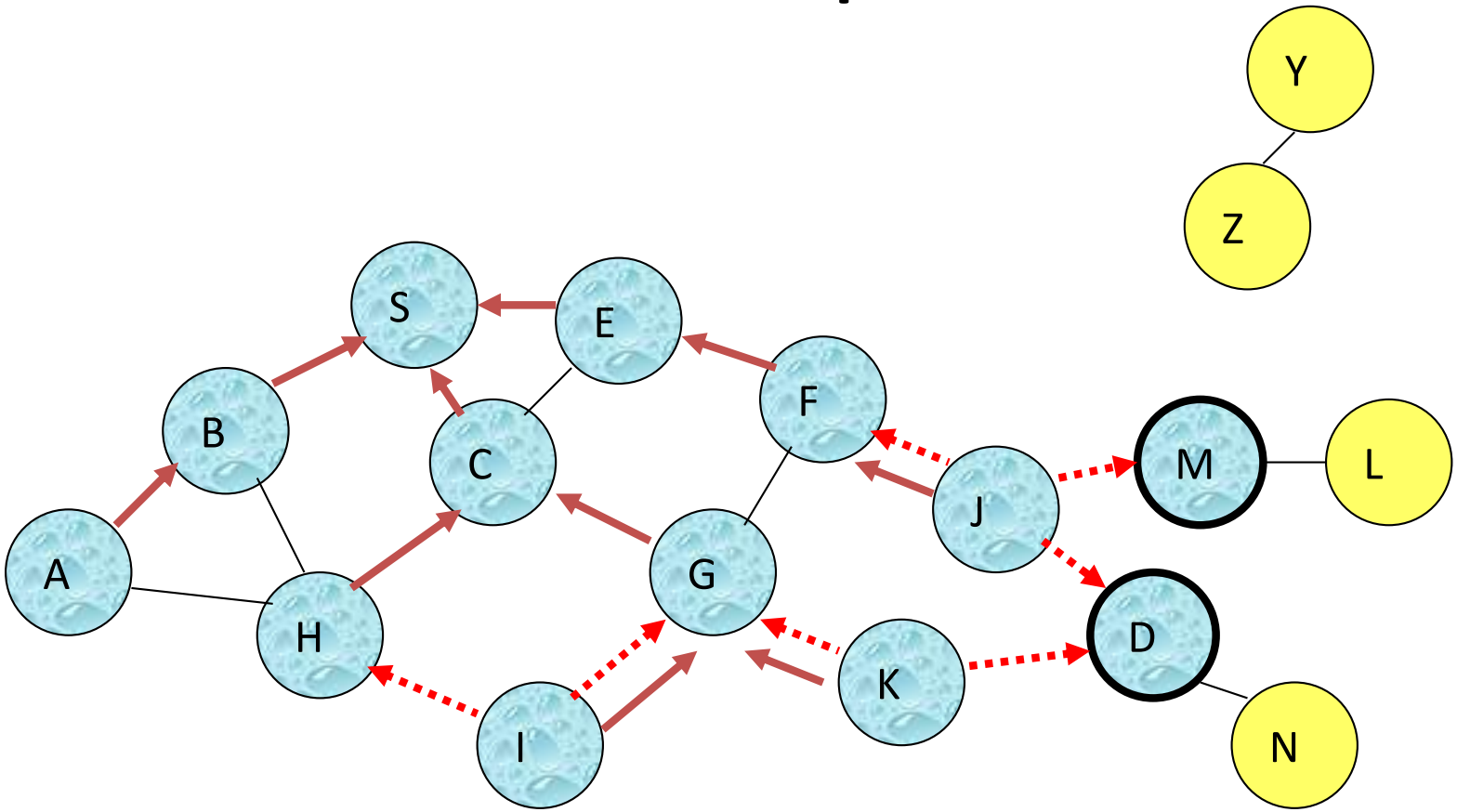
Represents links on Reverse Path

Reverse Path Setup in AODV

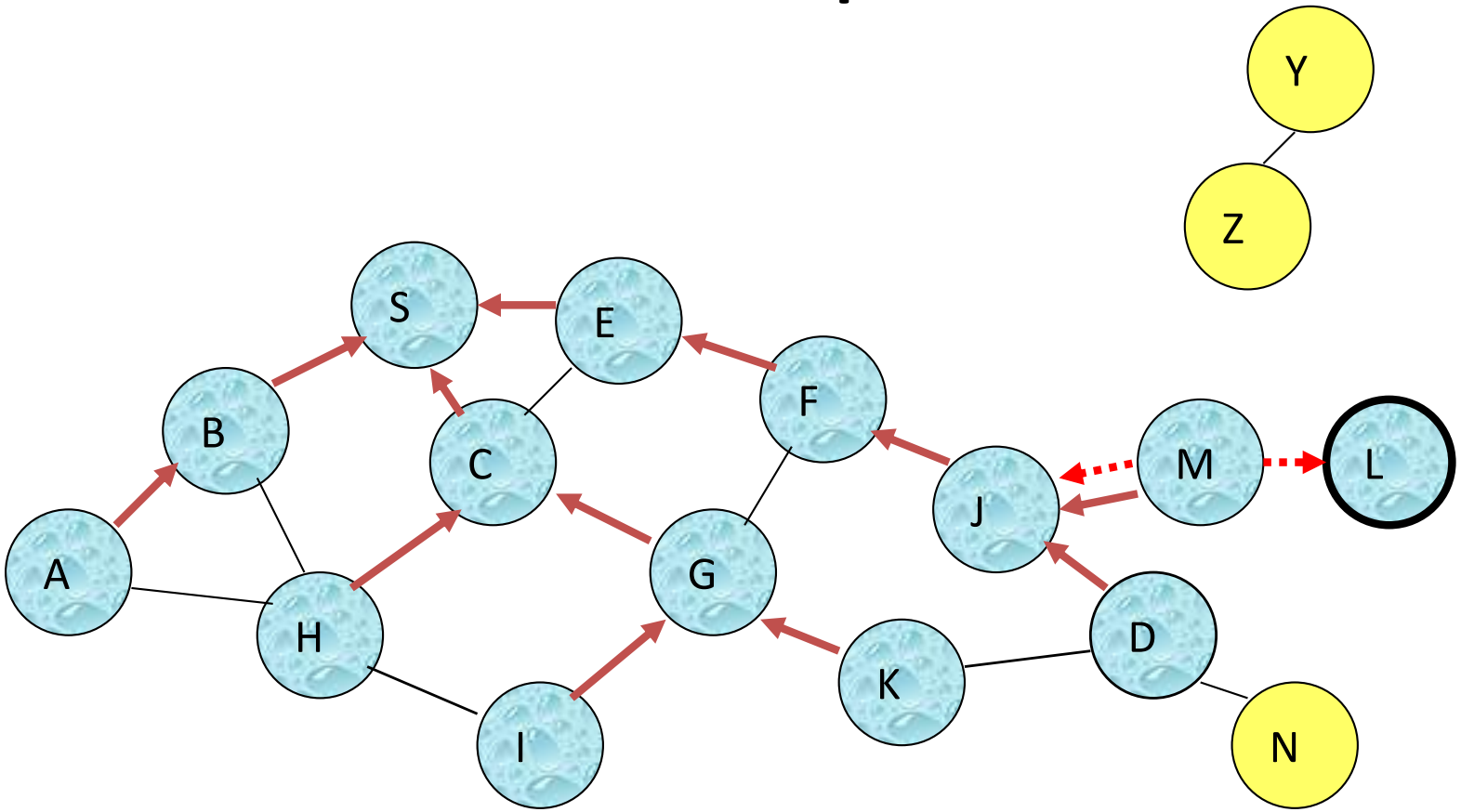


- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

Reverse Path Setup in AODV

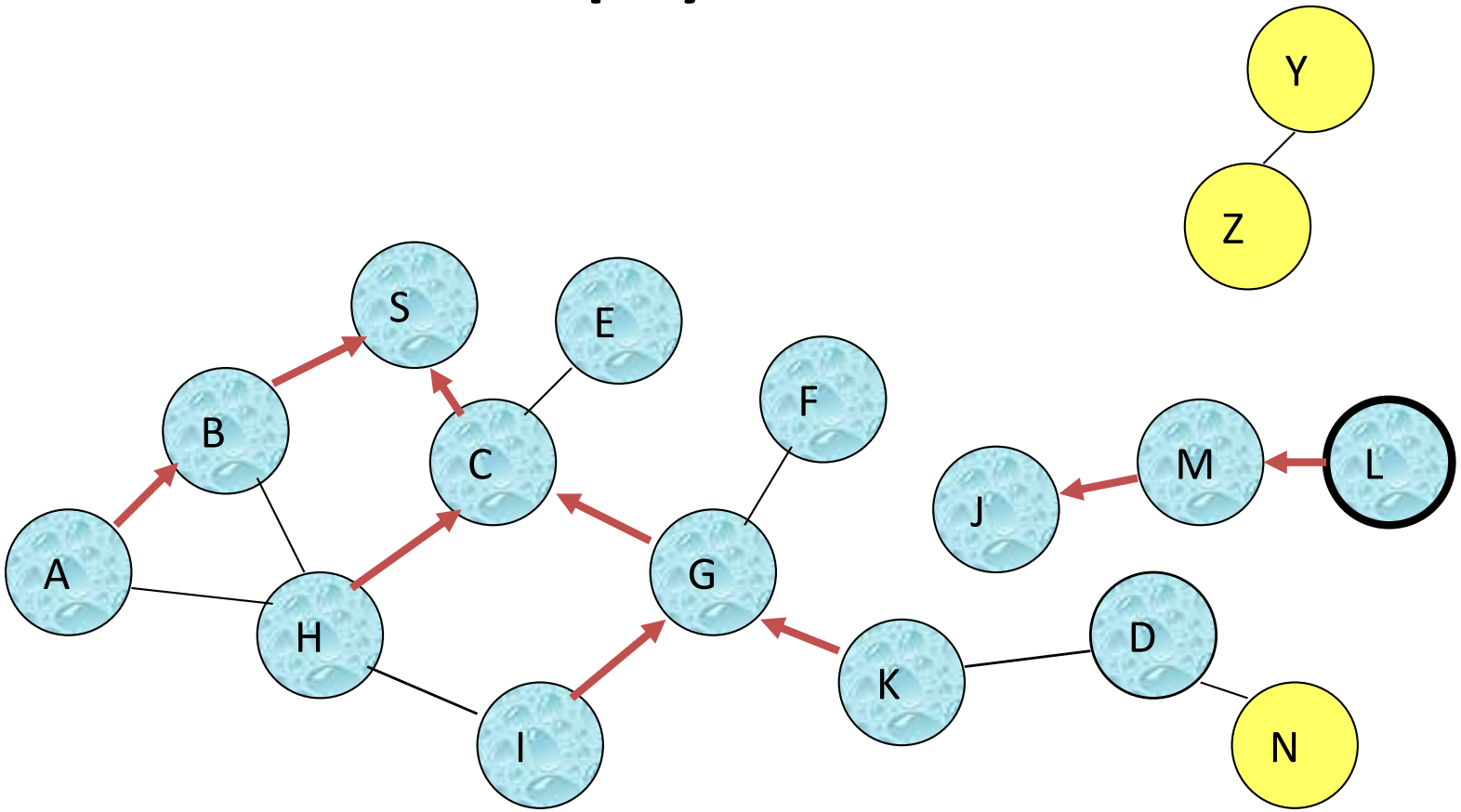


Reverse Path Setup in AODV



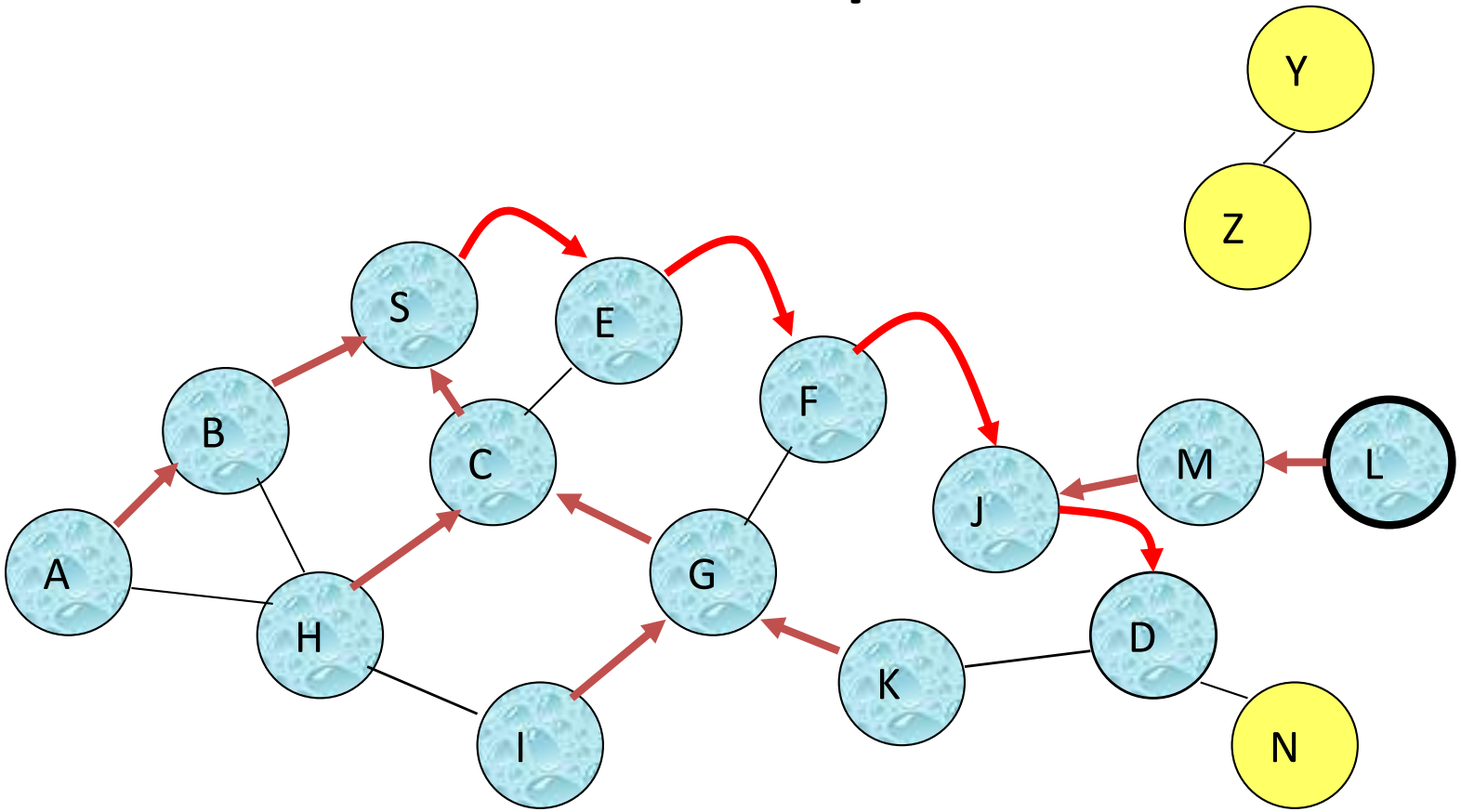
- Node D does not forward RREQ, because node D is the intended target of the RREQ

Route Reply in AODV



Represents links on path taken by RREP

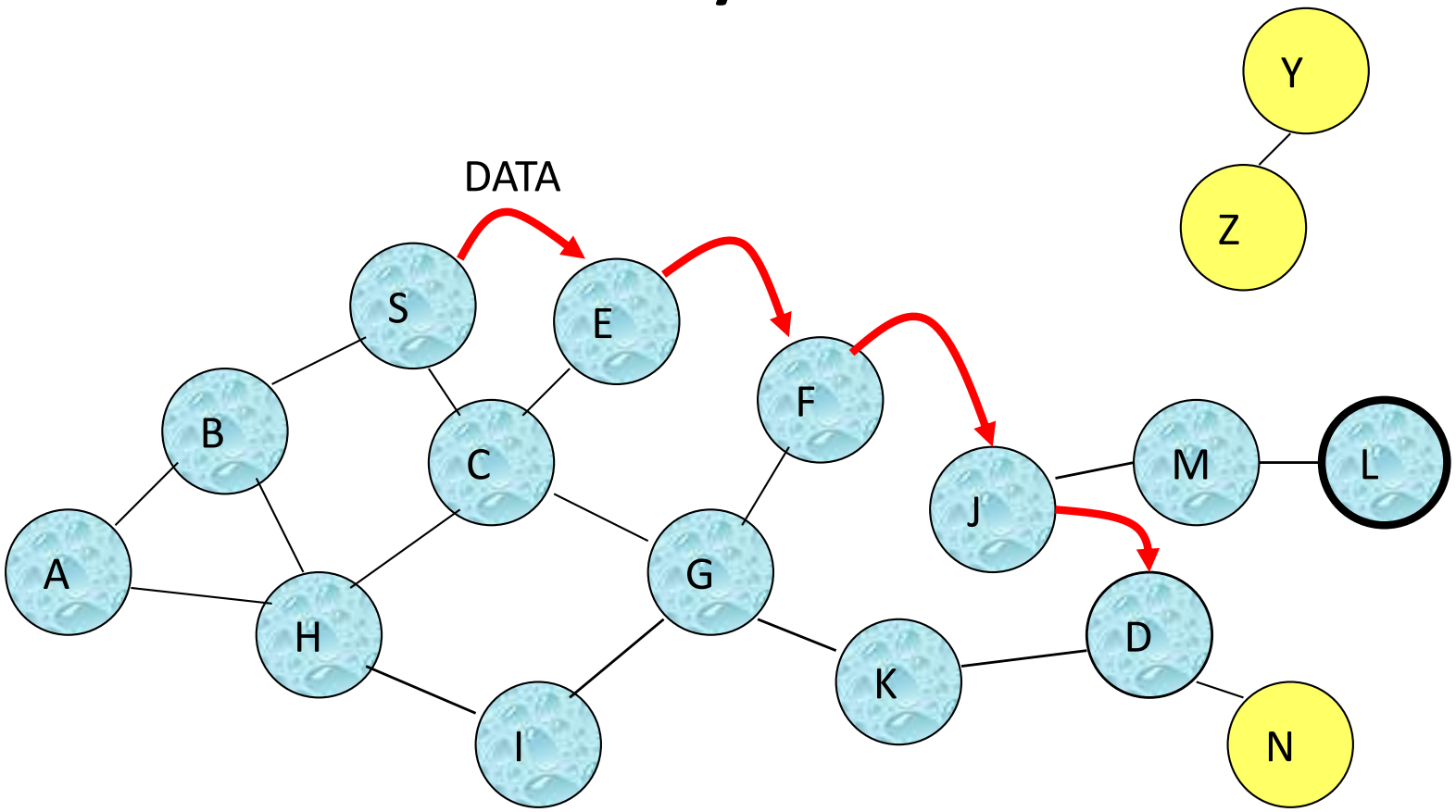
Forward Path Setup in AODV



Forward links are setup when RREP travels along the reverse path

Represents a link on the forward path

Data Delivery in AODV



Routing table entries used to forward data packet.

