Signal propagation

IoT, a.a 2017/2018

Prof. Chiara Petrioli

Un. of Rome "La Sapienza"

Chiara Petrioli[†]

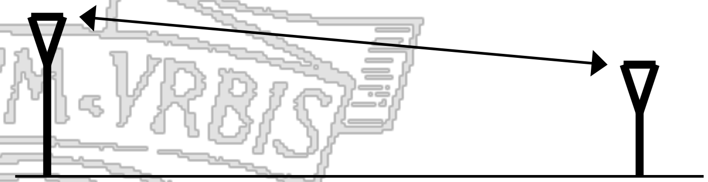
[†] *Department of Computer Science – University of Rome "Sapienza" – Italy*

Wireless channel

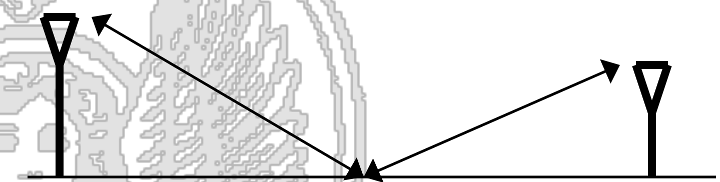
- Much less reliable than wired channels
- While propagating the signal can face
  - Attenuation as function of the distance from transmitter and receiver
  - Attenuation due to obstacles
  - Propagation over multiple paths (resulting in multipath fading)

Radio signal propagation

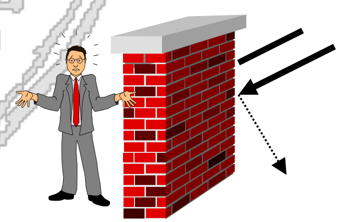
- Line of sight



- Reflection

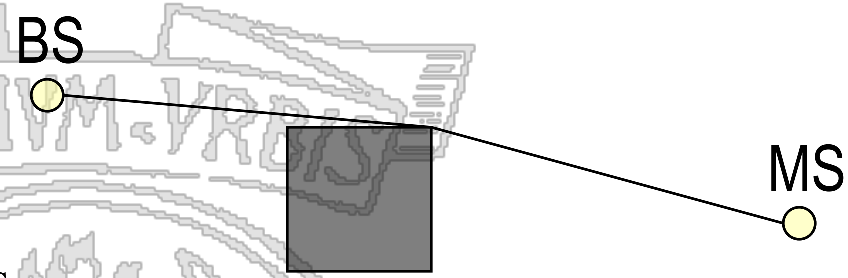


- Shadowing



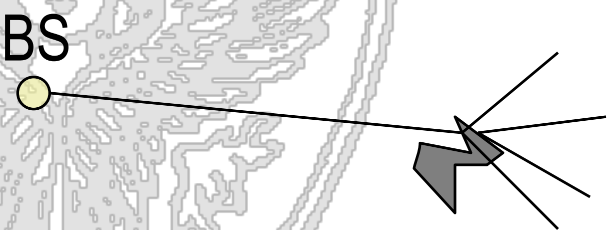
➔Diffraction

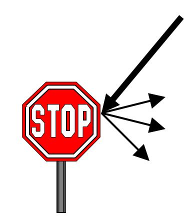
- ➔ When the surface encountered has sharp edges ➔ bending the wave



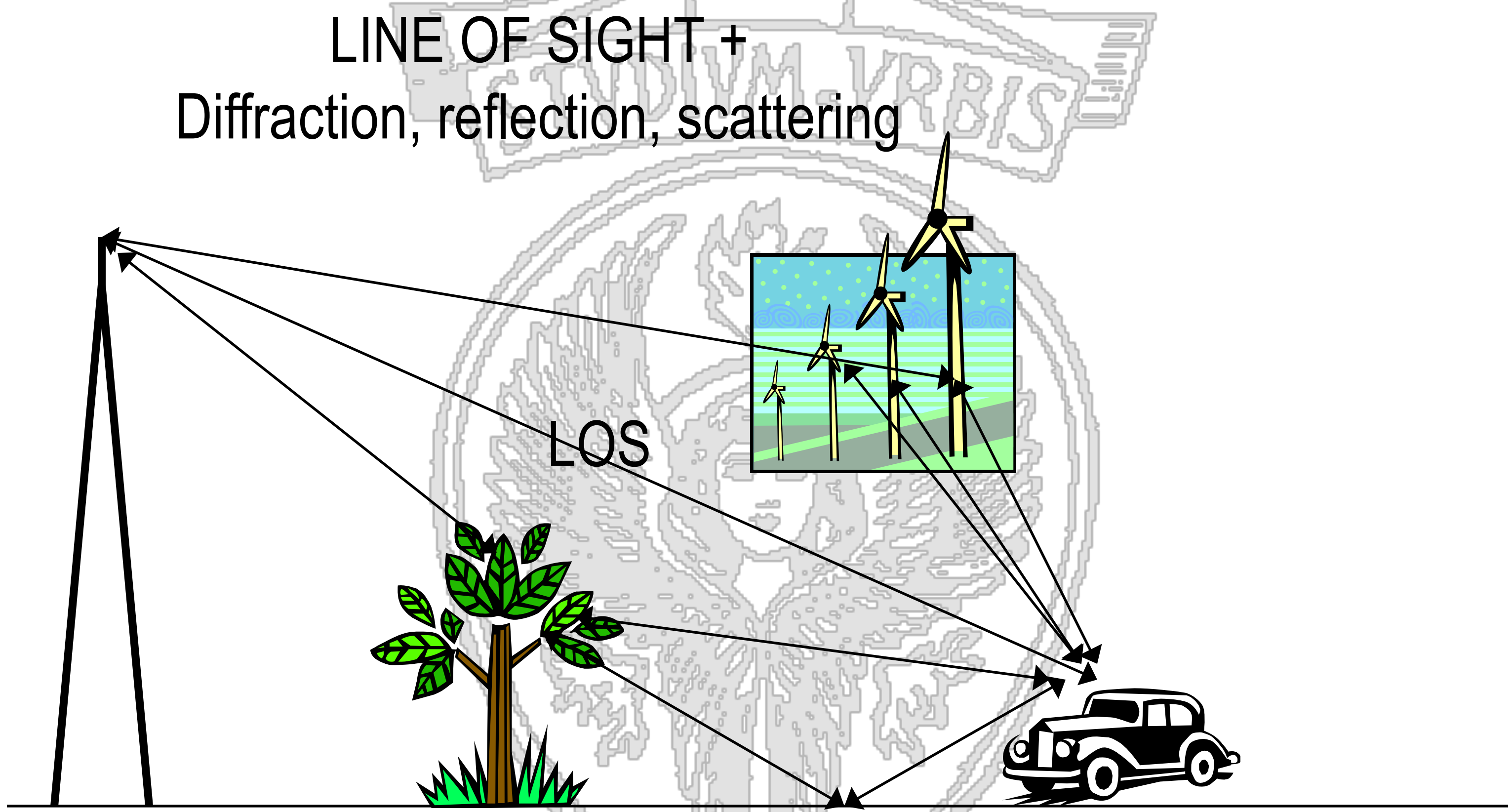
➔Scattering

- ➔ When the wave encounters objects smaller than the wavelength (vegetation, clouds, street signs)