Route is *not* included in packet header.

Timeouts

- A routing table entry maintaining a reverse path is purged after a timeout interval
  - timeout should be long enough to allow RREP to come back
- A routing table entry maintaining a forward path is purged if *not used* for a *active_route_timeout* interval
  - if no data is being sent using a particular routing table entry, that entry will be deleted from the routing table

Link Failure Reporting

- A neighbor of node X is considered active for a routing table entry if the neighbor sent a packet within *active_route_timeout* interval which was forwarded using that entry
- When the next hop link in a routing table entry breaks, all active neighbors are informed
- Link failures are propagated by means of Route Error messages

Route Error

- When node X is unable to forward packet P (from node S to node D) on link (X,Y), it generates a RERR message

- When node S receives the RERR, it initiates a new route discovery for D

Link Failure Detection

- *Hello* messages: Neighboring nodes periodically exchange hello message
- Absence of hello message is used as an indication of link failure
- Alternatively, failure to receive several MAC-level acknowledgement may be used as an indication of link failure

Optimization: Expanding Ring Search

- Route Requests are initially sent with small Time-to-Live (TTL) field, to limit their propagation
  - DSR also includes a similar optimization
- If no Route Reply is received, then larger TTL tried

Summary: AODV

- Routes need not be included in packet headers
- Nodes maintain routing tables containing entries only for routes that are in active use
- At most one next-hop per destination maintained at each node
  - Multi-path extensions can be designed
  - DSR may maintain several routes for a single destination
- Unused routes expire even if topology does not change

Proactive Protocols

- The schemes discussed so far are reactive
- Proactive schemes based on distance-vector and link-state mechanisms have also been proposed

Link State Routing

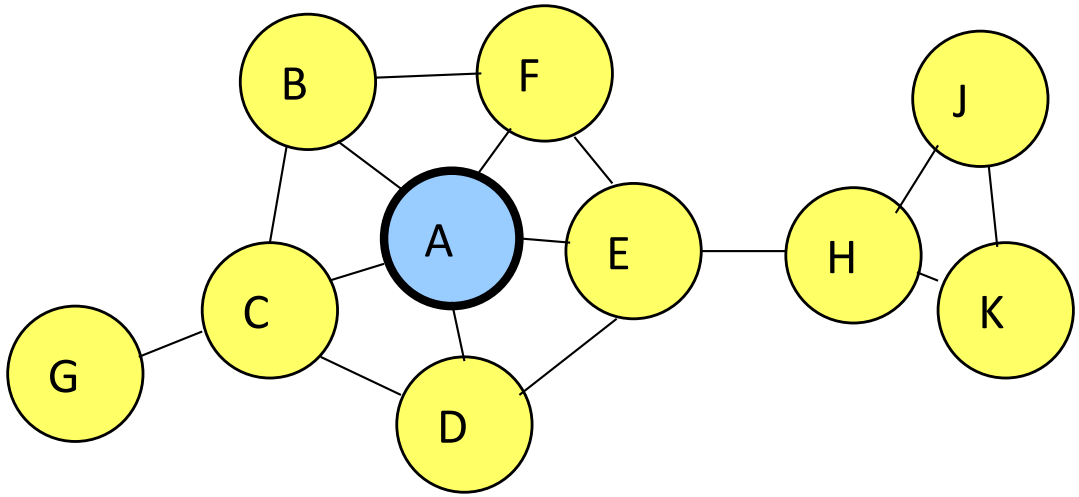
- Each node periodically floods status of its links
- Each node re-broadcasts link state information received from its neighbor
- Each node keeps track of link state information received from other nodes
- Each node uses above information to determine next hop to each destination

Optimized Link State Routing (OLSR) [Jacquet, Inria]

- The overhead of flooding link state information is reduced by requiring fewer nodes to forward the information
- A broadcast from node X is only forwarded by its *multipoint relays*
- Multipoint relays of node X are its neighbors such that each two-hop neighbor of X is a one-hop neighbor of at least one multipoint relay of X

Optimized Link State Routing (OLSR)

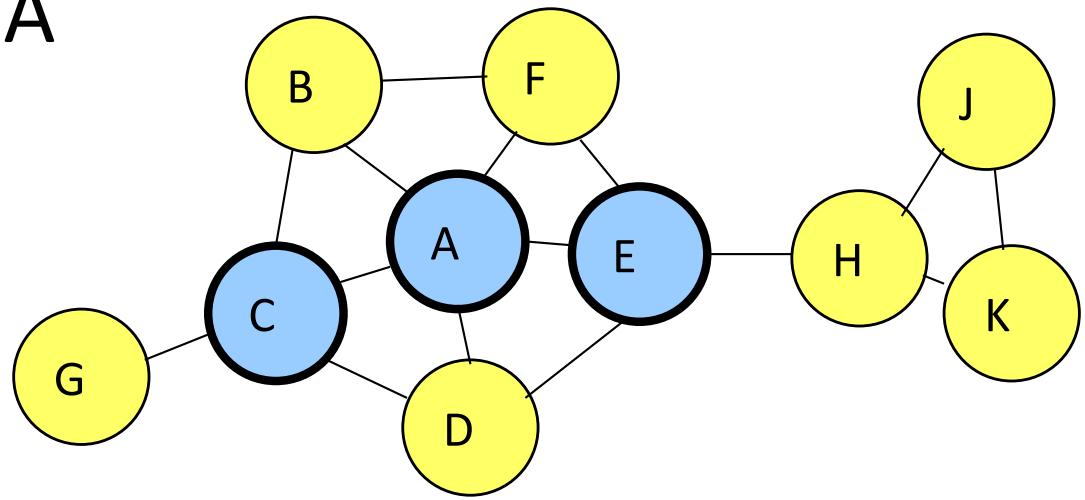
- Nodes C and E are multipoint relays of node A



Node that has broadcast state information from A

Optimized Link State Routing (OLSR)

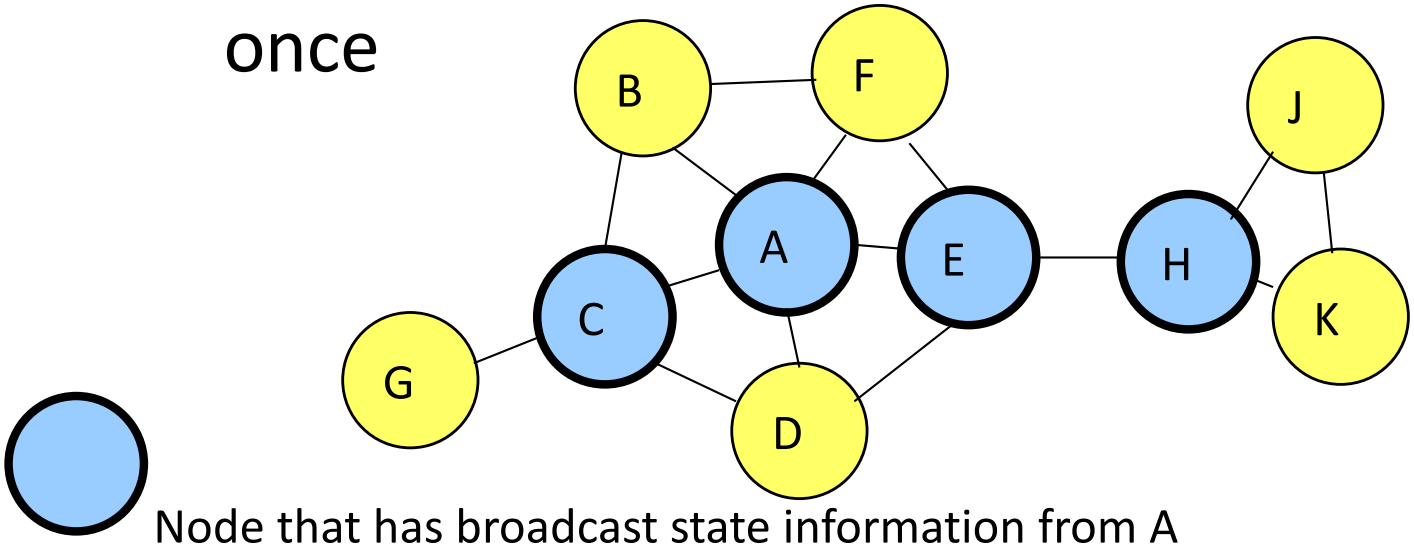
- Nodes C and E forward information received from A



Node that has broadcast state information from A

Optimized Link State Routing (OLSR)

- Nodes E and K are multipoint relays for node H
- Node K forwards information received from H
  - E has already forwarded the same information once



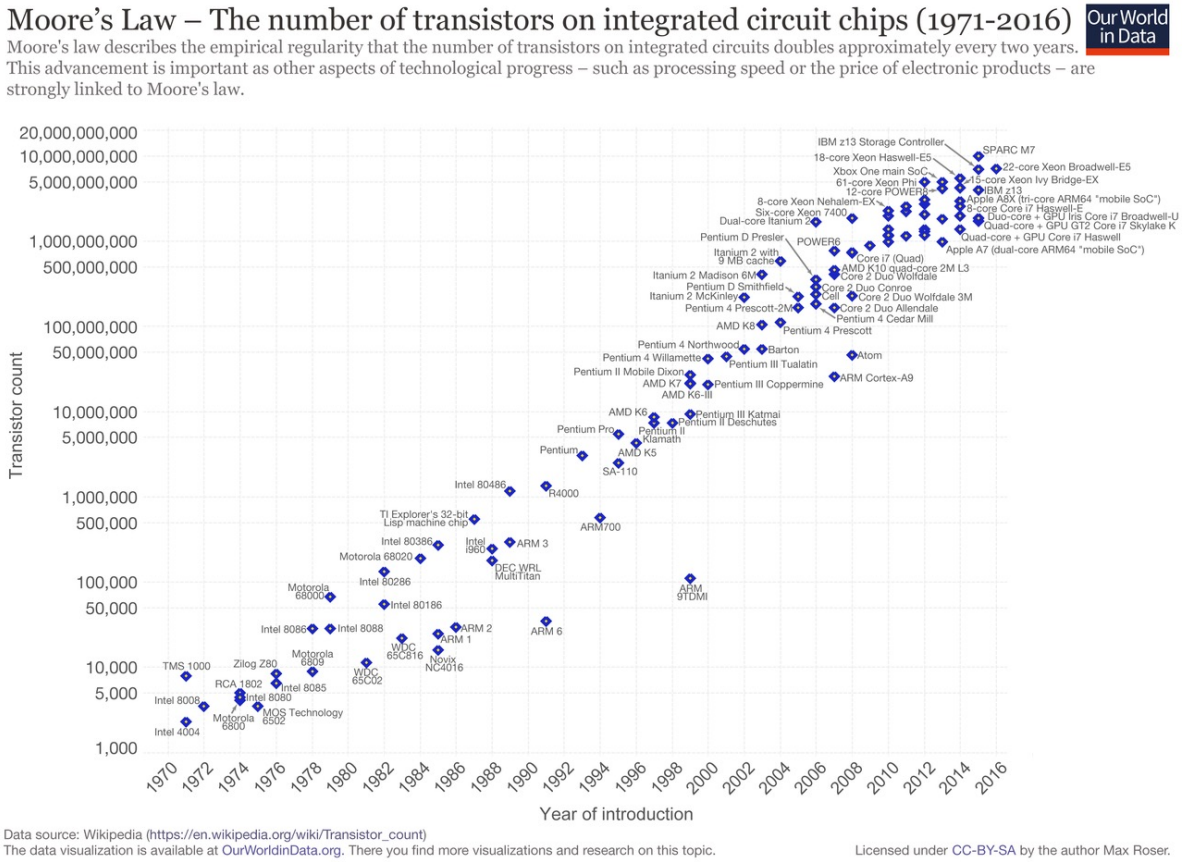
OLSR Summary

- OLSR floods information through the multipoint relays
- The flooded information itself is for links connecting nodes to respective multipoint relays
- Routes used by OLSR only include multipoint relays as intermediate nodes

IoT: What is it?

- Wikipedia: *The Internet of things (IoT) is the network of devices, vehicles, and home appliances that contain electronics, software, actuators, and connectivity which allows these things to connect, interact and exchange data*
- Google*: the interconnection via the Internet of computing devices embedded in everyday objects, enabling them to send and receive data*
- ITU*: a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies*
- Cisco *Internet Business Solutions Group (IBSG): IoT is simply the point in time when more "things or objects" were connected to the Internet than people*

Why now?


Moore's Law – The number of transistors on integrated circuit chips (1971-2016)
Our World in Data
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.
Transistor count
Year of introduction
Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.
Licensed under CC-BY-SA by the author Max Roser.

Why now?

- A given computing capacity becomes exponentially smaller and cheaper with each passing year
  - Build radios and exceptionally small mechanical structures that sense fields and forces in the physical world
- These inexpensive, low-power communication devices can be deployed throughout a physical space
  - Sensing close to physical phenomena, processing and communicating this information, and coordinating actions with other nodes.
- Combining these capabilities with the “Internet” makes it possible to measure the world in “real time”

A bit of history

- The term IoT was coined in by Kevin Ashton in 1999 during his work at Procter&Gamble.
  - Working in supply chain optimization, wanted to attract senior management's attention to a new exciting technology called RFID.
  - Because the internet was the hottest new trend in 1999 he called his presentation "Internet of Things"
- But the history of connected things is far older

A bit of history – It started with telemetry

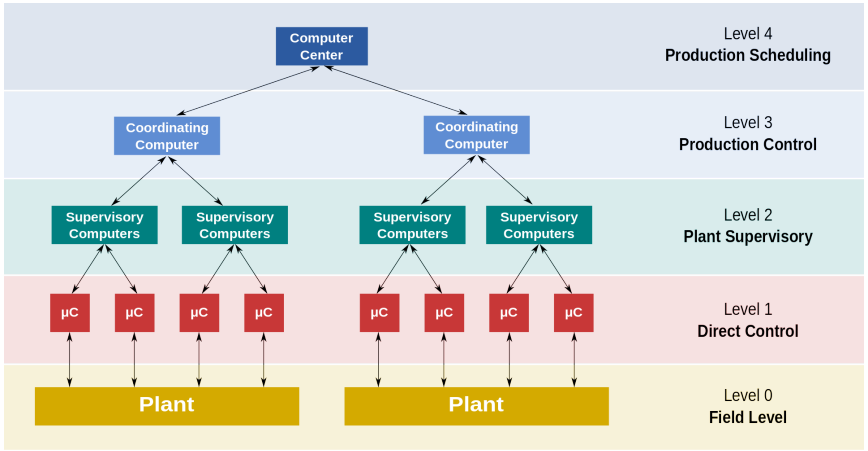
- Wired telemetry
  - 1874, French engineers built a system of weather and snow-depth sensors on Mont Blanc that transmitted real-time information to Paris
  - In 1912, Commonwealth Edison developed a system of telemetry to monitor electrical loads on its power grid
- Wireless telemetry
  - *Radiosonde*: weather sensors carried into atmosphere using balloon and transmitting data by radio to a ground receiver
  - The first true radiosonde by Robert Bureau in 1929