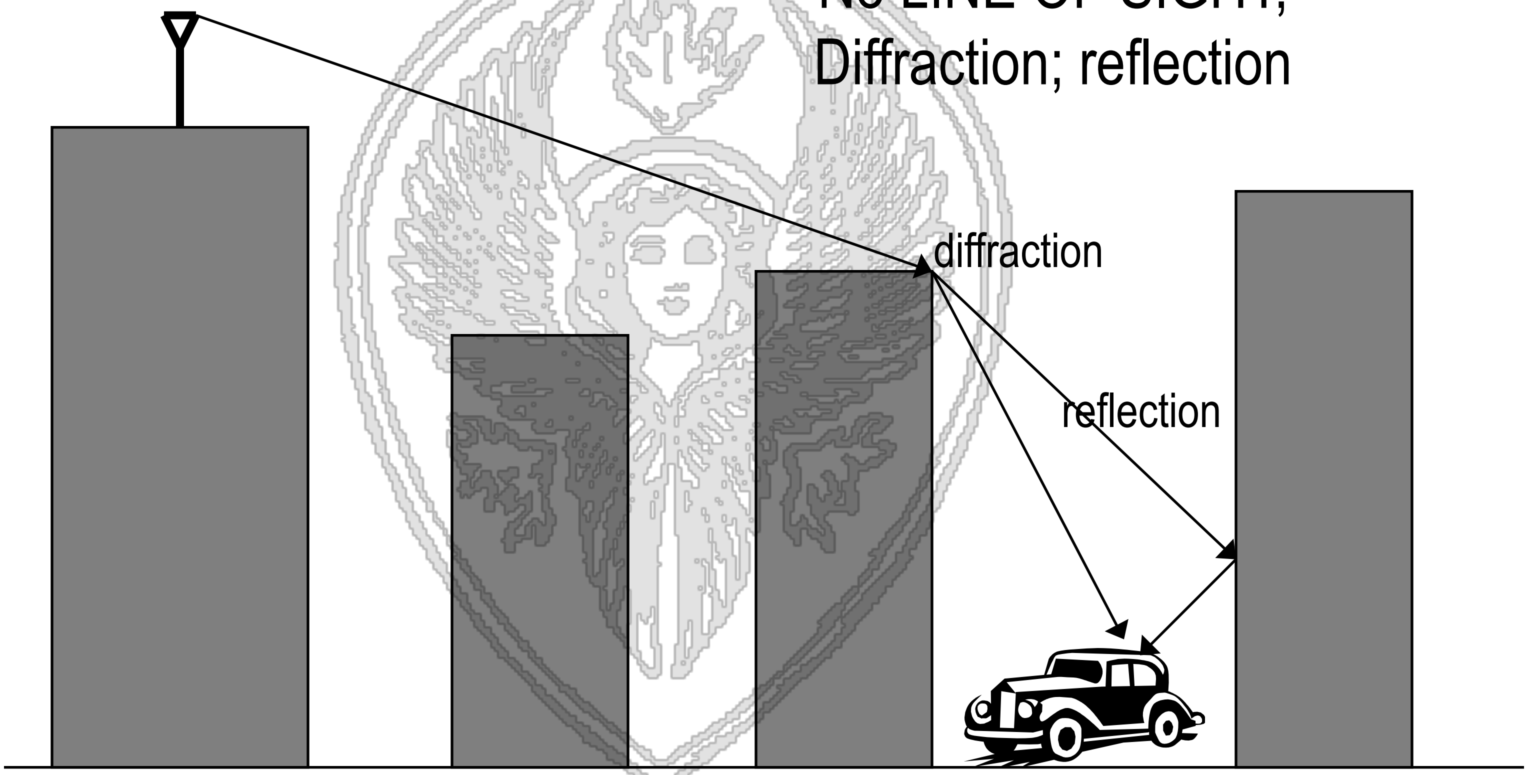


Radio signal propagation

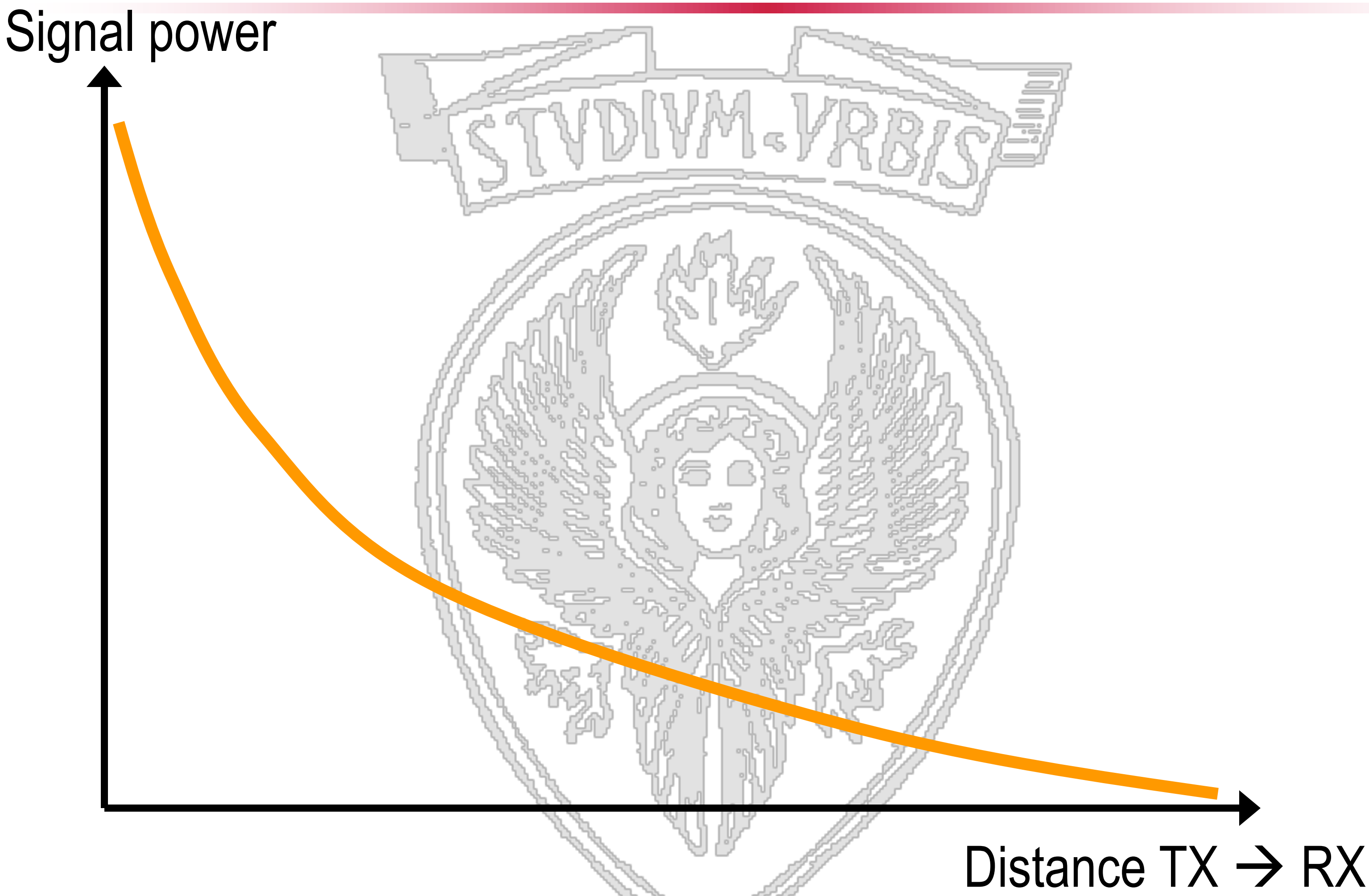


LOS path non necessarily existing

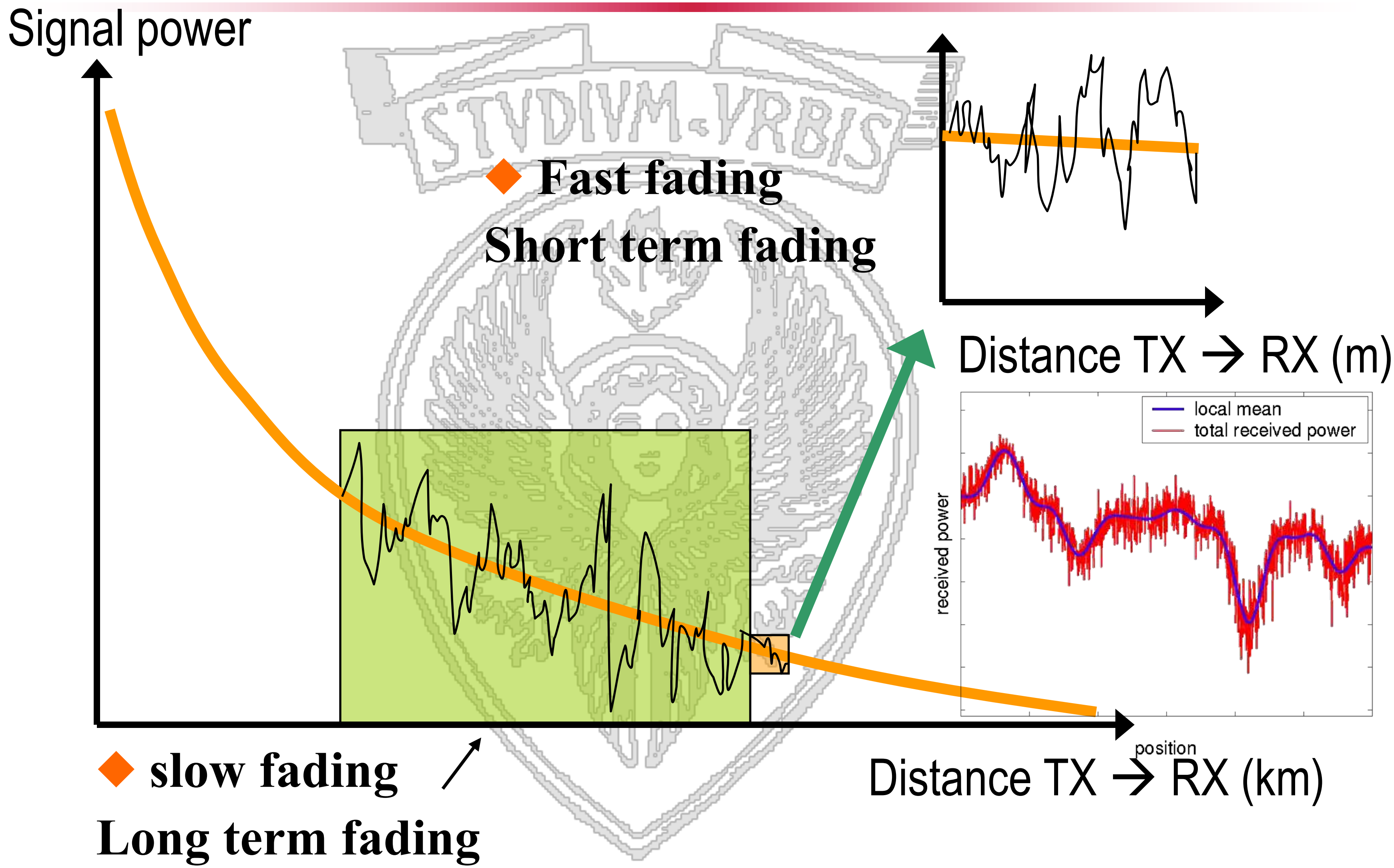
Example: city with large buildings;
No LINE OF SIGHT;
Diffraction; reflection



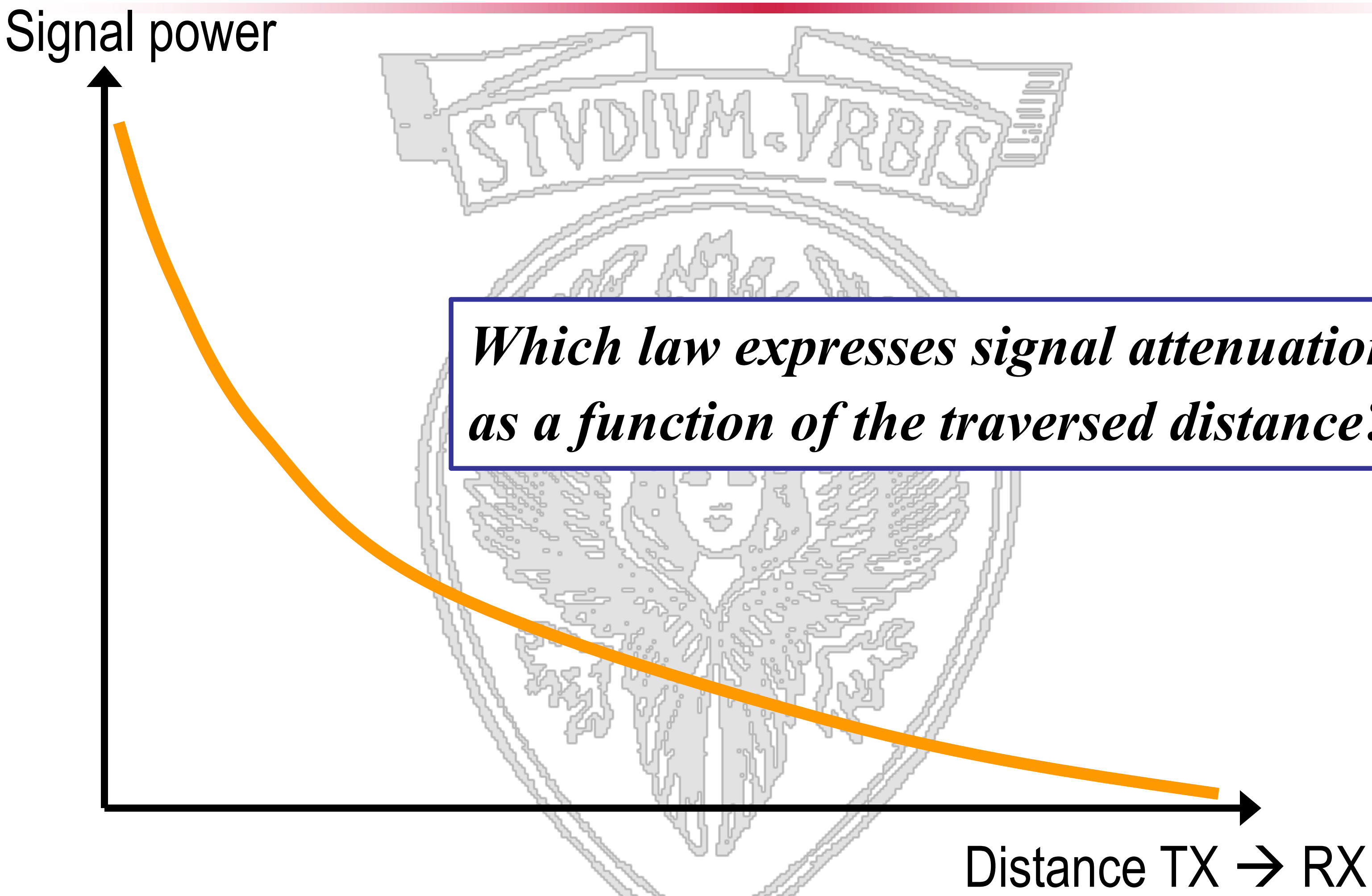
Radio signal attenuation



Slow fading – fast fading



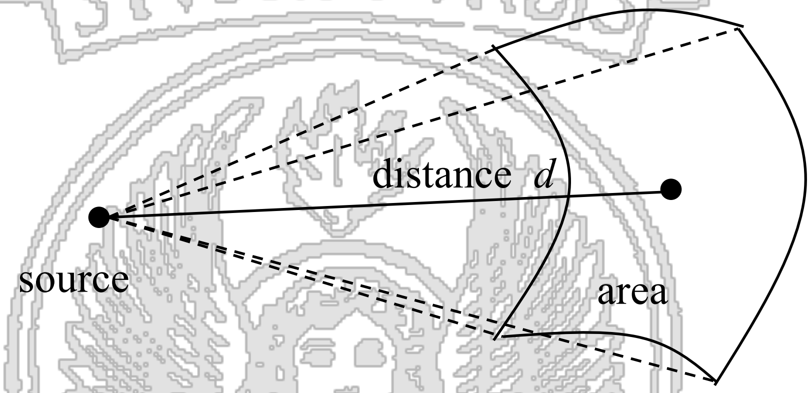
Radio signal attenuation



Which law expresses signal attenuation as a function of the traversed distance?

Radio signal attenuation geometrical spreading

- Assumption: A point source emits the signal uniformly in all directions (isotropic radiator) with a transmission power P_T

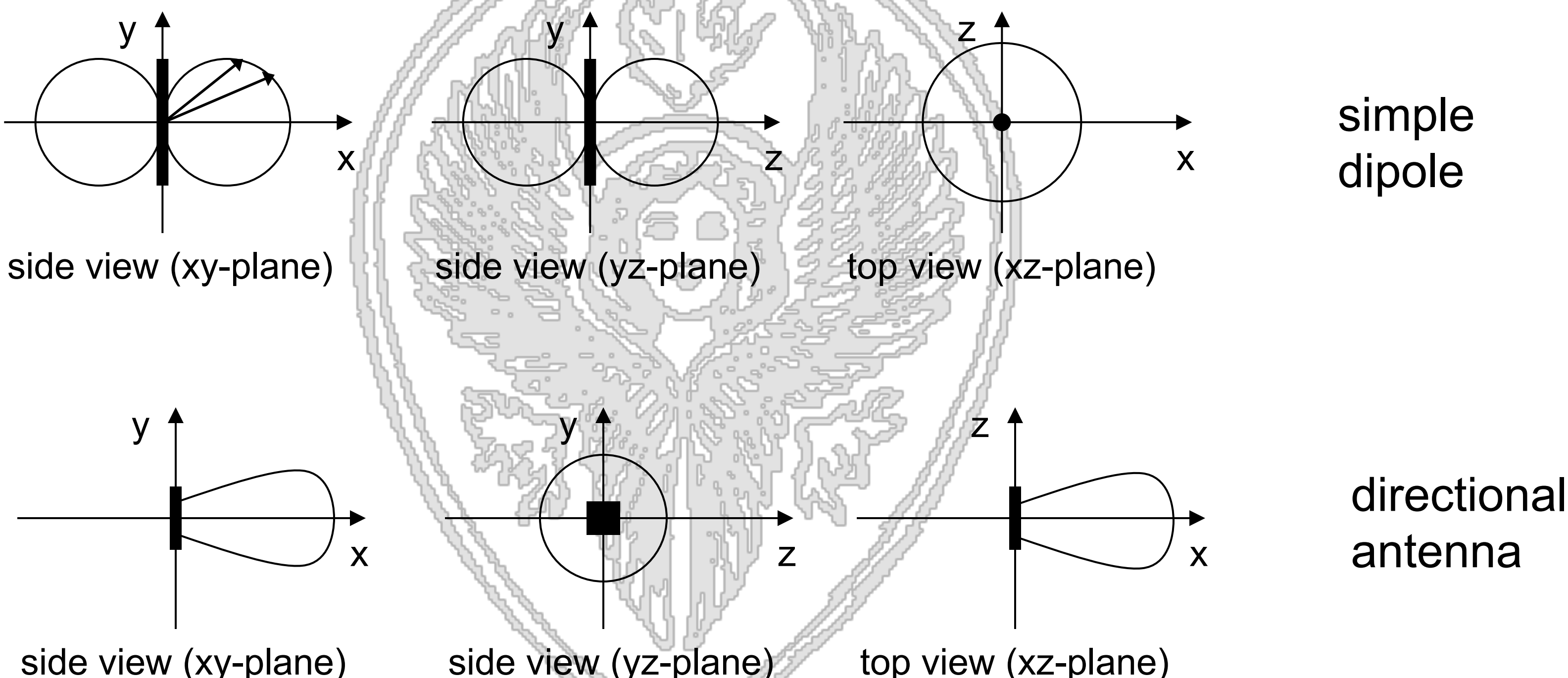


- The power density at distance d is equal to the ratio between the transmission power and the surface area of a sphere centered in the source and with radius d:

$$F = \frac{P_T}{4\pi d^2} \quad [\mathrm{W/m^2}]$$

Antenna types

- Graphical representation of radiation properties of an antenna
- Depicted as two-dimensional cross section



Antenna Gain

- Isotropic antenna (idealized)
  - Radiates power equally in all directions (3D)
  - Real antennas always have directive effects (vertically and/or horizontally)
- Antenna gain
  - Power output, in a particular direction, compared to that produced in any direction by a perfect omni-directional antenna (isotropic antenna)

$$\text{Directivity D} = \frac{\text{power density at a distance d in the direction of maximum radiation}}{\text{mean power density at a distance d}}$$

$$\text{Gain G} = \frac{\text{power density at a distance d in the direction of maximum radiation}}{\mathrm{P_T} / 4\pi d^2}$$

- Directional antennas "point" energy in a particular direction
  - Better received signal strength
  - Less interference to other receivers
  - More complex antennas

Wireless channel: attenuation wrt distance

- Let g_T *be* the maximum transmission gain. The received power density in the direction of maximum radiation is given by:

$$F = \frac{P_T g_T}{4\pi d^2} \quad [\mathrm{W/m^2}]$$

- $P_T\, g_T$ is the EIRP (Effective Isotropically Radiated Power) and represents the power at which an isotropic radiator should transmit to reach the same power density of the directional antenna at distance d

Wireless channel: attenuation wrt distance

- The power received by a receiver at distance d from the source, in case of no obstacles and LOS, can be expressed as:

Friis transmission equation

$$P_R = P_T g_T g_R \left(\frac{\lambda}{4\pi d} \right)^2 \frac{\mathbf{1}}{\mathbf{L}}$$

$$A_{eff} = \frac{\lambda^2}{4\pi}$$

- where P_T is the transmitter radiated power, g_T and g_R the gains of the transmitter and receiver antennas, λ is the wavelength (c/f) and d the distance between the transmitter and the receiver. Finally, parameter L>1 accounts for HW losses.