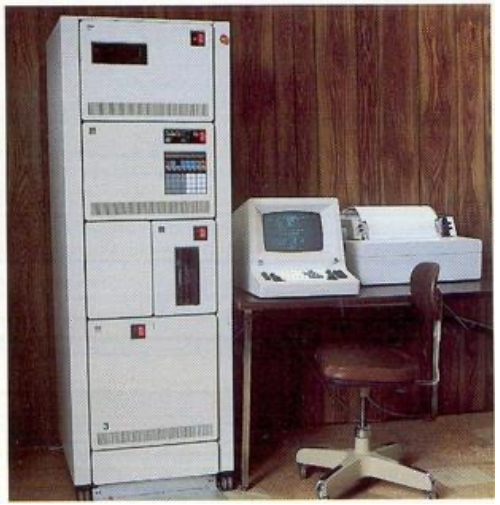
A bit of history – Then came (wired) M2M

1970's: Caller ID – A transmitter sending its number to the receiver

Automatic meter reading ("smart grid")

1960's: First generation of supervisory control and data acquisition (SCADA) systems





A bit of history – Then came (cellular) M2M

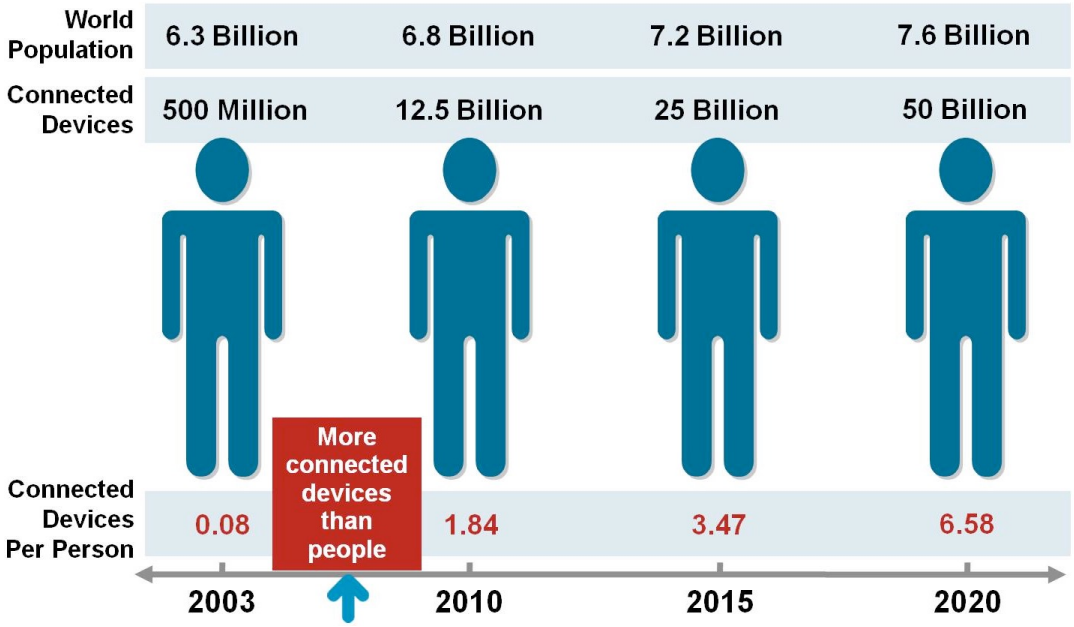
- 1995: Siemens launched a GSM data module called "M1" based on the Siemens mobile phone S6 for M2M industrial applications, enabling machines to communicate over wireless networks
  - Remote monitoring and tracking
  - Cash registers
  - Vehicle telematics

M2M vs. IoT

- Both involve machines communicating
- M2M point-to-point communication, usually industrial context, not necessarily IP
- IoT: all objects, including your jacket, connected via IP

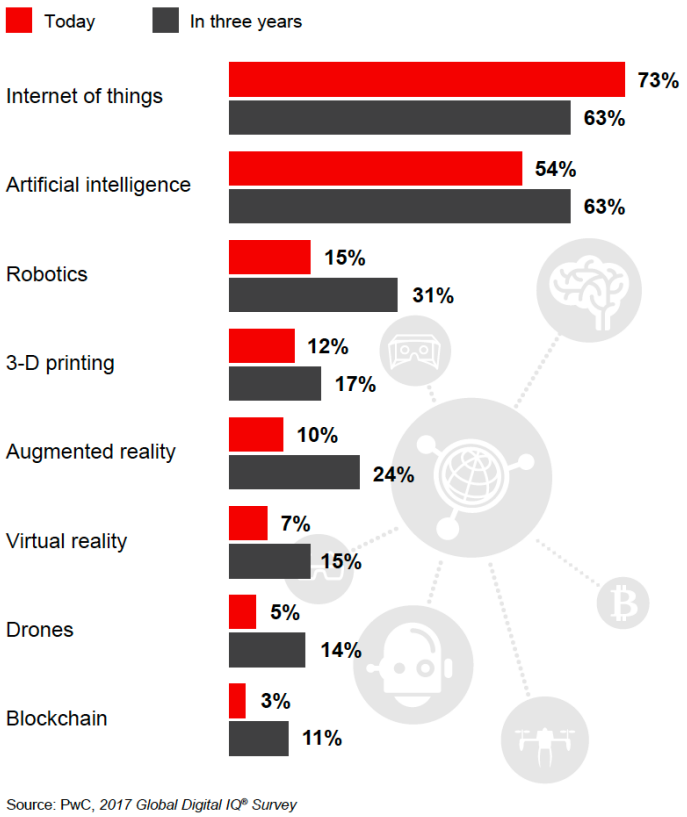
IoT today and tomorrow

- When was IoT born?
  - Cisco *IBSG: IoT is simply the point in time when more "things or objects" were connected to the Internet than people*
  - *Circa 2008-9*
- *Eventually trillion of sensors connected to the internet*



IoT Tomorrow

- What technology are companies investing in?



Source: PwC, *2017 Global Digital IQ® Survey*

IoT Applications

SMART THERMOSTATS

nest

Save resources and money on your heating bills by adapting to your usage patterns and turning the temperature down when you're away from home.

CONNECTED CARS

CAR 2GO

Tracked and rented using a smartphone. Car2Go also handles billing, parking and insurance automatically.

ACTIVITY TRACKERS

BASIS

Continuously capture heart rate patterns, activity levels, calorie expenditure and skin temperature on your wrist 24/7.

SMART OUTLETS

belkin

Remotely turn any device or appliance on or off. Track a device's energy usage and receive personalized notifications from your smartphone.

PARKING SENSORS

STREETLINE
CONNECTING THE REAL WORLD

Using embedded street sensors, users can identify real-time availability of parking spaces on their phone. City officials can manage and price their resources based on actual use.

Nest Labs

- It all seems like a toy – until Google pays $3.2 billions for it in 2014
- Nest Learning Thermostat
  - Can learn people's schedule, at which temperature they are used to and when
  - Input from built-in sensors and smartphones using WiFi
  - Adapt heating accordingly using a machine learning algorithms

Smart cow

Source: *The Economist*, 2010.

- Sparked, Dutch start-up company, implants sensors in the ears of cattle.
- This allows farmers to monitor cows' health and track their movements
  - Healthier, more plentiful supply of meat
- On average, each cow generates about 200 megabytes of information a year

IoT and Digital Health

- The global internet of things (IoT) in healthcare market size is expected to reach USD 534.3 billion by 2025 (Grand View Research)
- Medical devices
  - Biosensors (measure glucose levels, arterial pressure, heart rate, oxygen level, pulse, etc)
- System and software
  - Remote device management
  - Network bandwidth management
  - Data analytics
- Services
  - System integration services
  - Consulting, training, and education

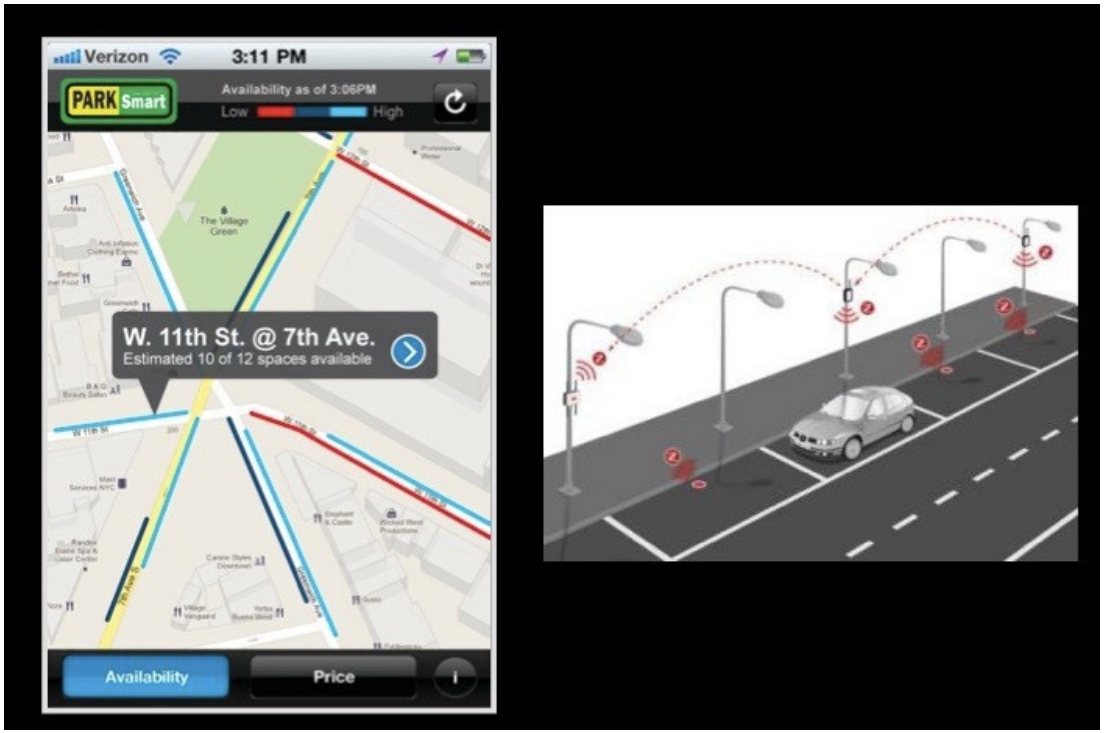
Better Quality of Life for the Elderly

- The world's population is aging: approximately 1 billion people age 65 and older by 2050
- Small, wearable devices can detect a person's vital signs
- Send an alert to a healthcare professional when a certain threshold has been reached
- Sense when a person has fallen down and can't get up.

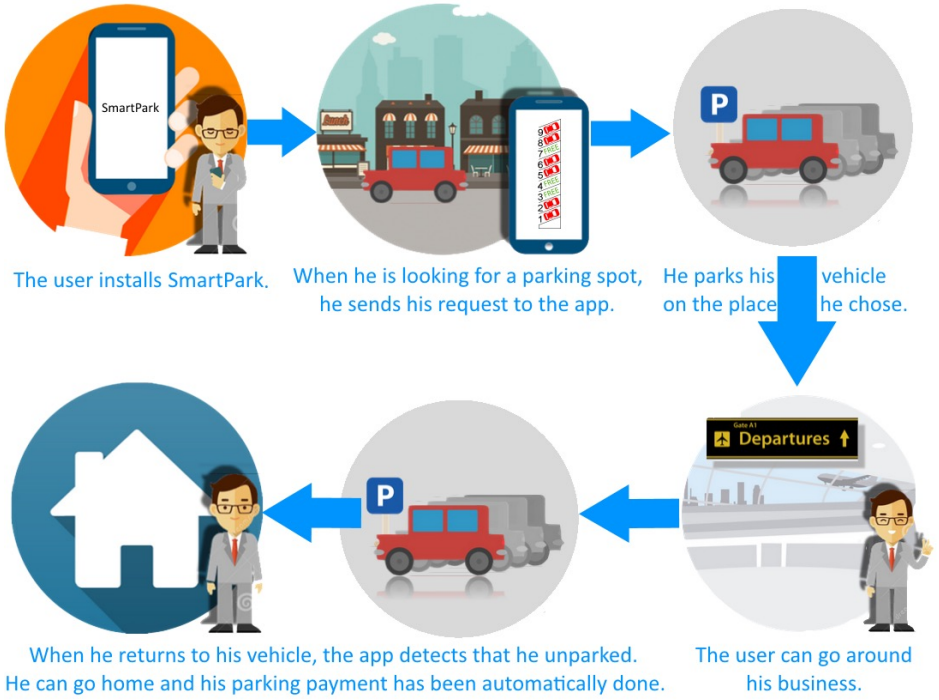
Transportation and smart cities

- More than 20 minutes in NYC for more than 40% drivers to find a free lot
- 30% of congested urban areas traffic
- $12 million unpaid parking tickets in Cincinnati (2005)

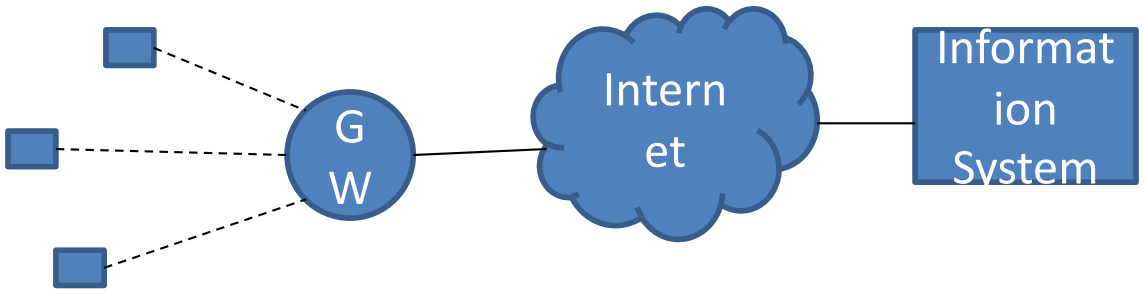
Transportation and smart cities



Transportation and smart cities



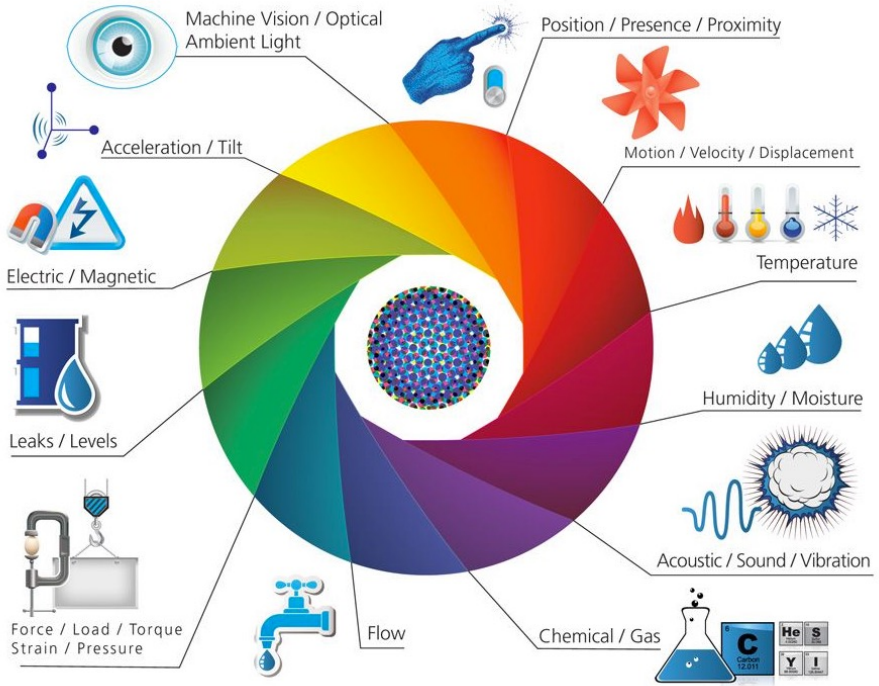
Basic IoT Architecture



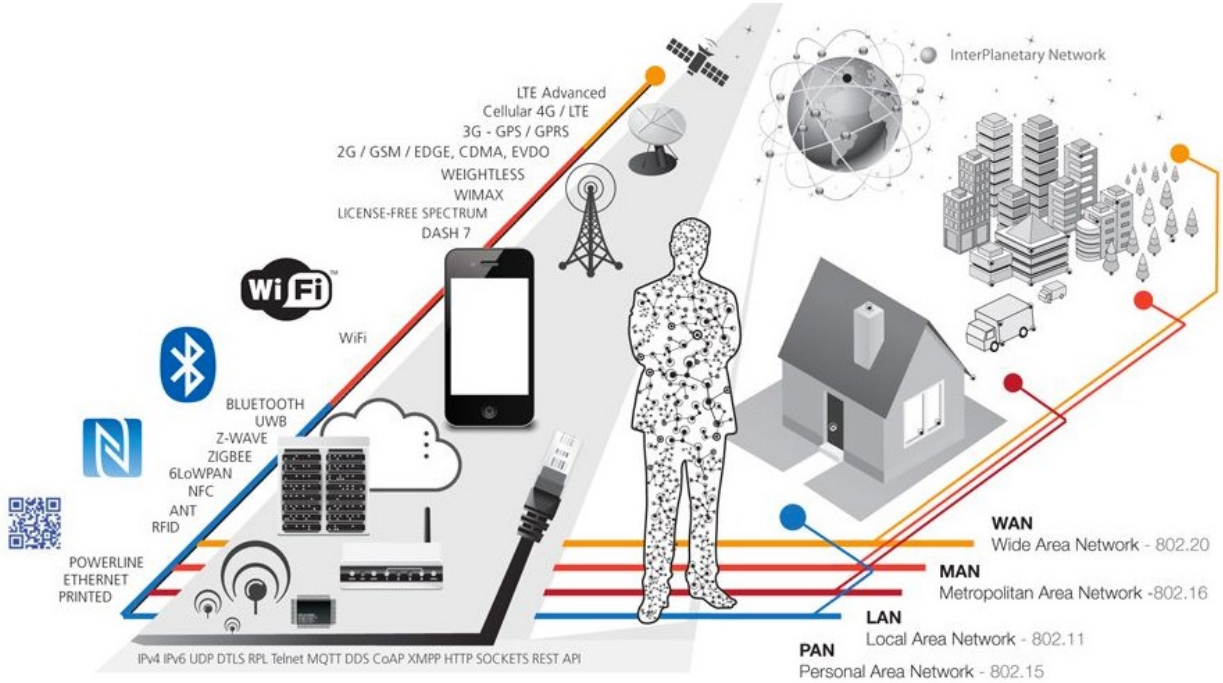
Fundamental Characteristics

- Heterogeneity
  - Sensors
  - Networking
    - Range, capacity, power consumptions, infrastructure)
  - Applications
    - Rail network maintenance vs “smart” home lightening system
- Scale
  - In number of devices rather than data
  - Billions of devices expected (IPV6 becomes mandatory)

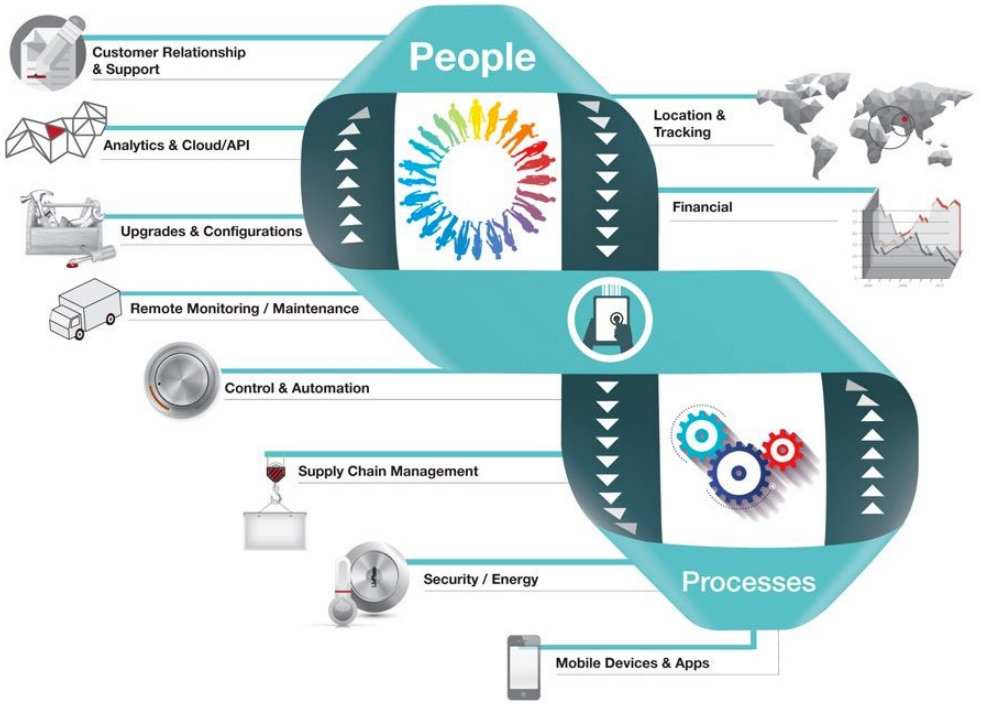
Sensors


Machine Vision / Optical
Ambient Light
Position / Presence / Proximity
Acceleration / Tilt
Motion / Velocity / Displacement
Electric / Magnetic
Temperature
Leaks / Levels
Humidity / Moisture
Acoustic / Sound / Vibration
Force / Load / Torque
Strain / Pressure
Flow
Chemical / Gas

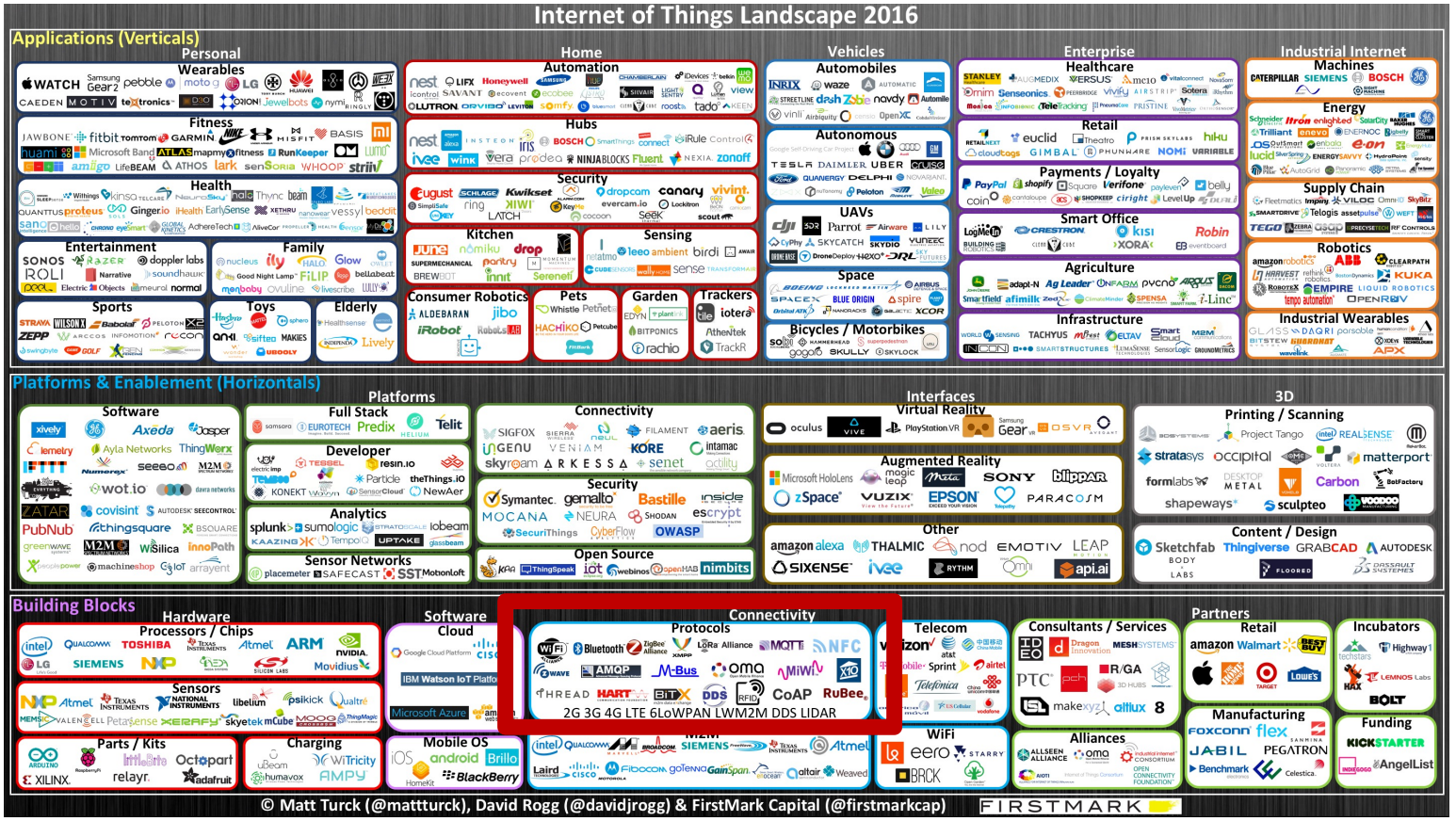
Networking


InterPlanetary Network
LTE Advanced
Cellular 4G / LTE
3G - GPS / GPRS
2G / GSM / EDGE, CDMA, EVDO
WEIGHTLESS
WIMAX
LICENSE-FREE SPECTRUM
DASH 7
WiFi
BLUETOOTH
UWB
Z-WAVE
ZIGBEE
6LoWPAN
NFC
ANT
RFID
POWERLINE
ETHERNET
PRINTED
IPv4 IPv6 UDP DTLS RPL Telnet MQTT DDS CoAP XMPP HTTP SOCKETS REST API
WAN
Wide Area Network - 802.20
MAN
Metropolitan Area Network -802.16
LAN
Local Area Network - 802.11
PAN
Personal Area Network - 802.15

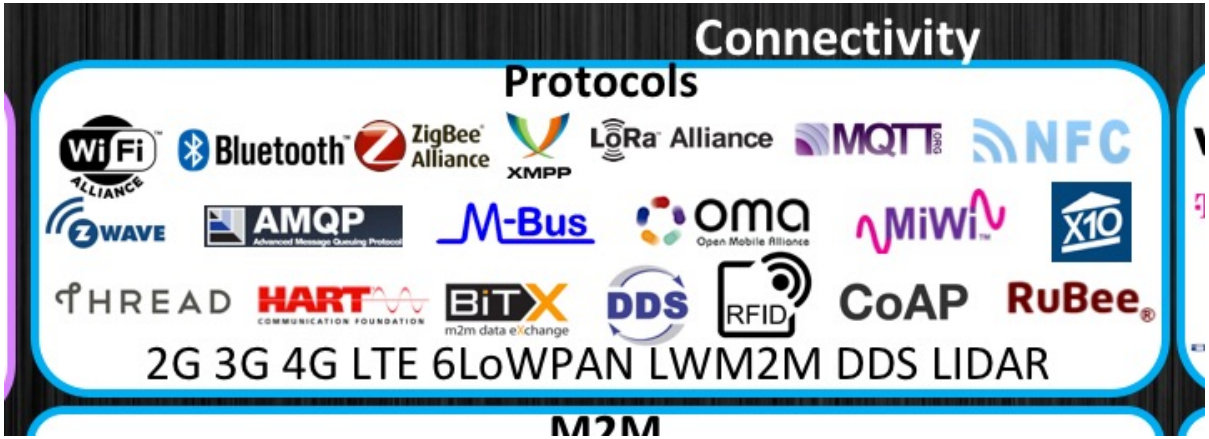
Services


People
Customer Relationship & Support
Analytics & Cloud/API
Upgrades & Configurations
Remote Monitoring / Maintenance
Location & Tracking
Financial
Control & Automation
Supply Chain Management
Security / Energy
Processes
Mobile Devices & Apps

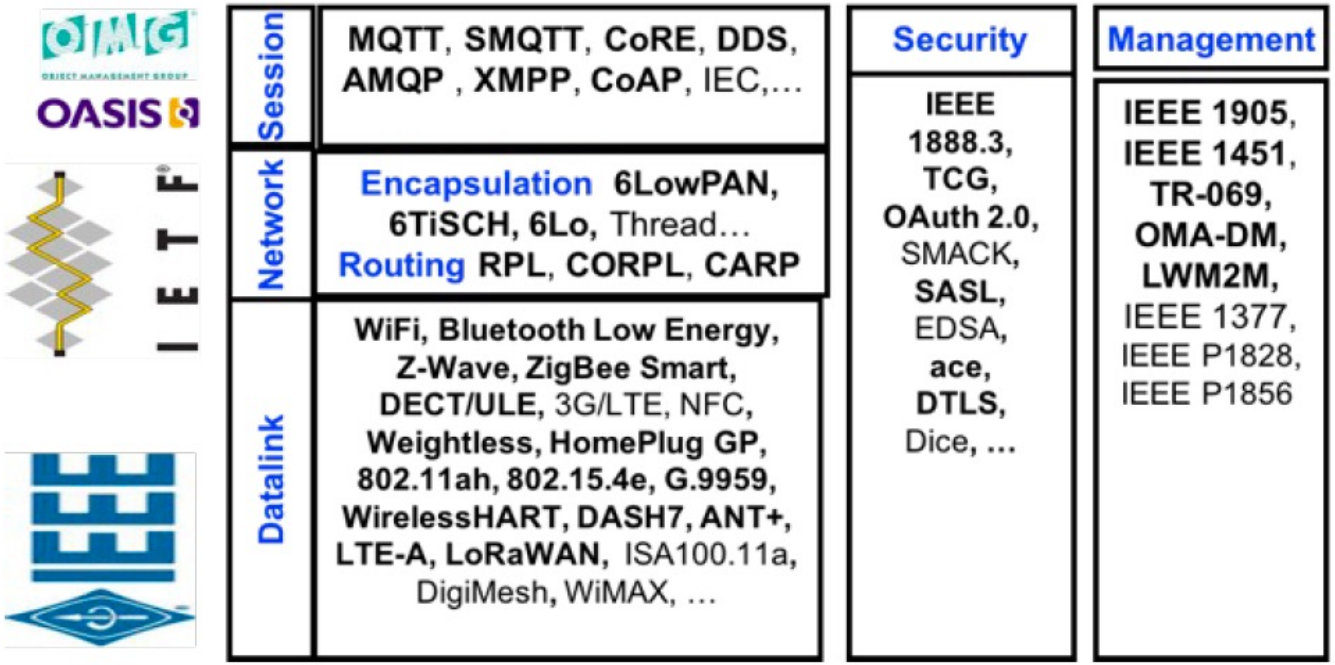
IoT Landscape



IoT Landscape



IoT Interconnection




Bluetooth

Bluetooth History

- Created in 1994 by Ericsson
  - Named after tenth-century king Harald I of Denmark and parts of Norway who united dissonant Danish tribes into a single kingdom
- **Goal: Low-power, low-cost, short-range cable replacement**
  - Range of 0-10m
  - Low data rates (19.2-100 kbps)
- Feb 1998: Ericsson, Nokia, IBM, Toshiba, Intel formed a special interest group (SIG) to focus on the development of such solutions
- Dec 1999: Specification (v1.0b) was released

IEEE 802.15 - WPAN

- Started in 1997 as a sub-group of IEEE 802.11
- Task Group 1- Based on Bluetooth
  - PHY and MAC layer design for wirelessly connecting devices entering a *personal operating space* (POS)
  - POS is a 10m space around a person who is stationary or in motion
- Task Group 2
  - Coexistence of WLANs and WPANs
  - Interoperability between a WLAN and WPAN device

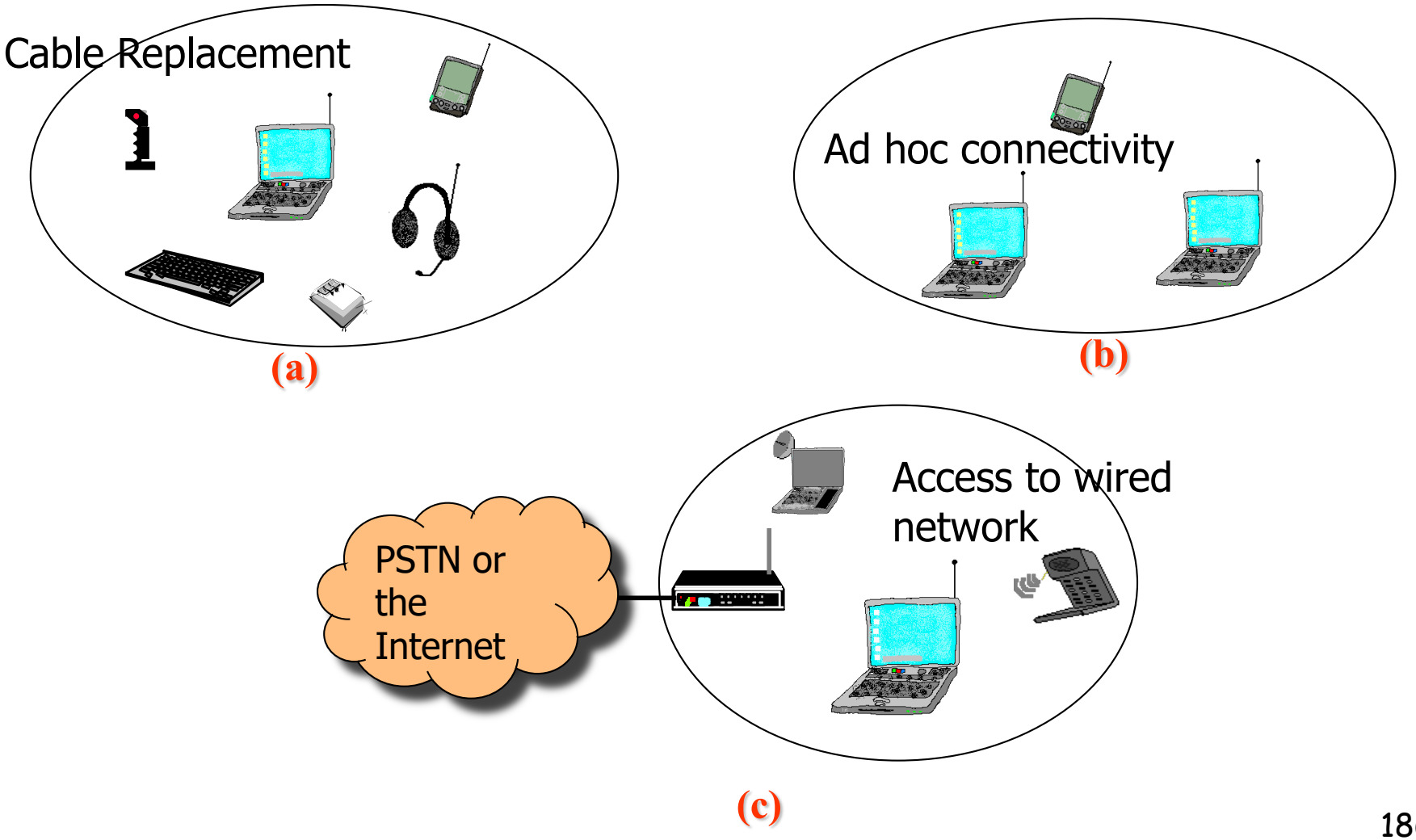
IEEE 802.15 - WPAN

- Task Group 3
  - Higher data rates (up to 20 Mbps)
  - Motivated by Kodak, Cisco, Motorola
  - Multimedia applications like digital imaging and video
  - Support for UWB
- Task Group 4
  - Low data rates and ultra low power/complexity devices for sensor networking
  - Home automation, smart tags, interactive toys, location tracking, etc.