Power units - decibel

- Decibel (dB): expresses according to a logarithmic scale a ratio among powers

$$10\log(P_1 / P_2)$$

Log= base-10 logarithm
P_A = 1 Watt
P_B = 1 milliWatt
30 dB→ PA = three orders of magnitudes higher than P_B

◆ **Gain of an antenna is expressed in dB**

3dB →P1 is twice P2, 10dB→P1 is one order of magnitude higher than P2

20dB→P1 is two orders of magnitude higher than P2

30dB→P1 is three orders of magnitude higher than P2

Decibels - dBm

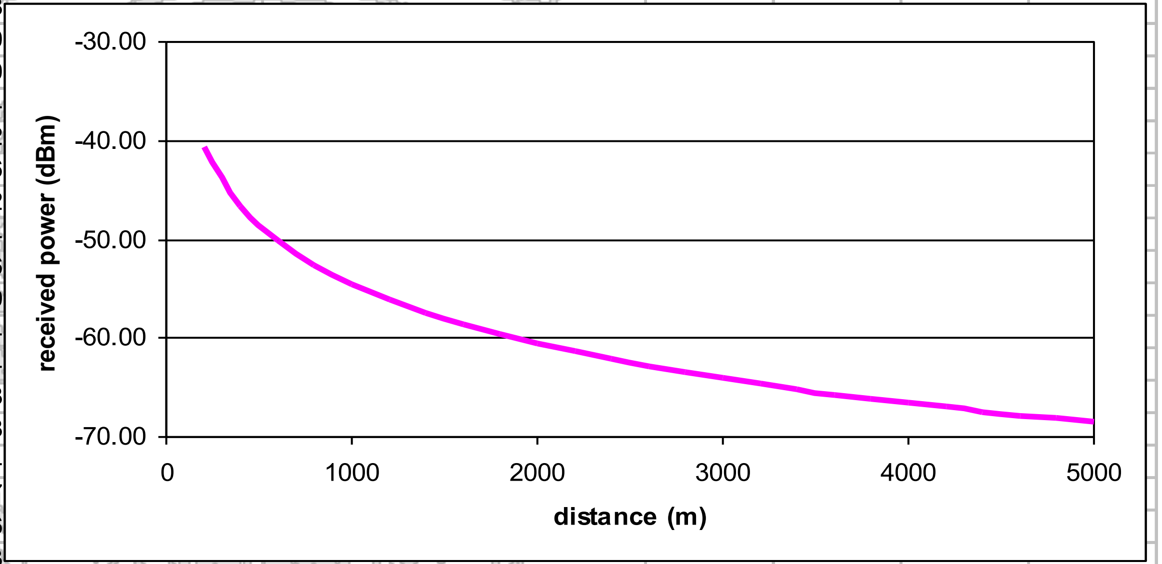
- dBm = ratio between the power and a nominal power of 1mW
  - Power in dBm = 10 log(power/1mW)
  - Power in dBW = 10 log(power/1W)

  Example
  - 10 mW = $10 \log_{10}(0.01/0.001)$ = 10 dBm
  - 10 μW = $10 \log_{10}(0.00001/0.001)$ = -20 dBm
  - S/N ratio = -3dB → S = 1/2 N
- Properties & conversions
  - P(dBm) = $10 \log_{10}$(P (W) / 1 mW) = P (dBW) + 30 dBm
  - (P1 * P2) (dBm) = P1 (dBm) + P2 (dBW)

    P1 * P2 (dBm) = $10 \log_{10}$(P1(W)*P2 (W)/0.001) =
    $10\log_{10}$(P1(W)/0.001) + $10 \log_{10}$P2(W) = P1 (dBm) + P2 (dBW)

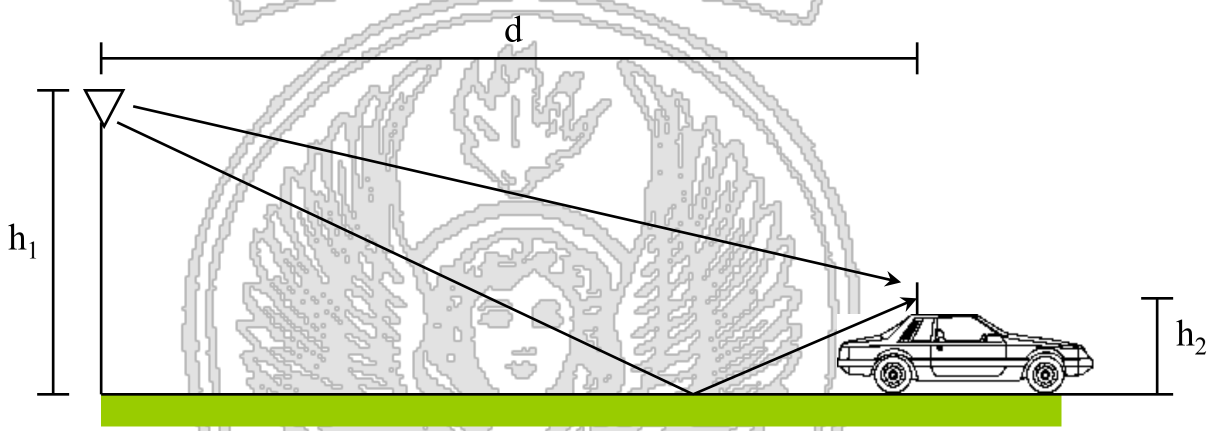
Example

		normalized
frequency [MHz]	900	900000000
speed of light [K	300000	300000000
lambda (m)		0.333333333
gain Tx	1	
Gain Rx	1	
Loss	1	
Ptx [W]	5	
distance (Km)	Prx W	Prx dBm
200	8.80E-08	-40.56
400	2.20E-08	-46.58
600	9.77E-09	-50.10
800	5.50E-09	-52.60
1000	3.52E-09	-54.54
1200	2.44E-09	-56.12
1400	1.79E-09	-57.46
1600	1.37E-09	-58.62
1800	1.09E-09	-59.64
2000	8.80E-10	-60.56
2200	7.27E-10	-61.39
2400	6.11E-10	-62.14
2600	5.20E-10	-62.84
2800	4.49E-10	-63.48
3000	3.91E-10	-64.08
3200	3.44E-10	-64.64
3400	3.04E-10	-65.17
3600	2.71E-10	-65.66
3800	2.44E-10	-66.13
4000	2.20E-10	-66.58
4200	1.99E-10	-67.00
4400	1.82E-10	-67.41
4600	1.66E-10	-67.79
4800	1.53E-10	-68.16
5000	1.41E-10	-68.52



Wireless channel- Two ray propagation model

- In case signal propagates over LOS and one reflected ray..



...the ratio between received power and transmitted power takes the following form:

$$\frac{P_R}{P_T} = g_R g_T \left(\frac{h_1 h_2}{d^2} \right)^2$$

Wireless signal propagation

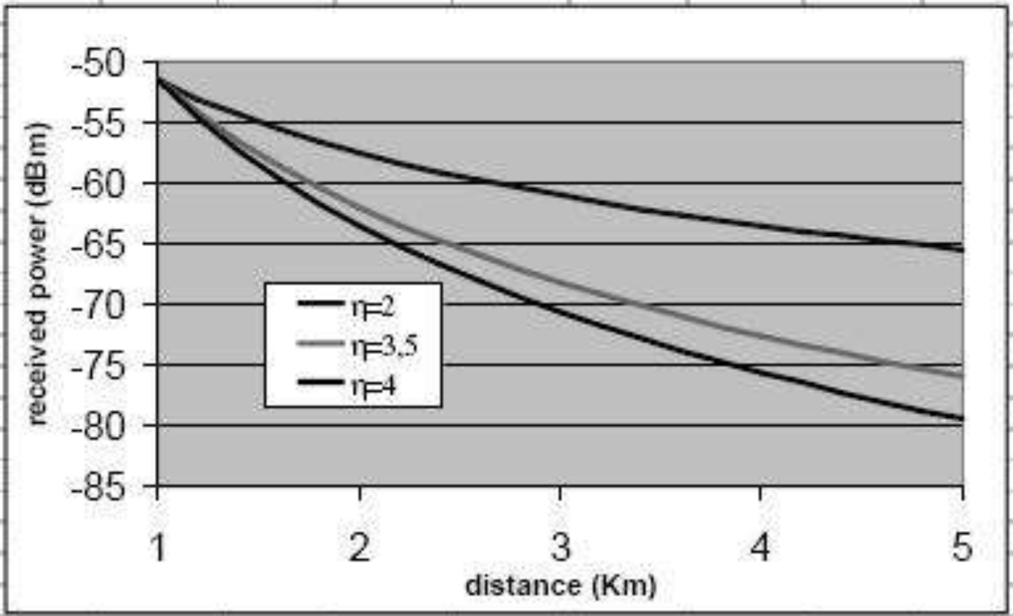
- In the two ray model the received power decreases much faster with distance (~$1/d^4$) than in the free space model (~$1/d^2$)
- Real life signal propagation is much more complex than what represented by the two models
- However, mean received power can be often expressed with a generalization of the Friis transmission equation (where the propagation coefficient is η instead of 2) . The propagation coefficient typically assumes values between 2 and 5 (as determined as a function of the propagation environment by empirical studies and models)

$$P_R = P_T g_T g_R \left(\frac{\lambda}{4\pi} \right)^2 \frac{1}{d^\eta}$$

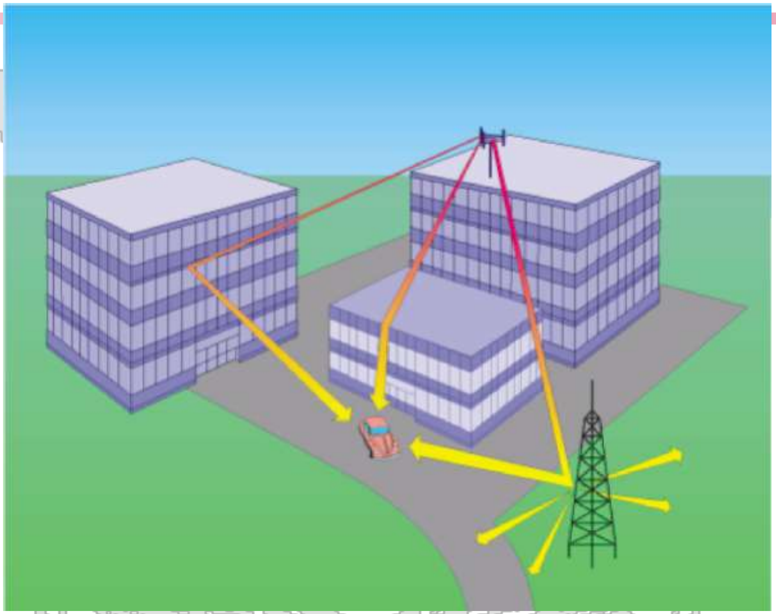
Extended formula

$$P_r(d)(dB) = 10\log_{10} P_r(d_o) + 10\eta\log_{10}\left(\frac{d_o}{d}\right)$$

d_ref	1	Km	
P_ref	-51,5266	dBm	(Ptx=10W; 900 MHz; 1000m)
distance	prx (eta=2	prx (eta=3,5)	prx (eta=4)
1	-51,5266	-51,5266206	-51,5266
1,2	-53,1102	-54,2979642	-54,6939
1,4	-54,4492	-56,6411016	-57,3717
1,6	-55,609	-58,67082	-59,6914
1,8	-56,6321	-60,4611582	-61,7375
2	-57,5472	-62,0626704	-63,5678
2,2	-58,3751	-63,5114144	-65,2235
2,4	-59,1308	-64,834014	-66,7351
2,6	-59,8261	-66,0506877	-68,1256
2,8	-60,4698	-67,1771517	-69,4129
3	-61,069	-68,2258645	-70,6115
3,2	-61,6296	-69,2068698	-71,7326
3,4	-62,1562	-70,1283827	-72,7858
3,6	-62,6527	-70,9972081	-73,7787
3,8	-63,1223	-71,8190464	-74,718
4	-63,5678	-72,5987203	-75,609
4,2	-63,9916	-73,3403457	-76,4566
4,4	-64,3957	-74,0474642	-77,2647
4,6	-64,7818	-74,7231447	-78,0369
4,8	-65,1514	-75,3700639	-78,7763
5	-65,506	-75,9905707	-79,4854

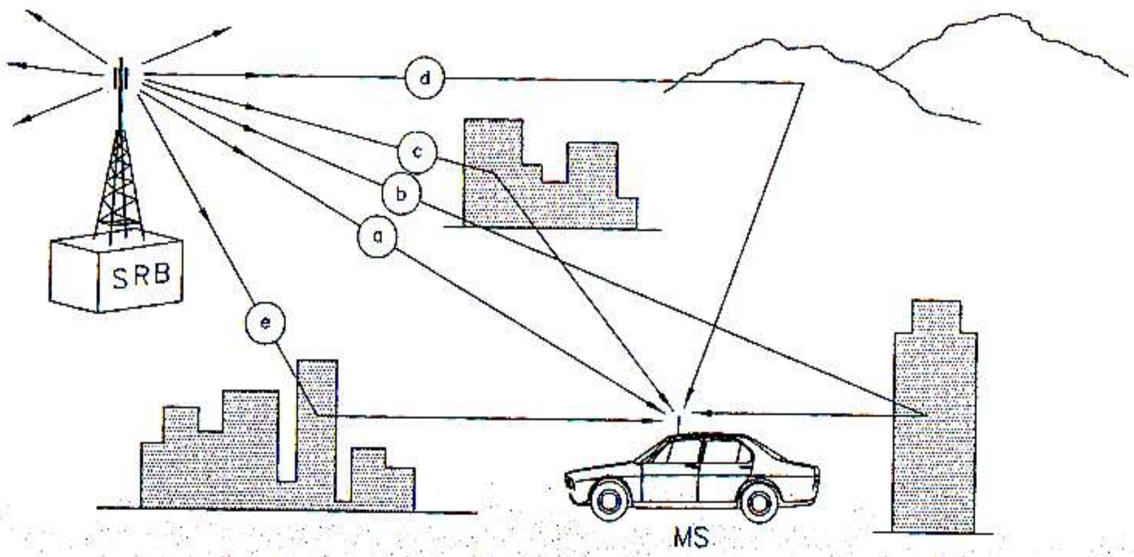


Wireless channel: multipath fading



- While propagating from source to destination the signal can follow multiple paths. At the receiver different components (received over different paths, with different phases and amplitudes) are combined.
- Signal can be reflected, diffracted, scattered based on the obstacles it founds over its path towards destination.
- Low frequencies can traverse without or with low attenuation many objects; when frequency increases waves tend to be absorbed or reflected by obstacles (at very high frequency– over 5 GHz – communication is LOS).

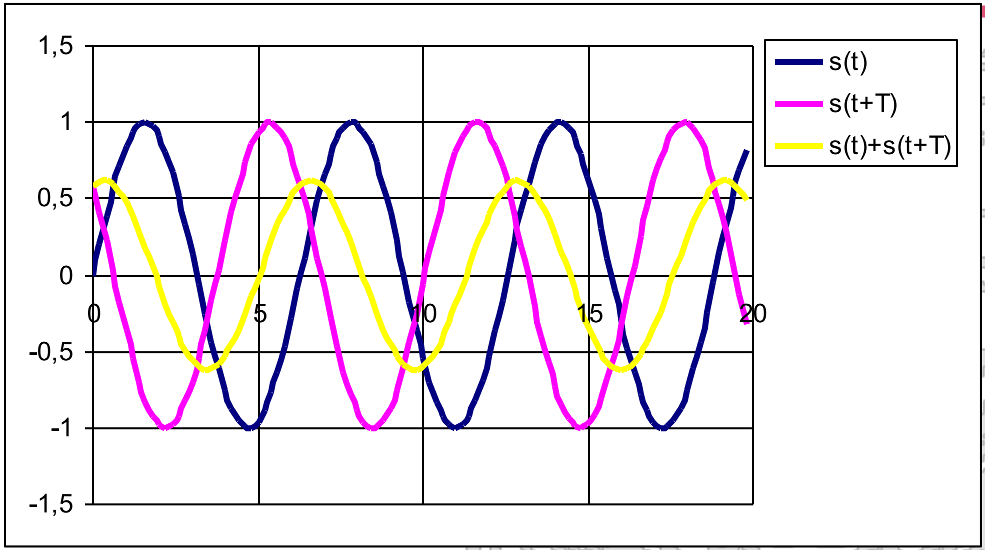
Multipath fading



- Signal replicas received via different propagation paths are combined at the receiver
- The results depends on
  - The number of replicas
  - Their phases
  - Their amplitudes
  - Frequency

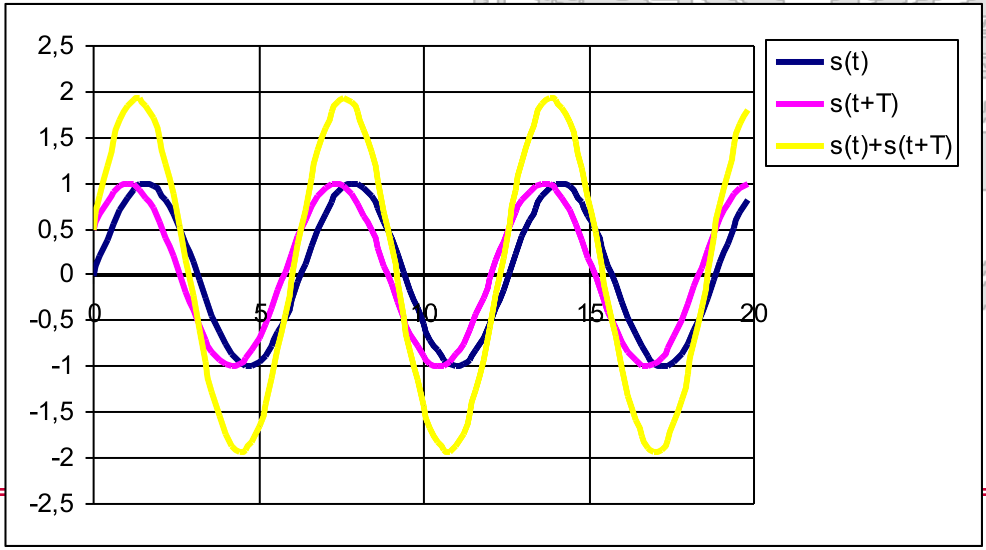
- **Received power differs, as a result**
  - **from place to place, from time to time!**