Bluetooth Standard

- Specifies the complete system from the radio level up to the application level
- Protocol stack is partly in hardware and partly in software running on a microprocessor

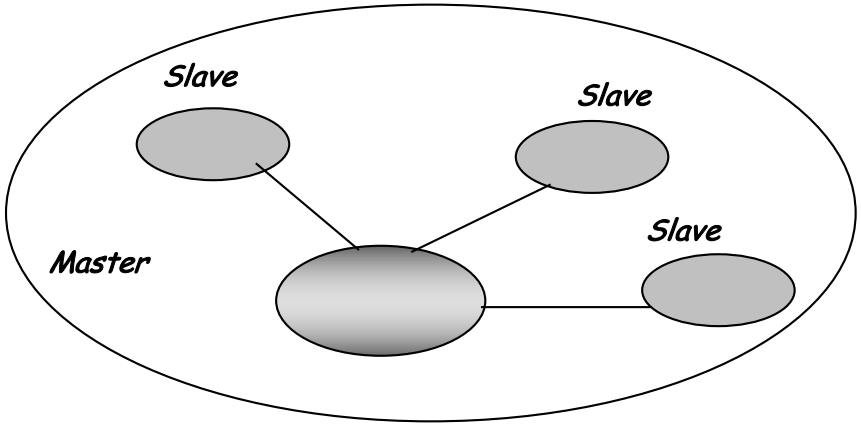
Applications of Bluetooth



Some basics of Bluetooth

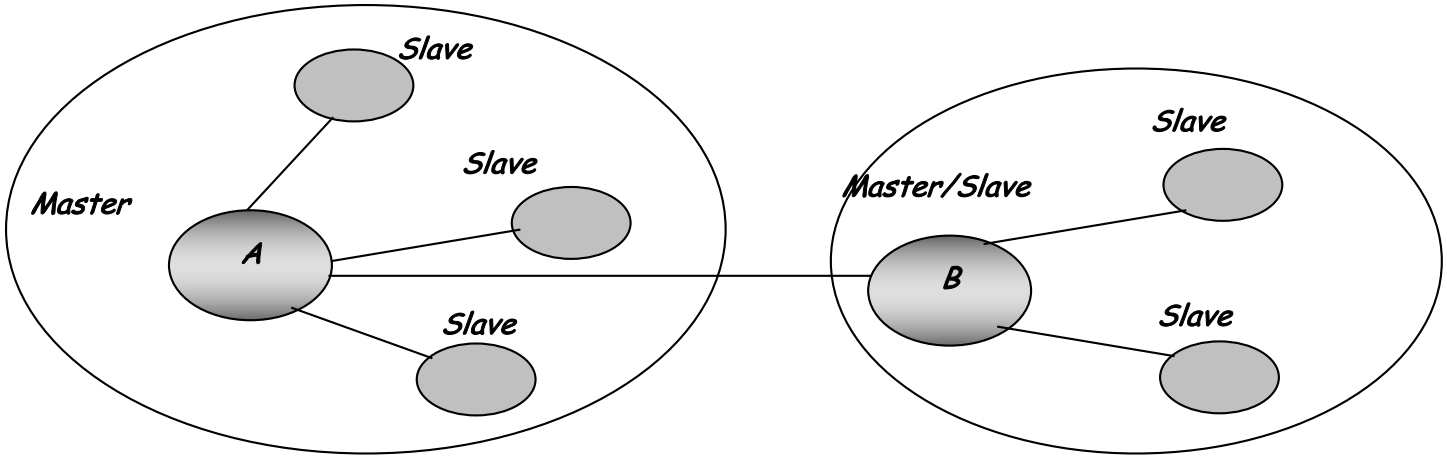
- Operates in the same 2.4 GHz bands as IEEE 802.11b
- It employs *frequency hopping* spread spectrum
  - Channels are 1 MHz wide
  - The frequency is changed *every* packet
- Devices within 10 m can share up to 720 kbps of capacity

Bluetooth Architecture: Piconet



- Basic unit of Bluetooth networking
- One master and up to 7 slaves
- The master also controls the transmission within its piconet
- There is NO contention within a piconet

Bluetooth Architecture: Scatternet



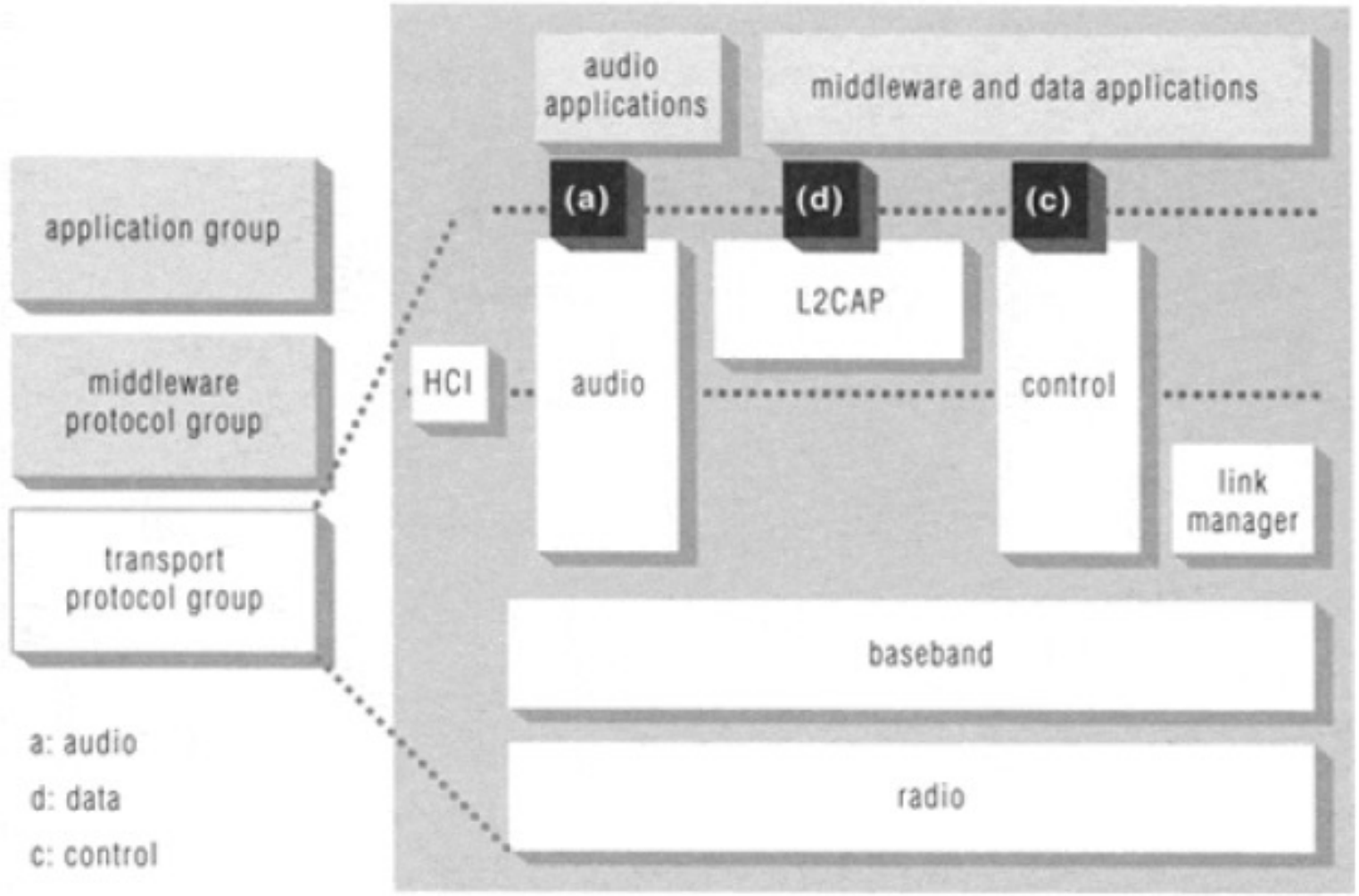
- A device can belong to several piconets
- A device can be the master of only one piconet
- A device can be the master of one piconet and slave of another piconet or a slave in different piconets

The Bluetooth Protocol Stack

The Bluetooth Protocol Stack

- Transport Protocol group
  - Protocols designed to allow Bluetooth devices to locate each other and to create, configure and manage both physical and logical links
- Middleware protocol group
  - Protocols needed for existing and new applications to operate over Bluetooth links
  - PPP, IP, TCP, RFCOMMM (enables legacy applications that normally would interface with a serial port to operate seamlessly over Bluetooth)

Transport Protocol Group Stack



Audio traffic bypasses all of the intermediary protocol layers and is funneled directly from the audio application to the baseband layer

The L2CAP Layer

- Logical link control and adaptation protocol
- Shields higher-layer protocols and applications from the details of the lower-layer baseband protocols
  - No need to know about Frequency Hoping
  - It enables segmentation of large packets used by higher layers into smaller packets for baseband transmission and the corresponding
  - Reassembly of those packets by the receiving device

The Link Manager Layer

- Supervise device *pairing*
  - Creation of a trust relationship between the devices by generating and storing an authentication key for future device authentication
- If authentication fails, the link managers may sever the link between the devices, thus prohibiting any communication between the devices
- Encryption of the data flowing over the air-interface between the devices whenever needed

Baseband

- The equivalent of the MAC layer in IEEE 802.11
- Defines the protocol for multiple channel access
- Defines the master and slave roles
- Provides functionality to determine nearby Bluetooth devices
- Etc.

Host Controller Interface (HCI)

- Most Bluetooth systems consist of two processors:
  - The higher layers of the protocol stack (L2CAP, SDP, RFCOMM) are run on the host device's processor
  - The lower layers of the protocol stack (Baseband and radio) are run on specific Bluetooth hardware
- HCI provides an interface between the higher and the lower layers of the protocol stack
  - To enable the development of interoperable Bluetooth modules by different vendors

BASEBAND LAYER

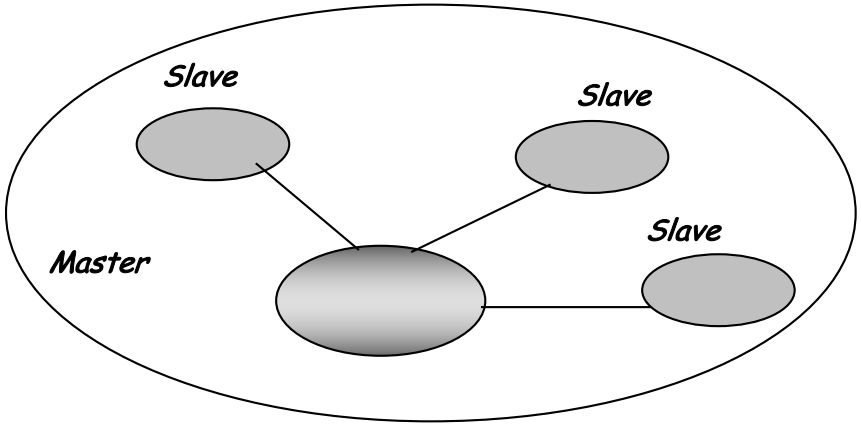
Baseband Layer Functionality

Two main functionalities

1. Create a piconet

2. Handle channel access

Bluetooth Architecture: Piconet



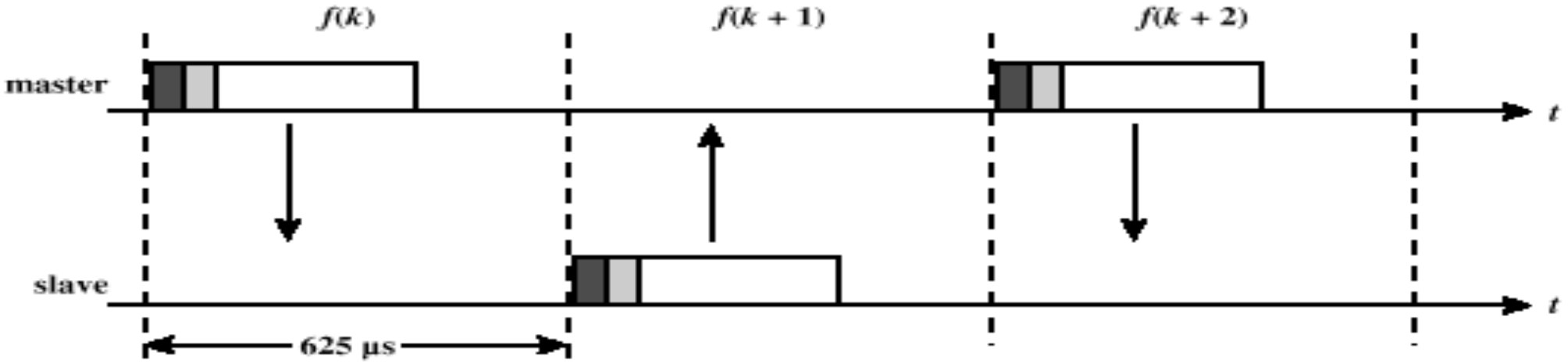
- Basic unit of Bluetooth networking
- One master and up to 7 slaves
- The master also controls the transmission within its piconet
- There is NO contention within a piconet

Channel Access in bluetooth

Frequency Hopping (FH)

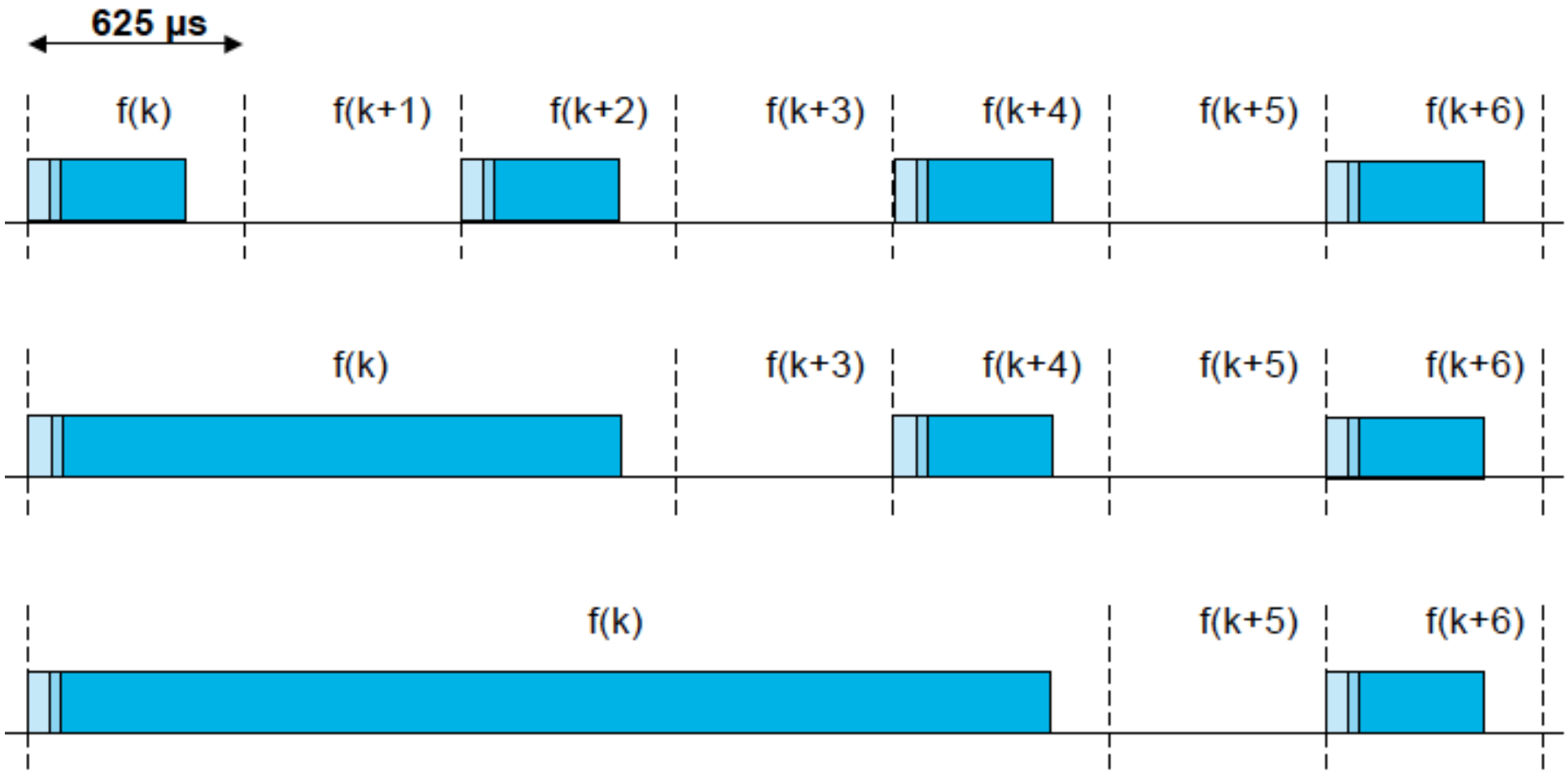
- Total bandwidth divided into 1 MHz channels for a total of 79 channels
- FH occurs by jumping from one channel to another in pseudorandom sequence
- The pseudorandom hopping sequence is shared across the entire piconet
  - As seed serve the 28 least significant bit of the master's Bluetooth device address
- Resists interference and multipath effects
- Provides a form of multiple access among co-located devices in different piconets