Multipath fading



- Resulting signal can be attenuated

T=4/5π



- Or amplified

T= π /6

Rayleight fading

$$e_r(t) = \sum_{k=1}^{N} a_k \cos(2\pi f_0 t + \phi_k) =$$

recall that : $\cos(2\pi f_0 t + \phi_k) =$
$= \cos(2\pi f_0 t)\cos(\phi_k) - \sin(2\pi f_0 t)\sin(\phi_k)$

$$= \cos(2\pi f_0 t)\sum_{k=1}^{N} a_k \cos\phi_k - \sin(2\pi f_0 t)\sum_{k=1}^{N} a_k \sin\phi_k =$$
$$= X\cos(2\pi f_0 t) - Y\sin(2\pi f_0 t)$$

In the assumptions:

- N large (many paths)
- ϕ_k uniformly distributed in $(0,2\pi)$
- a_k comparable (no privileged path such as LOS)

X,Y are gaussian, identically distributed random variables

Rayleigh fading power distribution

$$f_p(x) = \frac{1}{2\sigma^2} e^{-x/2\sigma^2}$$

Sigma² is the Variance of The X,Y variables

SAPIENZA
UNIVERSITÀ DI ROMA

Rayleight fading

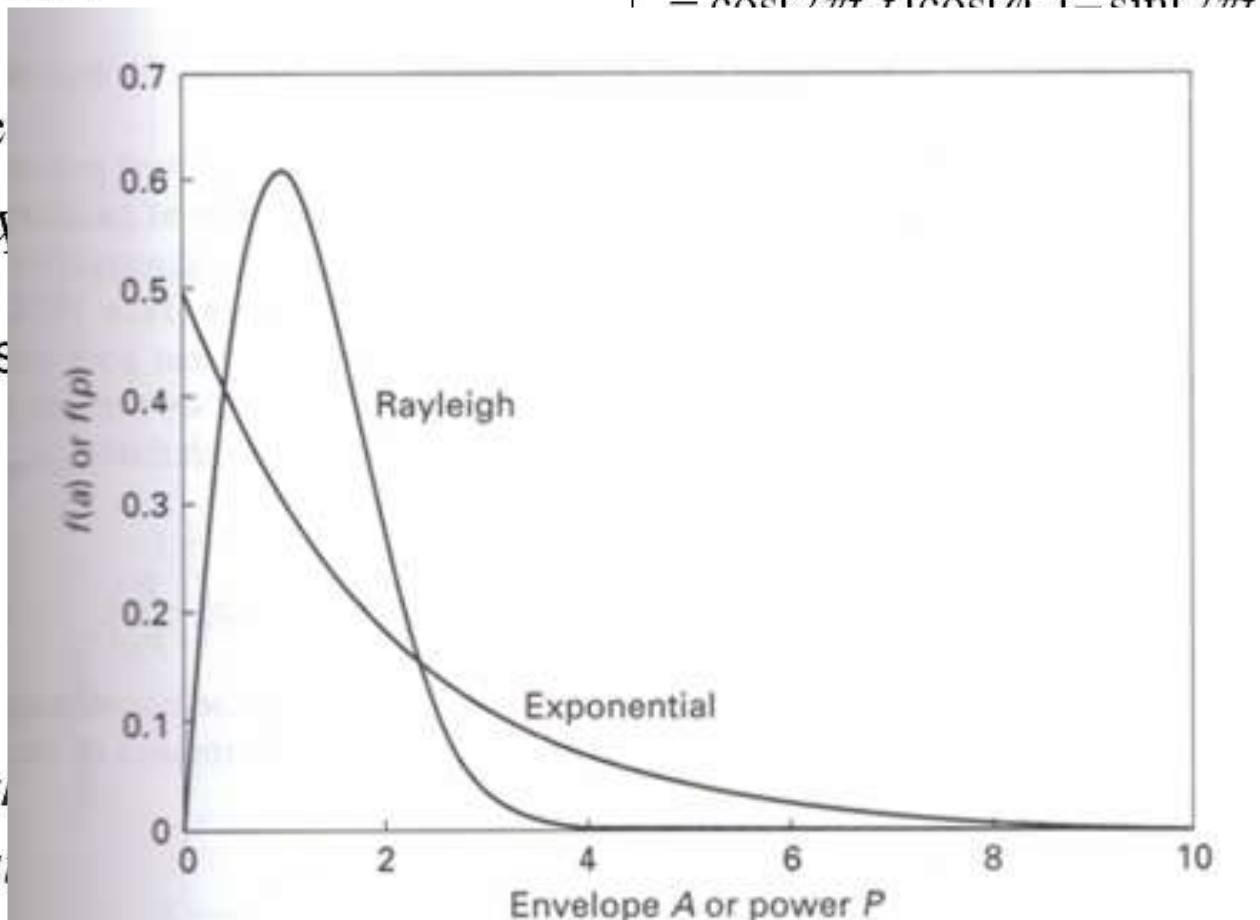
$$e_r(t) = \sum_{k=1}^{N} a_k \cos(2\pi f_0 t + \phi_k) =$$

recall that: $\cos(2\pi f_0 t + \phi_k) =$
$= \cos(2\pi f_0 t)\cos(\phi_k) - \sin(2\pi f_0 t)\sin(\phi_k)$

$= c$

$= X$

In the as



X,Y are

s the
e of
variables

Rayleig
distribu

Rayleight fading

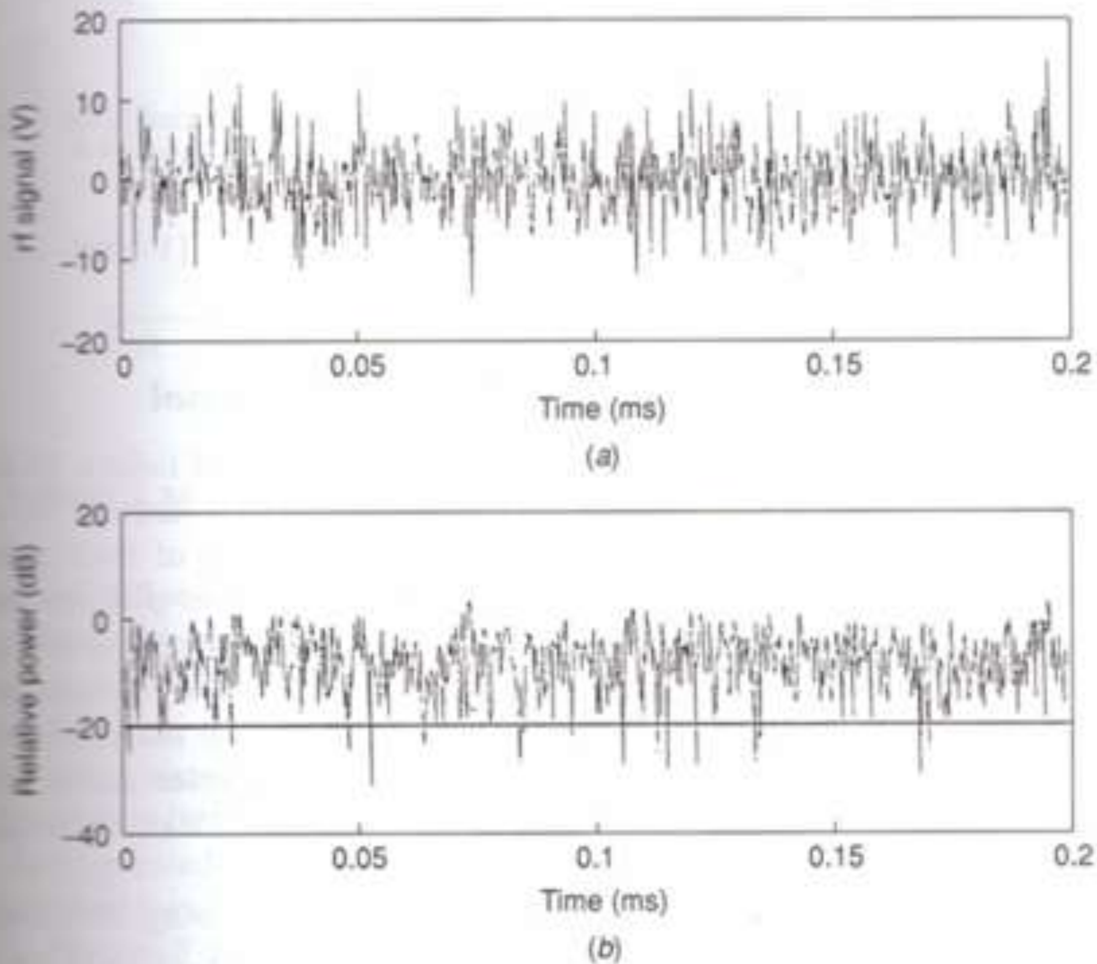


FIGURE 2.19 Rayleigh-faded rf signal (*a*) and its power (*b*). The plots were generated from 11 multiple paths. The envelope was obtained by demodulating the rf signal.

Fading-why is it important?

Answer1:

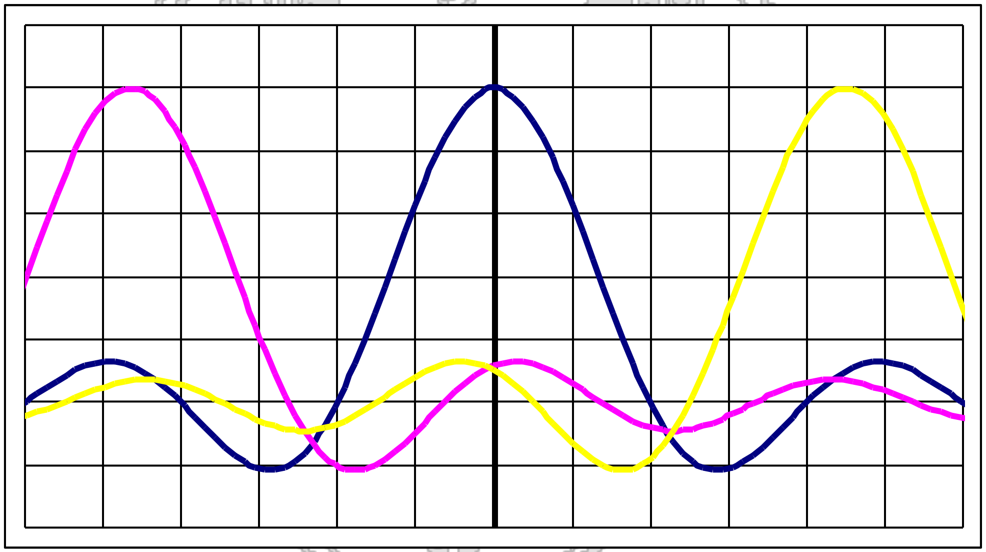
Outage Probability→Probability that received power is lower than a given threshold

⇨Below which signal cannot be correctly received

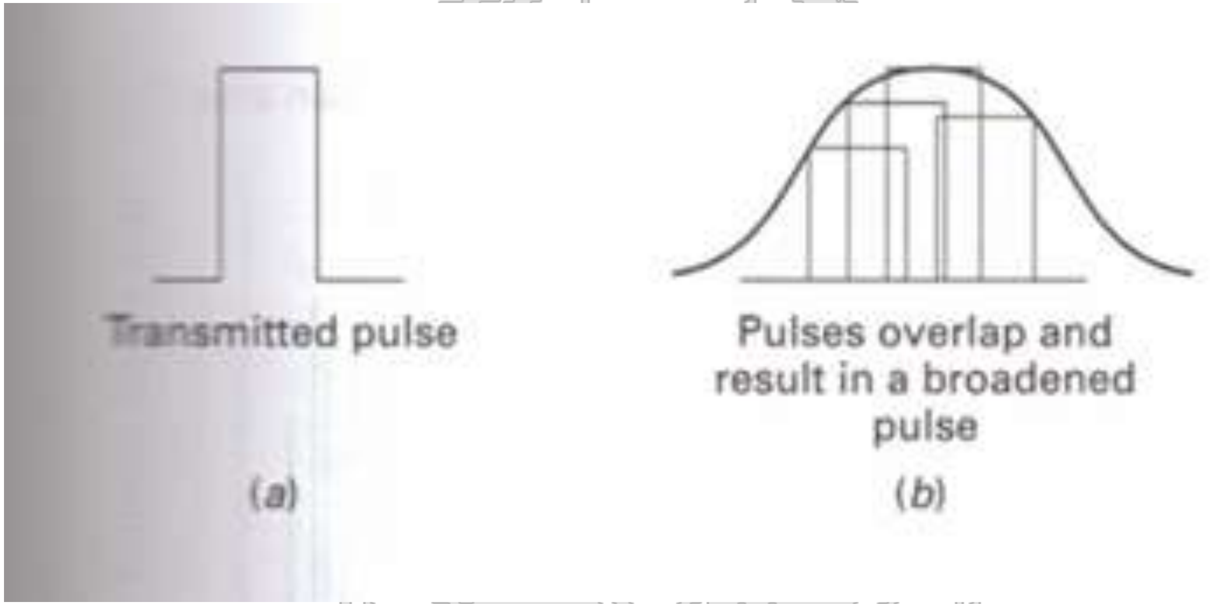
$$P_{out}=\int_{o}^{pthr} f(p)\, dp$$

Multipath fading

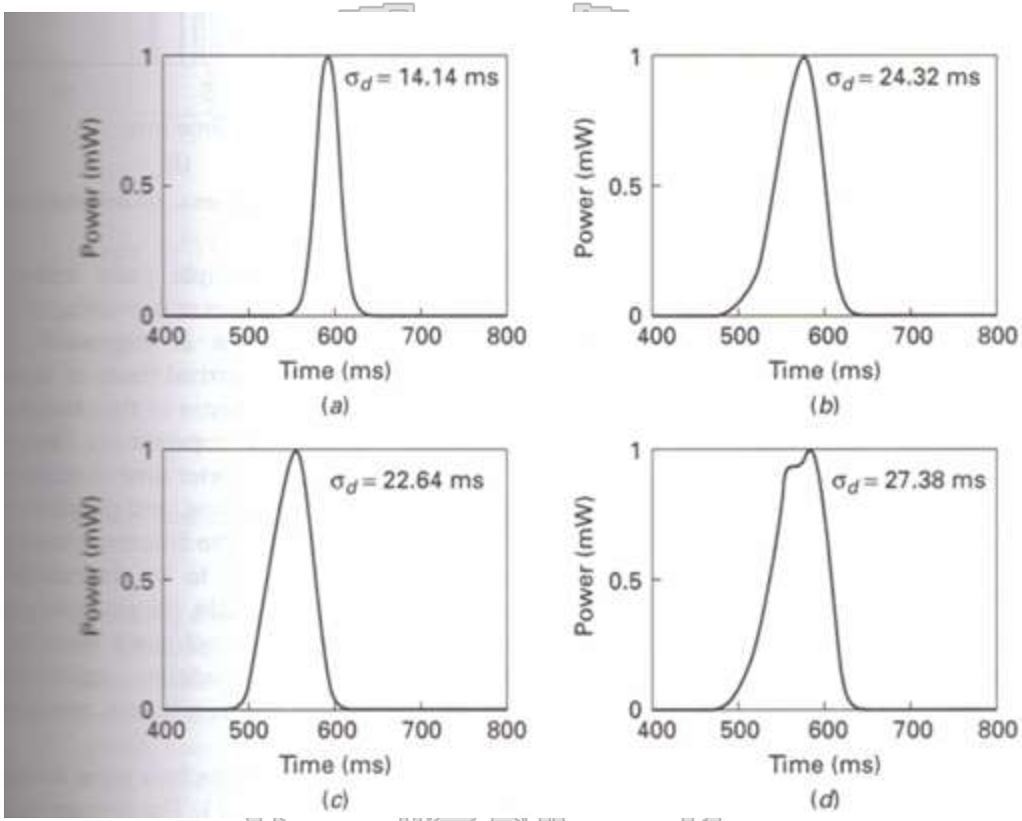
- Different delays experienced by the different signal replicas (*delay spread*) can widen the channel impulse response leading to intersymbol interference (ISI – Inter-Symbol Interference)



Examples

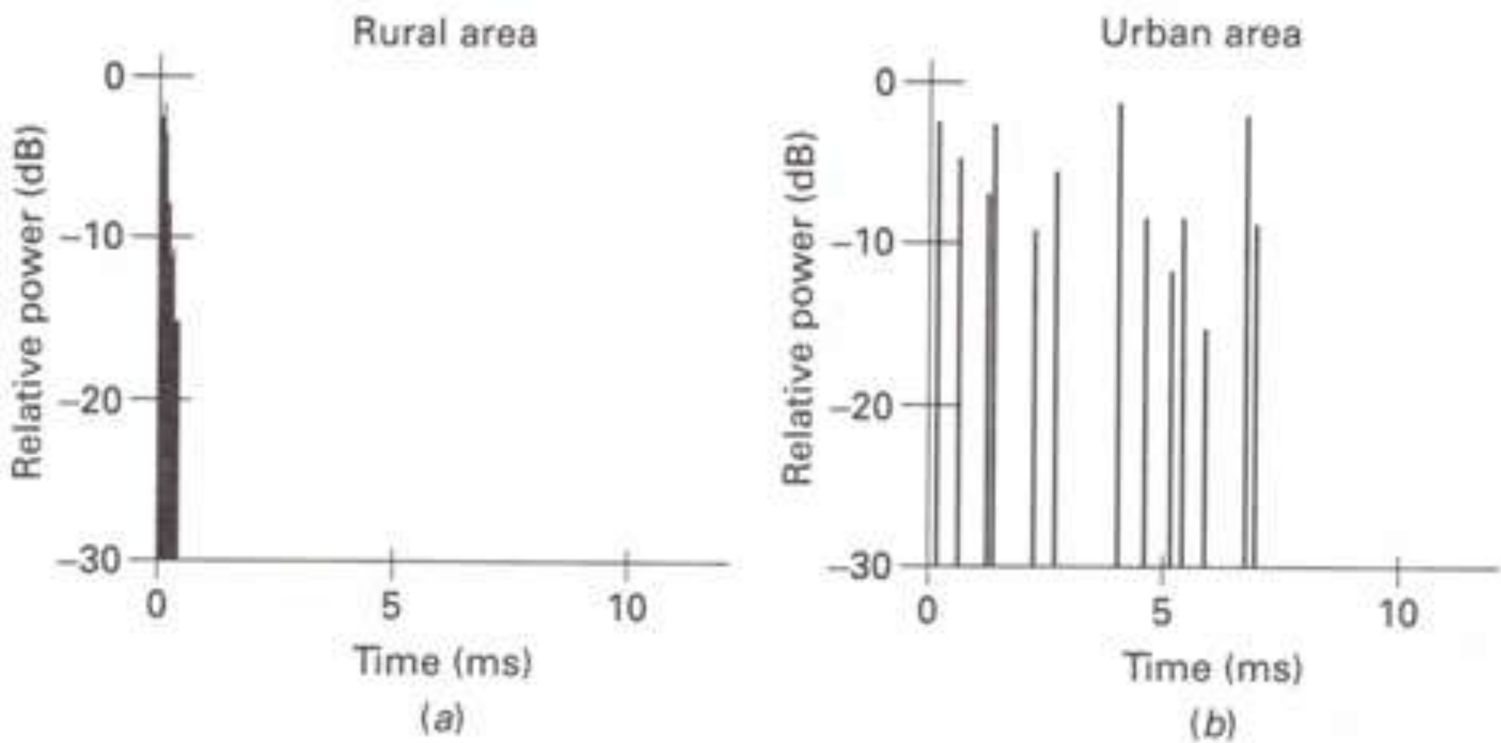

Transmitted pulse
Pulses overlap and result in a broadened pulse
(a)
(b)

Examples



σd = 14.14 ms
σd = 24.32 ms
σd = 22.64 ms
σd = 27.38 ms
Power (mW)
Time (ms)
0
0.5
1
400
500
600
700
800
(a)
(b)
(c)
(d)

Impulse response



Multipath fading

- Impact of delay spread can be quantified by computing the root mean square (RMS Delay Spread):

$$\tau_{RMS} = \sqrt{\frac{1}{\sum_{i=1}^{n} P_i} \sum_{i=1}^{n} \left(\tau_i^2 P_i\right) - \tau_d^2}$$

- **with**

$$\tau_d = \frac{\sum_{i=1}^{n} \left(\tau_i P_i\right)}{\sum_{i=1}^{n} P_i}$$

- **τ_{RMS}** **RMS delay spread**
- **τ_i** **delay on path i**
- **P_i** **power received on path i**
- **n** **number of paths**

Multipath fading

- The coherence bandwidth, which is a statistical measurement of the bandwidth interval over which the channel is 'flat' is approximated by the inverse of the delay spread
- If coherence bandwidth is >> signal bandwidth the channel is flat
- If coherence bandwidth is comparable to the signal bandwidth then delay spread results into intersymbol interference and reception errors

In case of intersymbol interference equalization is used, introducing complexity.