Frequency Hopping



- The frequency is changed every slot
  - The RF shall remain fixed for the duration of the packet
- The slot size is set to 625 µsec (1600 hopings/sec)
- A packet transmission can span 1,3 or 5 slots

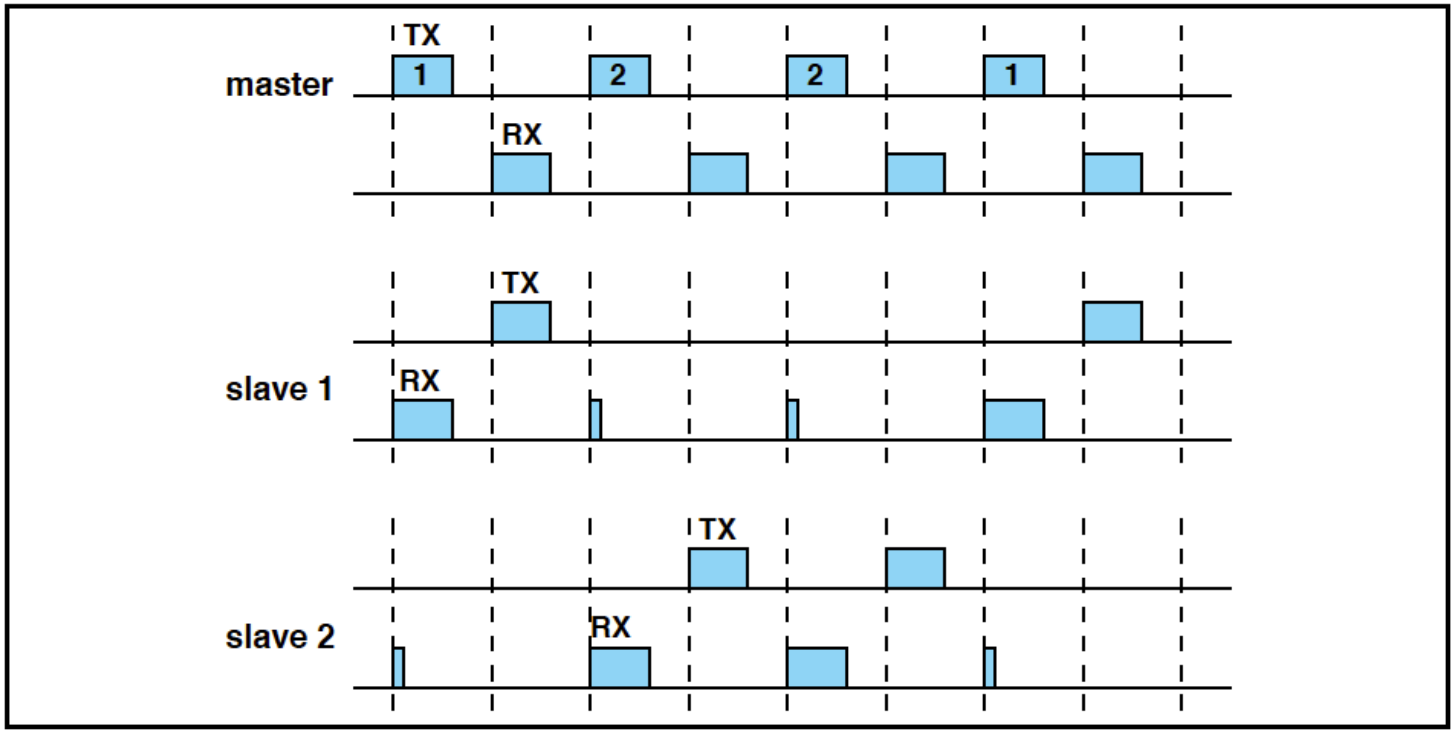
A Frame Can Span 1, 3 or 5 Slots



Channel Access: TDD

- The Master device is the one initiating an exchange of data
- The Slave device responds to the Master
  - Slaves use the frequency hopping pattern specified by the Master
- A slave can transmit ONLY in response to a Master

Channel Access: Example



- The master always transmits on the odd-numbered slots
- The slaves transmits on the even-numbered slots in response to the master

Establishing a Piconet

Creating a Piconet

- Piconet comprises a shared communications channel through which members of the piconet communicate
- The communication channel consists of a well-defined sequence of frequency hops
- *The question is, how do devices create a piconet?*

Establishing a Bluetooth Connection

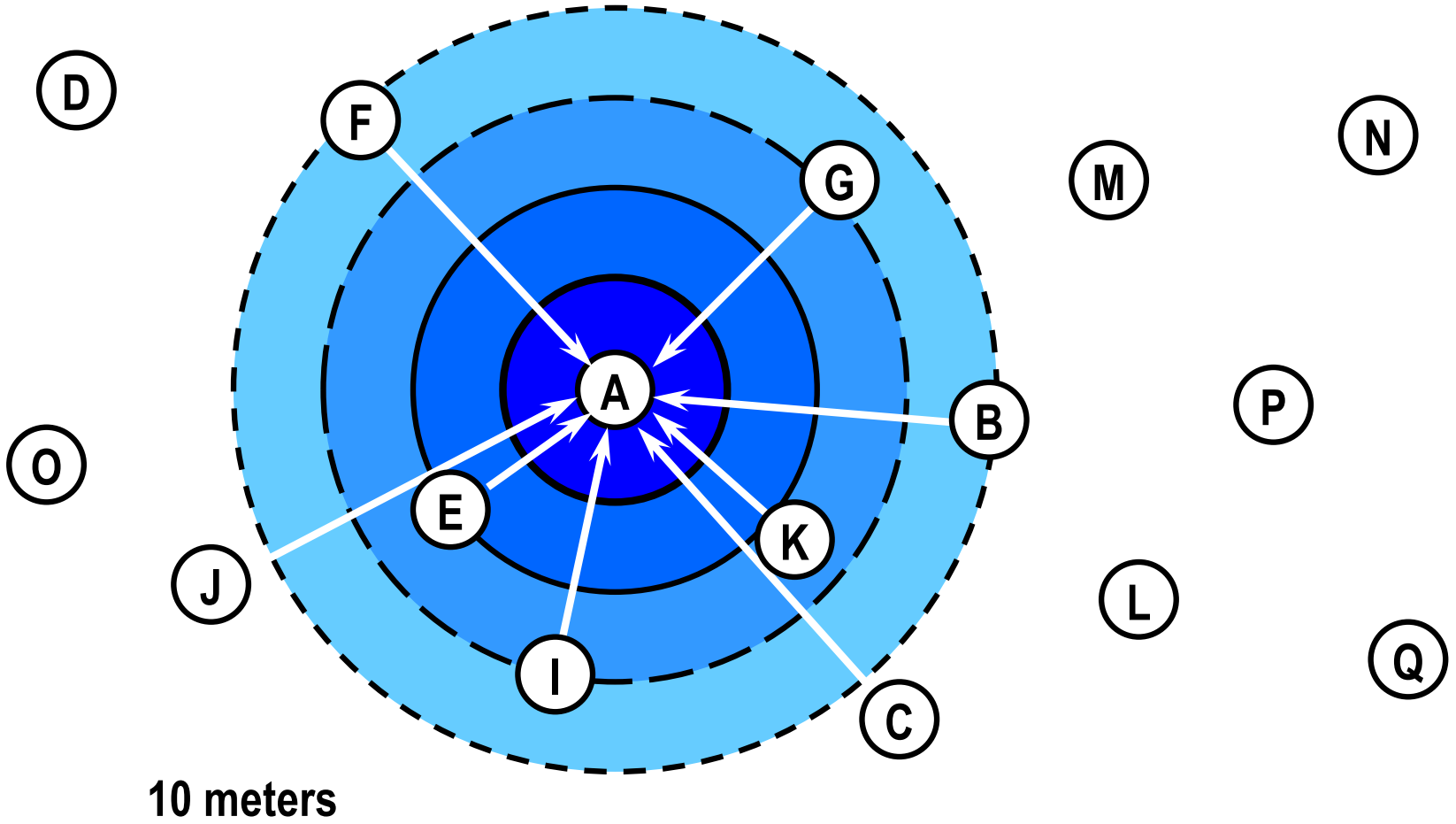
Two step process

1. Inquiry
2. Paging

Inquiry Procedure

- Sends out an inquire, which is a request for nearby devices (within 10 meters)
- Devices that allow themselves to be discoverable issue an inquiry response
  - Can take up to 10.24 seconds, after which the inquiring device should know everyone within 10 meters of itself

Node A Sends an Inquiry



After inquiry procedure, A knows about others within range

Issues with Inquire Messages

- Are the inquirer transmitting and the receiver listening on the same frequency?
  - Since they are not yet connected, they are on totally different hop sequences, and most likely on different channels
- If they are on the same frequency, what if they are on a noisy channel?
  - Bluetooth provides the capability for receivers to issue multiple inquiry responses

Transmitting Inquiry Messages

- Inquiring device sends out an inquiry on 16 different frequencies
- The 16 frequencies form the inquiry hop sequence, called a train
- There are two trains
  1. Generated by using the 28 least significant bits of the General Inquiry Access Code (GIAC)
  2. Generated by using the 28 least significant bits of the Dedicated Inquiry Access code (DIAC) -> printers

Transmitting Inquiry Messages - 2

- The transmission is carried out in every alternate slot and the intermediate slots are used for listening to responses if any
  - Two inquiries are transmitted per slot

What about noise?

- Devices always reply to received inquiry messages with an inquiry response
  - An inquirer is allowed to received multiple responses from one device
- In order to account for the fact that channels can be noisy and transmissions can get lost, the train scan is repeated up to 4 times for each train
  - Designed to successfully communicate at least once with all devices within range

Inquiry Scan

- A device periodically listens for inquiry packets at a single frequency – chosen out of 16 frequencies

- Stays in the state long enough for a inquiring device to cover 16 frequencies

Inquiry Response

- When an inquiry message is received in the inquiry scan state, a response packet containing the responding device address must be sent
  - It is not sent in the immediately following slot after the slot in which inquiry is received to avoid collisions
- Waits for a random number of full 16-channel scans (0-127) and then sends its FHS packet to the inquirer that contains
  - The device address and its clock
  - Information about when the device enters its page scan states
- The inquiring device on receiving an inquiry does not acknowledge the response but continues its inquiry procedure
- Only when the inquiring device wants to page the device that responded it will use the response information to page

10.24 seconds?

- 2 trains x 128 times x 4 times x 16 x 625μsec

Paging

- After the inquiry has been successfully carried out the device will start a paging procedure if a connection is desired
- Paging requires only the address of the device to be paged but the clock information, from the FHS response packet, may be used to speed up the procedure
- The device starting the paging procedure will be the master of the piconet consisting of itself and the paged device if the paged device accepts the connection

Paging Process

- Very similar to inquire
- Connection process involves 6 steps of communication between the master and the slave

Step 1: The Page Scanning

- Device transmits a page message out to the device that it wants to set up a connection with
  - Does this in a similar manner as inquire messages (on 2 frequency trains of 16 frequencies each)
- Once the device receives a page response, it will stop paging and move on to step 2

Paging: Steps 2 & 3

- Step 2: In the page response, an acknowledgement is sent back to the master containing the slave ID
- Step 3: Master sends an FHS packet to the slave informing the slave of the master's clock

Paging: Step 4

- Using the data from the FHS packet, the slave adopts the master's frequency hopping pattern and synchronizes to its clock

- The slave issues a final slave response transmission that is aligned to the slave's native clock

Paging: Step 5

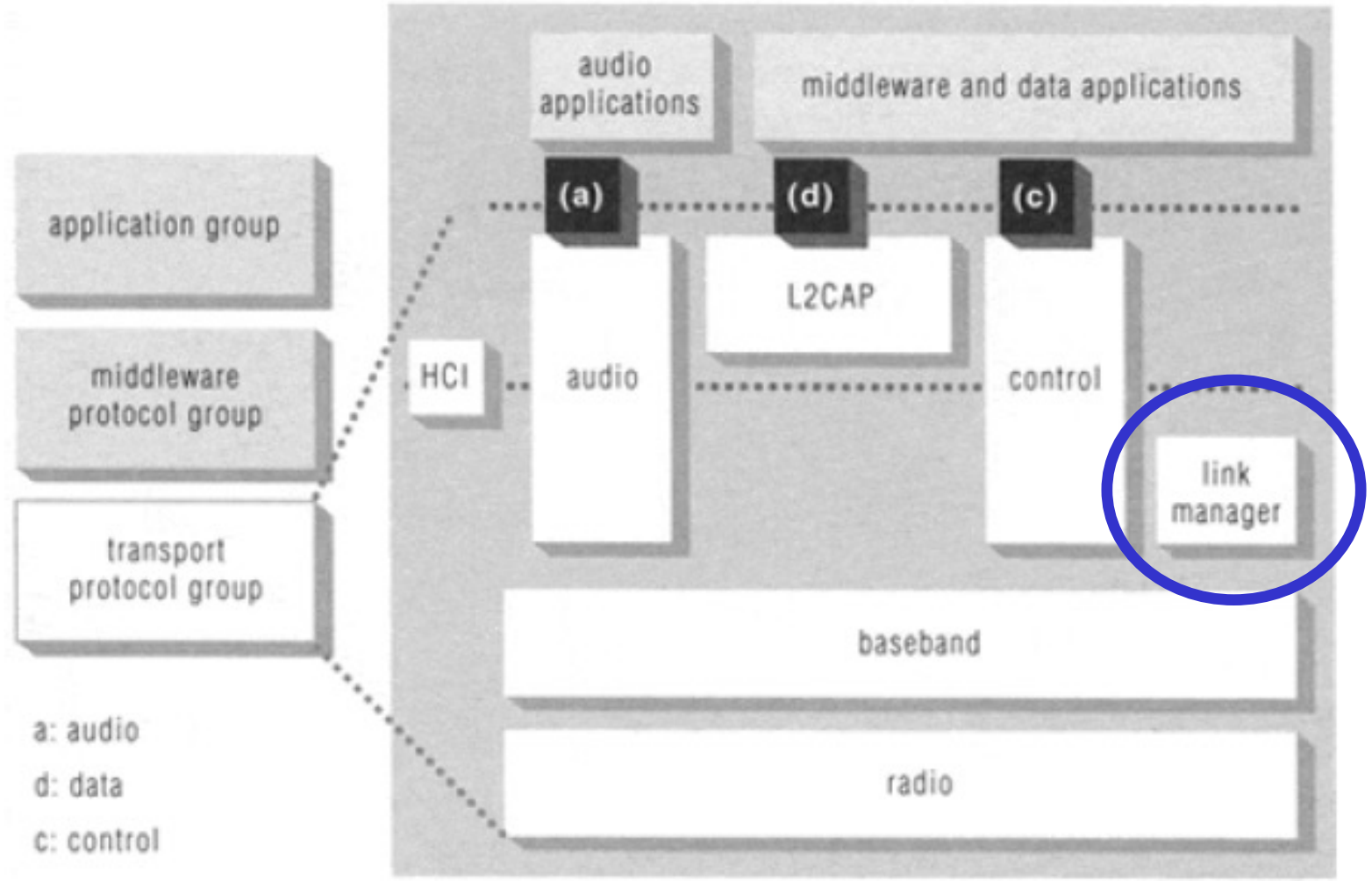
- When the master receives the packet, it jumps back to its frequency hopping pattern and assigns the slave an Active Member Address (AMA) for the piconet

- Master sends out a poll packet to ensure that the slave is on its frequency hopping pattern

Paging: Step 6

- Once the slave receives the poll packet, the slave replies with any kind of packet to ensure that it is on the right channel
- The acknowledgement must be received by the Master within the timeout period
- At the conclusion of step 6, a new synchronized connection is established between the master and the slave

Transport Protocol Group Stack: Link Manager



Link Manager

- Performs all link creation, management, and termination operations
- Responsible for all the physical link resources in the system
  - Handles the control and negotiation of packet sizes used when transmitting data
- Controls Operation Modes for devices in a piconet
- Sets up, terminates, and manages baseband connections between devices
  - Establishes different types of links dependent on requests from the L2CAP layer
    - Synchronous Connection-Oriented (SCO)
    - Asynchronous Connection-Less (ACL)

Asynchronous Connection-Less (ACL)

- Designed for data traffic
- Packet switched connection where data is exchanged sporadically as and when data is available from higher up the stack
- Data integrity is checked through error checking and retransmission
- One ACL link between a master and a slave

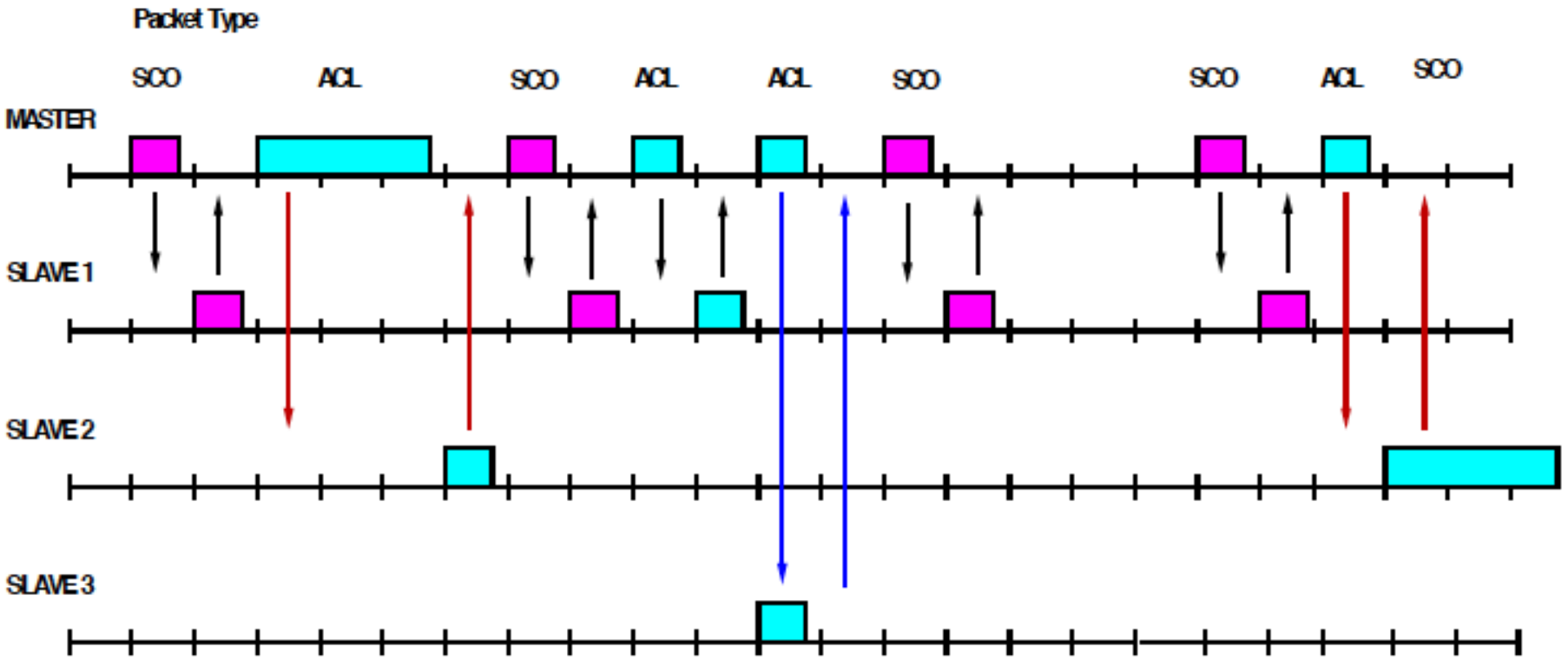
Synchronous Connection Oriented (SCO)

- Intended for use with time-bounded information such as audio or video
- Provides a circuit-switched connection where data is regularly exchanged
- Retransmission is not necessary, since data is real-time
- Up to 3 SCO links per piconet

ACL Links vs. SCO Links

	Intended Traffic Type	Retransmission	Max # links between master and slave	Supported during hold mode	Switched connection type
ACL	Data	Yes	1	No	Packet
SCO	Time bounded info (Audio or Video)	No	3	Yes	Circuit

Multiple Links with Mixed Packets



Power Management and Power-Managed States

Link Manager Operation

- Devices operate in standby mode by default until they become connected to a piconet
- 4 Connection Modes
  - Active
  - Hold
  - Park
  - Sniff
- Modes allow devices to adjust power consumption, performance, and the number/role of participants in a piconet

Active Mode

- Limited to 7 Active slaves for each master
- Three bit address (AM_ADDR) given to each active slave
- Unit actively participates on channel
- Can receive communications in any given frame
- Active slaves are polled by master for transmissions
- Unit operates on high-power

Hold Mode

- Frees slave to
  - Attend another Piconet
  - Perform scanning, paging, or inquiry operations
  - Move into low-power sleep
- Unit keeps active member address
- Unit does not support ACL packets on the channel but may support SCO packets
- Master and slave agree on a one time hold duration after which the slave revives and synchronizes with channel traffic
- Unit operates on low-power

Sniff Mode

- Very similar to hold mode
- Slave is freed for reoccurring fixed time intervals
- Master can only communicate during arranged “sniff” time slots

Park Mode

- Parked unit gives up active member address
- To manage its fast and orderly readmission to the piconet, the master assigns to the slave two temporary 8-bit addresses
  - 8 bit Parked member address (PM_ADDR) – allows master to unpark slave
  - Parked devices could be recalled using their 48-bit *BD_ADDR* if needed, but the use of the much shorter *PM_ADDR* allows for increased efficiency
- Unit stays synchronized to channel
- Operates in very low-power sleep

Park Mode (cont.)

- Provides the ability to connect more than 7 devices to a master (8 bit PM_ADDR allows 255 parked devices)
- Active and Parked slaves can be switched in and out to allow many connections to a single piconet

Park Mode (cont.)

- Master establishes a beacon channel and beacon interval when a slave is parked
- Parked slave wakes up at regular beacon interval to
  - Maintain synchronization
  - Listen for "broadcast" messages (packets with all zero AM_ADDR)
  - Potentially make access request to master through (AR_ADDR)

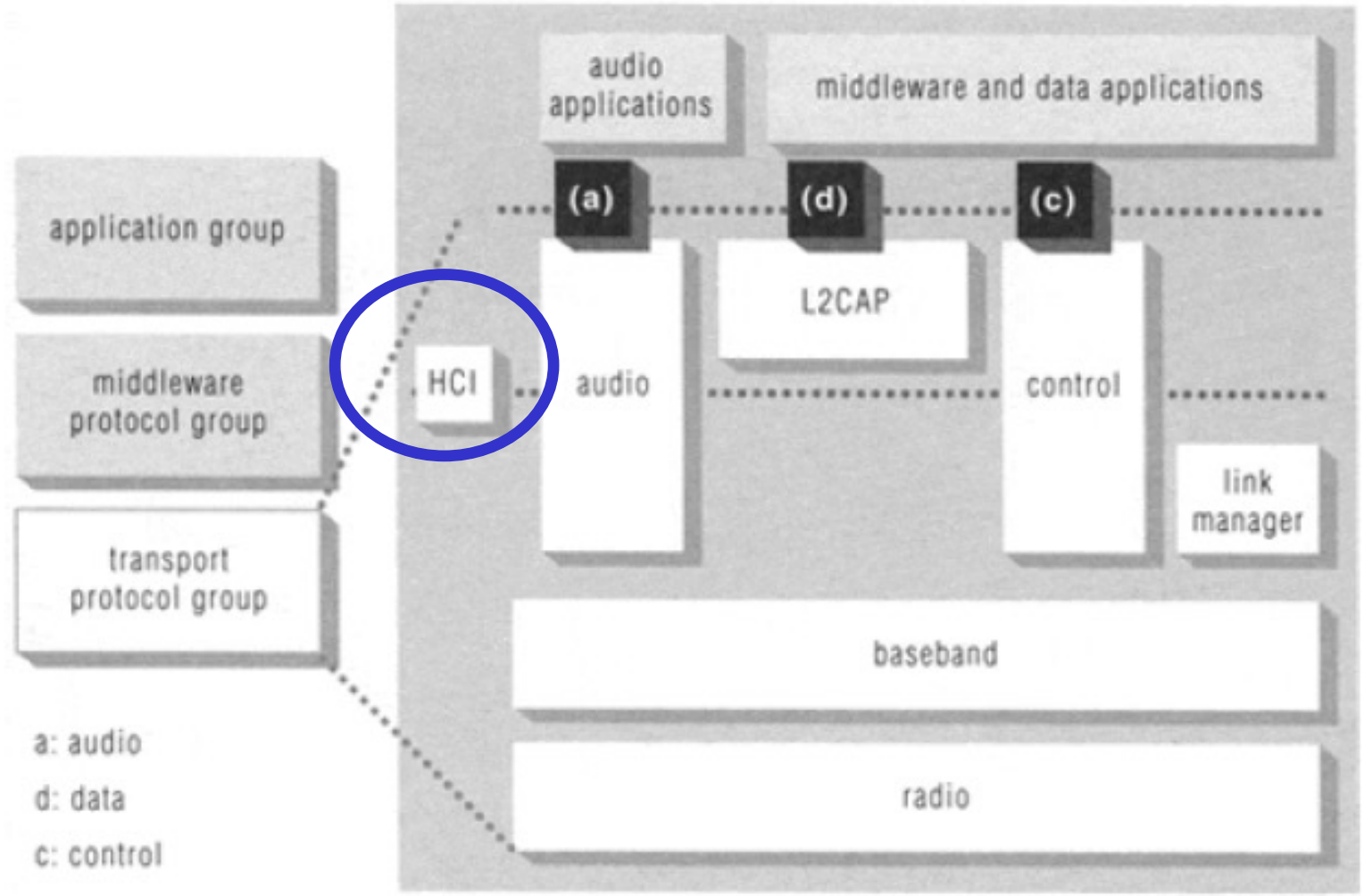
Park Mode (cont.)

- Beacon slots must have at least “null” master-to-slave traffic
- Master-to-slave transmissions may extend over multiple beacon slots

Security

- Link manager provides mechanism used by devices at either end of a link for
  - Negotiating encryption mode
  - Coordinating encryption keys
- Baseband handles encryption and key generation

Transport Protocol Group Stack: HCI



Host Controller Interface (HCI)

- Most Bluetooth systems consist of two processors:
  - The higher layers of the protocol stack (L2CAP, SDP, RFCOMM) are run on the host device's processor
  - The lower layers of the protocol stack (Baseband and radio) are run on specific Bluetooth hardware
- HCI provides an interface between the higher and the lower layers of the protocol stack

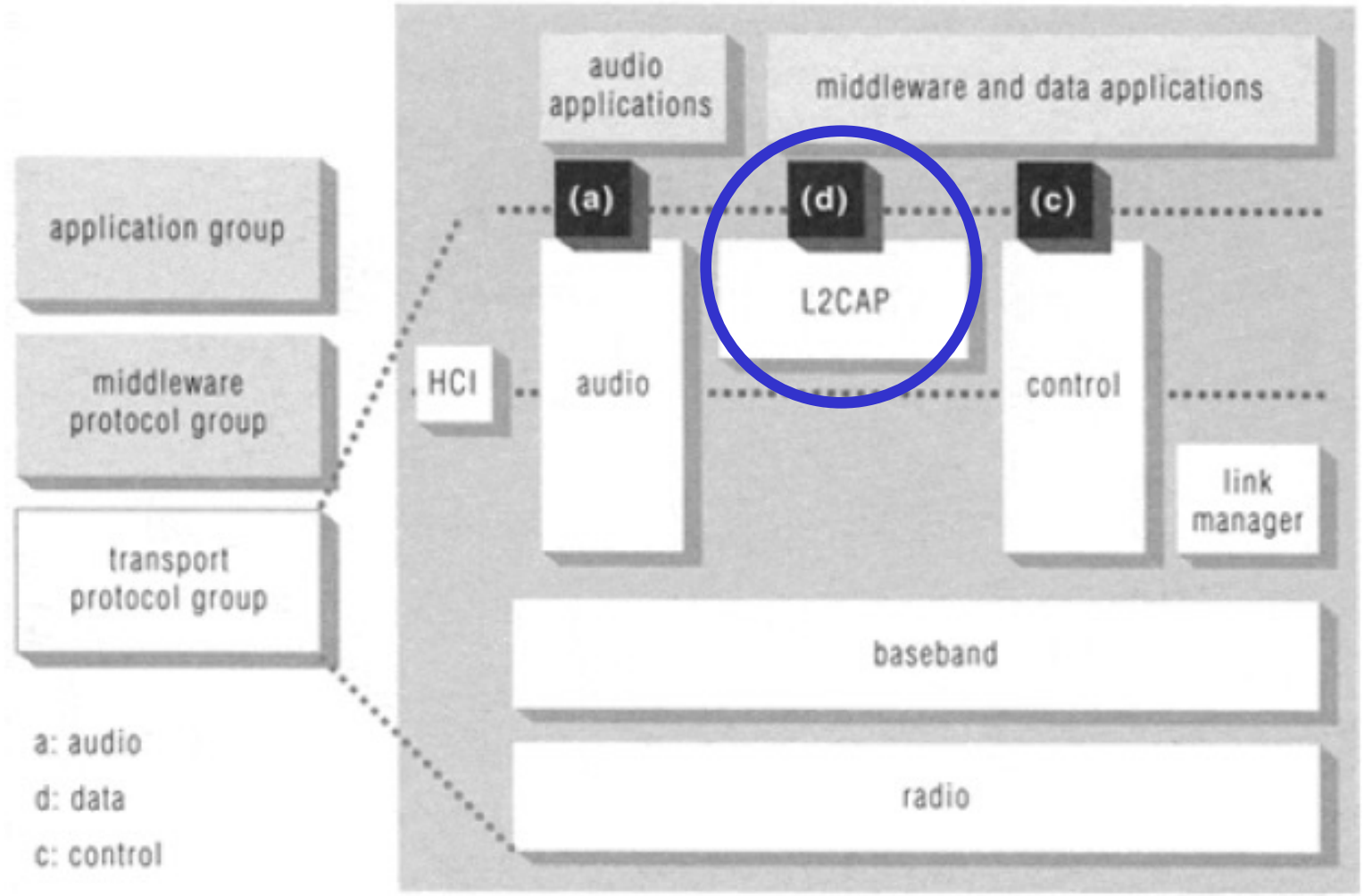
HCI Flow Control

- Main function of the Host Controller Interface
- Many times higher layer protocols have data rates much larger than data rate across Bluetooth radio and air interfaces
  - Also need to handle the reverse situation when the host cannot accept data as fast as the Bluetooth module can send it

Two Pieces of HCI

- Host controller resides on Bluetooth hardware accepting communications over the physical bus (radio and air)
- HCI Driver resides on the host accepting communications from higher layer protocols

Transport Protocol Group Stack: L2CAP



Logical Link Control and Application Protocol (L2CAP)

- Performs 4 major functions
  - Managing the creation and termination of logical links for each connection through "channel" structures
  - Enforcing and defining QoS requirements
  - Adapting Data, for each connection, between application (APIs) and Bluetooth Baseband formats through Segmentation and Reassembly (SAR)
  - Performing Multiplexing to support multiple concurrent connections over a single common radio interface (multiple apps. using link between two devices simultaneously)

Segmentation/Reassembly

- Baseband packet size is limited
  - Can handle payload of up to 2745 bits
- L2CAP accepts packet size up to 64kb
- L2CAP segments large packets into smaller baseband manageable packets
- Smaller received baseband packets are reassembled coming back up the protocol stack

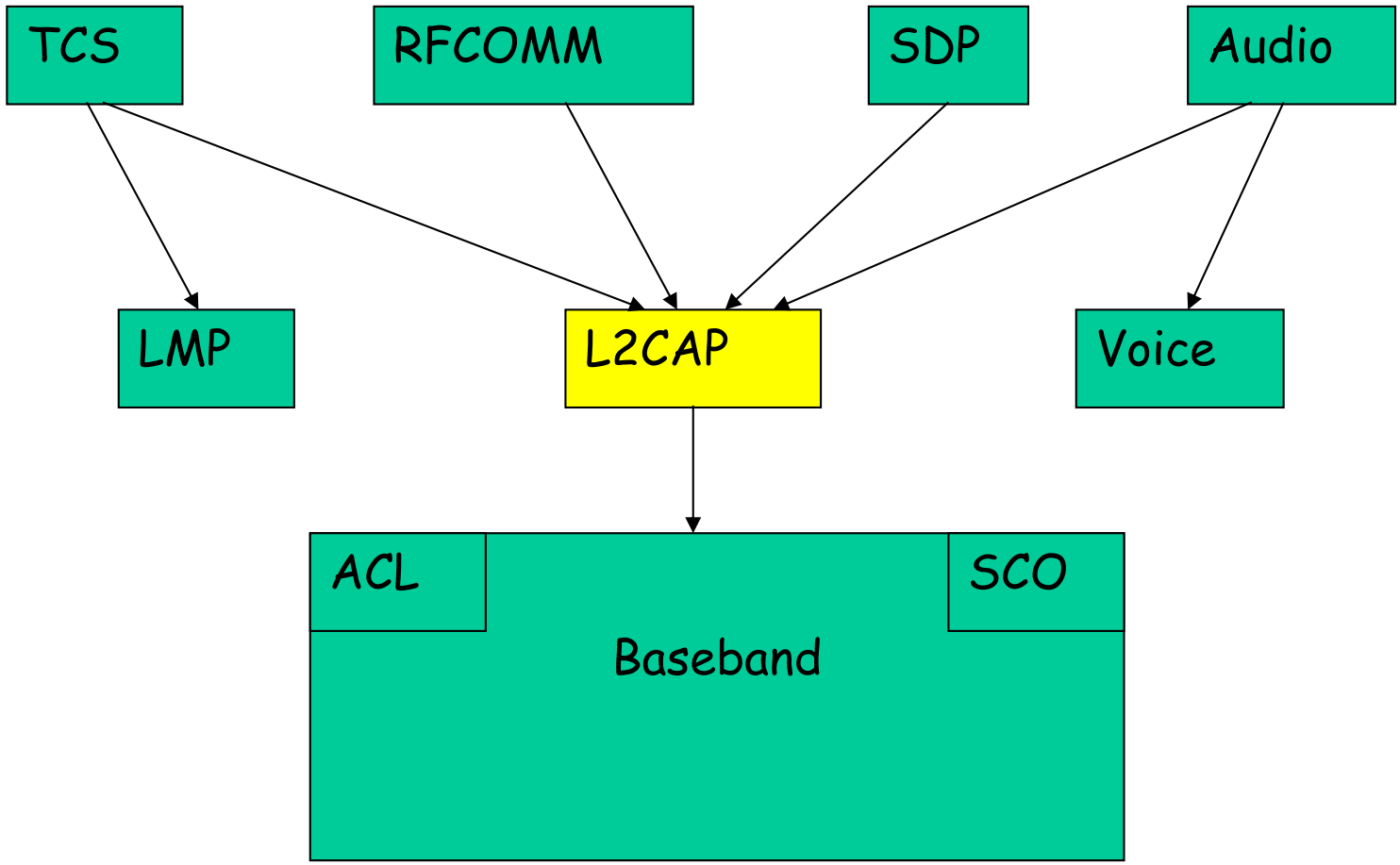
Quality of Service

- Applications may demand QoS on specific parameters
  - Peak bandwidth
  - Latency
  - Delay variation
  - Token rate
  - Token bucket size
- L2CAP provides requested QoS if possible and notifies application if link can not support demands

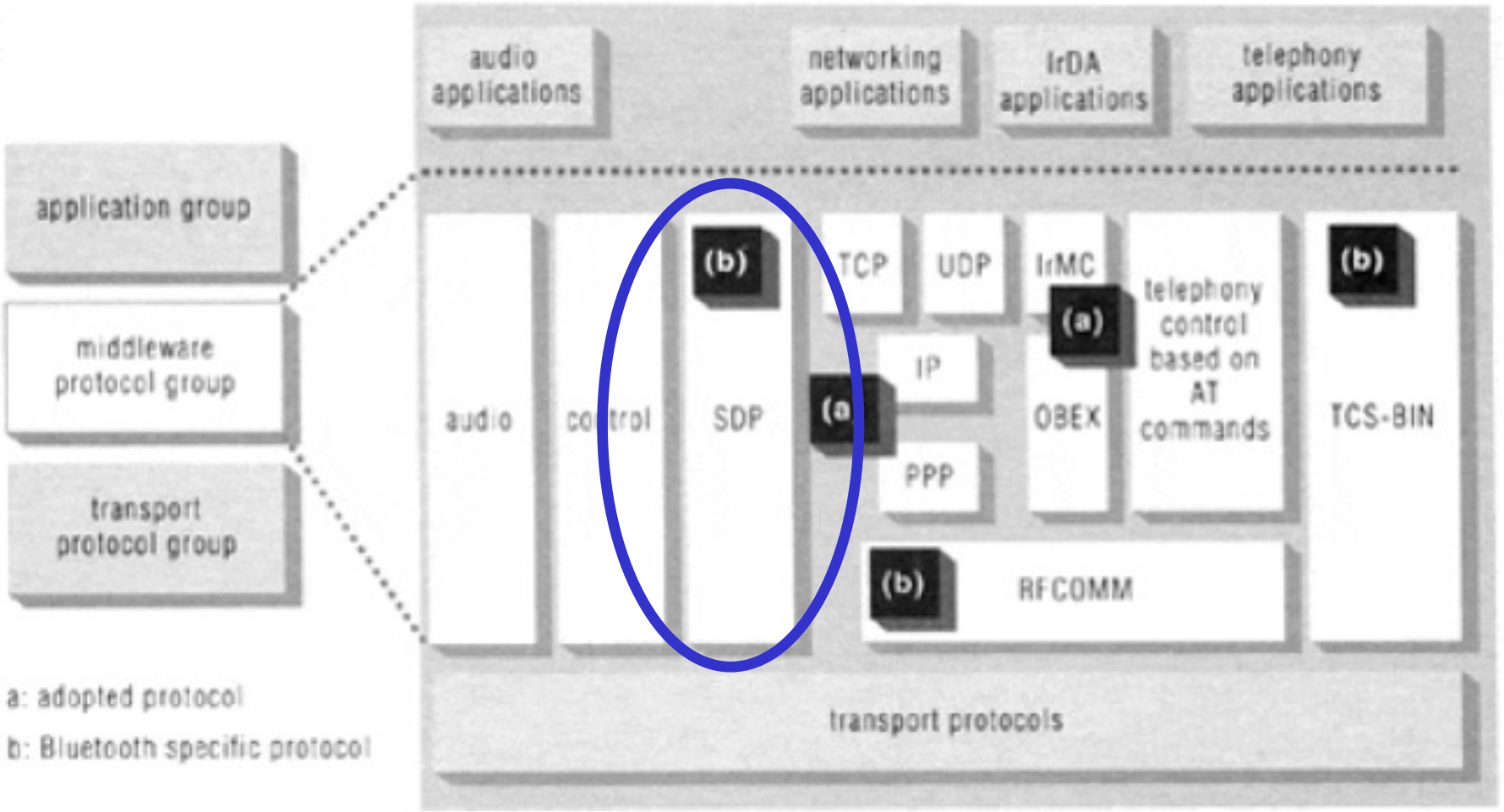
Protocol Multiplexing

- Applications may access L2CAP through different support protocols
  - Service Discovery Protocol (SDP)
  - RFCOMM
  - Telephony Control Protocol Specification (TCS)
- Baseband is not concerned with operation protocols meaning L2CAP must distinguish between them

Protocol Multiplexing Illustrated



Middleware


audio applications
networking applications
IrDA applications
telephony applications
application group
middleware protocol group
transport protocol group
(b)
TCP
UDP
IrMC
(a)
telephony control based on AT commands
(b)
audio
control
SDP
(a)
IP
OBEX
TCS-BIN
PPP
(b)
RFCOMM
transport protocols
a: adopted protocol
b: Bluetooth specific protocol

Service Discovery Protocol (SDP)

Idea:

- Traditional LANs: Find a connection to a printer (or other resource) and keep that connection for a long time
- Bluetooth: Walk into an area, find a printer (or other resource), use it, then walk away forgetting any details of the connection

SDP Client/Server Model

- SDP Server is any Bluetooth device that offers services to other Bluetooth device (ex. Bluetooth-enabled printer, etc.)
  - Each SDP Server maintains its own database that contains information about the services that it offers
- SDP Client is any Bluetooth device that uses the services offered by an SDP Server

SDP Query

- The SDP client queries an SDP server to find out what services are available
  - Uses the L2CAP link that is set up between the client and the server
  - L2CAP link provides information on services but doesn't handle any connection to services
  - Need to specify a class of services that the client wants to use (e.g. printing services)

SDP Database

- SDP Database is a set of records that describes the different services that the server can provide to another Bluetooth device
- When the SDP server gets a query, it looks up the service that the client is requesting and returns information to the client on how to connect to the service

Using the Services

- The SDP client establishes a separate (non-SDP) connection to use the service
  - SDP connection is only used to determine service availability
- The L2CAP connection used to get information for the service can be dropped (if no more services are needed) or retained (if the client still needs more services from the server)

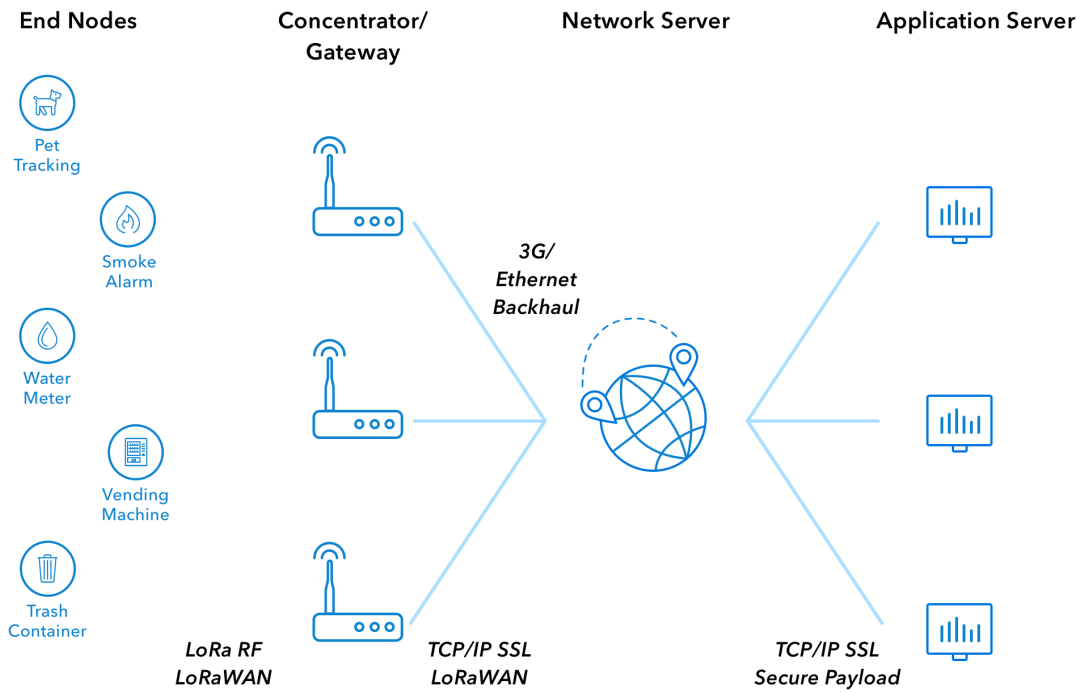
Summary

- Advantages of Bluetooth
  - Low power consumption
  - Low price on Bluetooth components
  - Non line-of-sight
- Disadvantages of Bluetooth
  - Wireless LANs offer faster data rates and larger communication ranges
  - Possibility of interference on 2.4GHz frequency band

LoRaWAN

Specification	LoRa Technology Support
Standard	LoRa Alliance
Operational Frequencies	Unlicensed ISM band 868, 915 MHz
Modulation	Chirp spread spectrum (CSS)
Coverage Range (Km)	2 - 5 (urban) / 15 (rural)
Data Rate (kbps)	0.3 - 50 (EU) / 0.9 - 100 (US)
Topology	Star

LoRaWAN Architecture



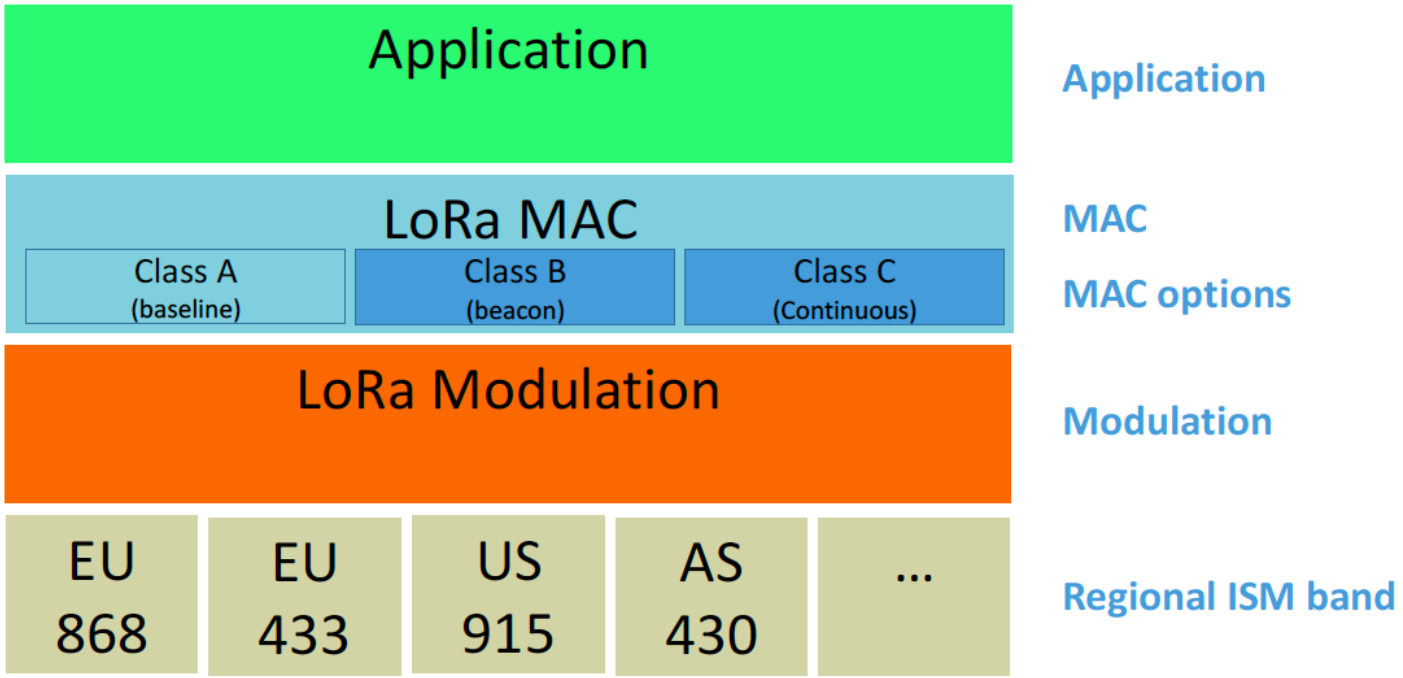
Components

- End device (ED): the sensors
- GW: provide connection to the internet; resolve collisions (the key to both LoRa and Sigfoxx)
- Network server
  - Monitoring the GWs and EDs
  - Aggregate incoming data – remove duplicates
  - Selecting via which GW to communicate to an ED
- Application server
  - The actual IoT application

End devices

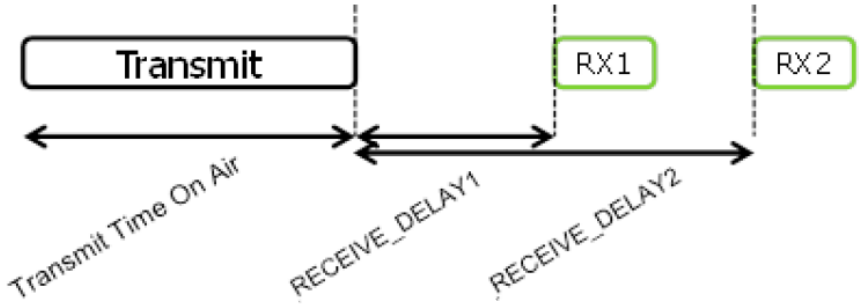
- LoRaWAN specification (https://lora-alliance.org/lorawan-for-developers):
- End-devices may transmit on any channel available at any time, using any available data rate, as long as the following rules are respected:
  - The end-device changes channel in a pseudo-random fashion for every transmission (the GW does not need to !)
  - The end-device respects the maximum transmit duty cycle relative to the sub-band used and local regulations
  - The end-device respects the maximum transmit duration (or dwell time)

End device classes



End device classes – Class A

- Bi-directional end-devices (Class A, mandatory)
  - Each end-device's uplink transmission is followed by two short downlink receive windows (1 s +/- 20 microseconds)
  - Transmission slot scheduled by the end-device is based on its own communication needs
  - Aloha-type protocol



End device classes – Class B

- Bi-directional end-devices with scheduled receive slots (optional)
  - In addition to A, it opens received slots at scheduled time intervals
  - Needs synchronization – GW beacons

End device classes – Class C

- Bi-directional end-devices with maximal receive slots (optional):
  - Nearly continuously open receive windows, only closed when transmitting
  - Worst in terms of energy, best in terms of delay

The ED Classes Tradeoff