Energy efficient design

IoT, a.a 2017/2018

Prof. Chiara Petrioli

Un. of Rome "La Sapienza"

Chiara Petrioli[†]

[†] *Department of Computer Science – University of Rome "Sapienza" – Italy*

Techniques for energy efficient communications

Energy efficient communication protocols

- Portable devices rely on external sources of energy (batteries, solar cells) to be able to communicate
- Battery lifetime is limited
  - Demanding to recharge battery operated devices after some time- for the device to remain operational;
- Despite improvements in battery technologies the problem has not been solved (and is not expected to be solved by better battery technology only):
  - energy demand is increasing;
  - users expectations in terms of device/network lifetime are increasing;

↓

  - Energy efficient techniques have been developed
  - Energy consumption is a critical metric driving wireless systems design

- Portable devices rely on external sources of energy (batteries, solar cells) to be able to communicate
- Battery lifetime is limited
  - Demanding to recharge battery operated devices after some time for the device to remain operational (finite recharging cycles);
- Despite improvements in battery technologies the problem has not been solved (and is not expected to be solved by better battery technology only):
  - energy demand is increasing;
  - users expectations in terms of device/network lifetime are increasing;

Network lifetime:

- Time till the first node in the network dies having depleted its battery;
- Time before the network gets disconnected or fails to perform critical tasks (e.g., coverage of an Area of Interest)

- Portable devices rely on external sources of energy (batteries, solar cells) to be able to communicate
- Battery lifetime is limited
  - Demanding to recharge battery operated devices after some time for the device to remain operational (finite recharging cycles);
- Despite improvements in battery technologies the problem has not been solved (and is not expected to be solved by better battery technology only):
  - energy demand is increasing;
  - users expectations in terms of device/network lifetime are increasing;

↓

  - Energy efficient techniques have been developed
  - Energy consumption is a critical metric driving wireless systems design

- Portable devices rely on external sources of energy (batteries, solar cells) to be able to communicate

Energy efficiency:

- Expresses how efficiently given tasks are performed.
  - Energy-efficient communication: energy spent by the network per bit correctly delivered to the final destination;
- To be considered in combination with other E2E metrics (throughput/latency)

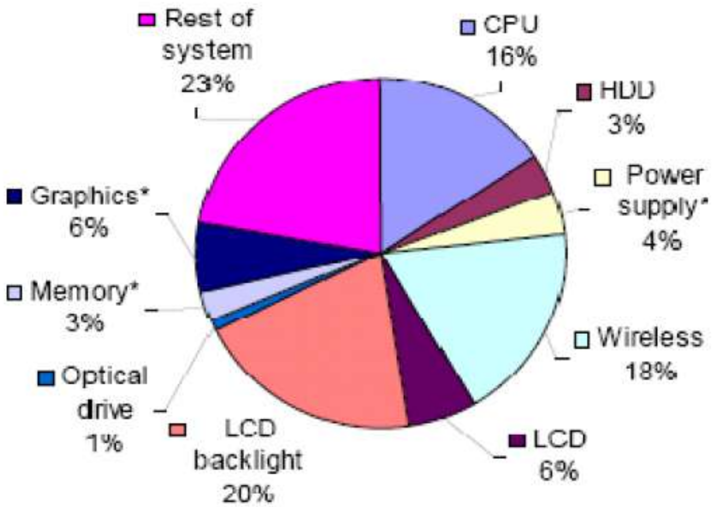
  - Energy efficient techniques have been developed
  - Energy consumption is a critical metric driving wireless systems design

Energy consumption components

- Laptop most energy consuming components include CPU, liquid crystal display (LCD) and **wireless network interface card**
  - Toshiba 410 CDT (2001): 36% of energy consumption due to display, 21% due to CPU+memory, 18% due to wireless network interface card
- **Around mid nineties the area of energy efficient communication moved its first steps…**

Energy consumption components

- Laptop most energy consuming components include CPU, liquid crystal display (LCD) and **wireless network interface card**
  - Toshiba 410 CDT (2001): 36% of energy consumption due to display, 21% due to CPU+memory, 18% due to wireless network interface card
- **Around mid nineties the area of energy efficient communication moved its first steps…**



Somavat, Pavel, Shraddha Jadhav, and Vinod Namboodiri. "Accounting for the energy consumption of personal Computing including portable devices." in *Proceedings of the 1st ACM International Conference on Energy-Efficient Computing and Networking*, 2010.

Energy-efficient techniques

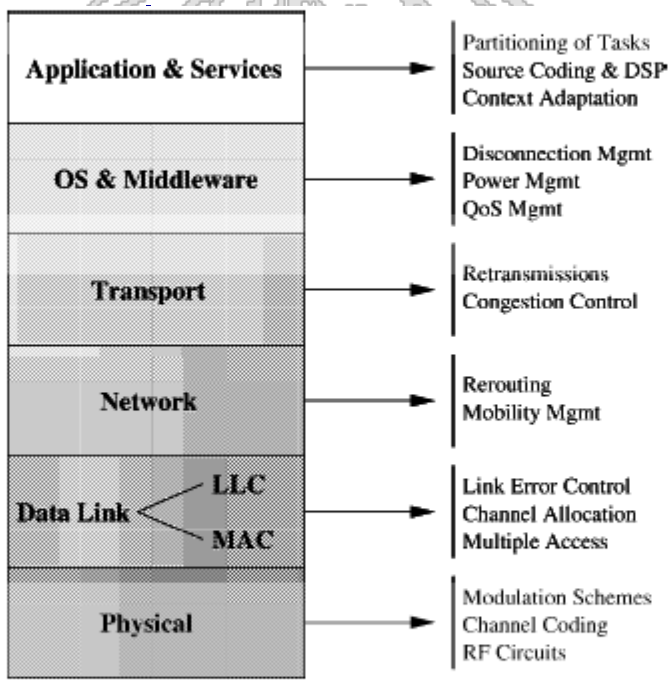
- Network-related energy consumption has two components
  - Computing: in network data processing, data fusion and aggregation, protocol operations;
  - Communications: Wireless transceiver consumes energy either to transmit/ receive data and control packets, or when it is idle, ready ro receive.
- Trade-off between computation and communication
  - Energy-efficient communication protocols can add overhead and computational complexity.
  - There is an inherent computing vs. communication trade-off:
    - ✓ Where should the 'intelligence' of the system be placed? Which data should be processed in network (→higher energy consumption due to computing in nodes which can be energy constrained, but →more compact data transmitted, thus lower energy consumption due to communication) and which data should instead be transmitted to "higher end" devices or computing systems for processing ?(e.g., to the base station, to the sink, or which tasks should be offloaded to the cloud)
    - ✓ Not a one fit all answer.

The objective of the energy efficient communication techniques is to optimized these trade-offs, and the trade-offs amongs different E2E performance metrics (not just energy consumption but also throughput, latency).

Energy-efficient techniques

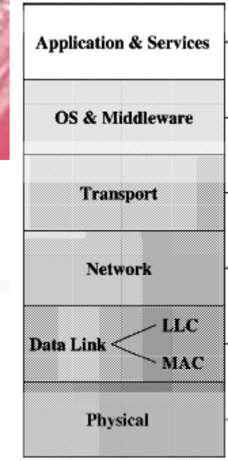
- Network-related energy consumption has two components
  - Computing: in network data processing, data fusion and aggregation, protocol operations;
  - Communication … energy either to transmit/ receive data an… e, ready ro receive.

- Trade-off betwee… tion
  - Energy-efficient co… d computational complexity.
  - There is an inheren…
    - ✓ Where should th… a should be processed in network (→higher energy … n be energy constrained, but →more compact data tra… ommunication) and which dta should instead be transr… for processing (e.g., to the base station, to the si… ?)
    - ✓ Not a one fit all a…



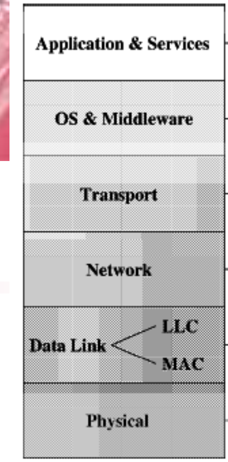
The objective of the er… niques is to optimized these trade-offs, and the trade-offs amongs different E2E performance metrics (not just energy consumption but also throughput, latency).

Energy-efficient techniques

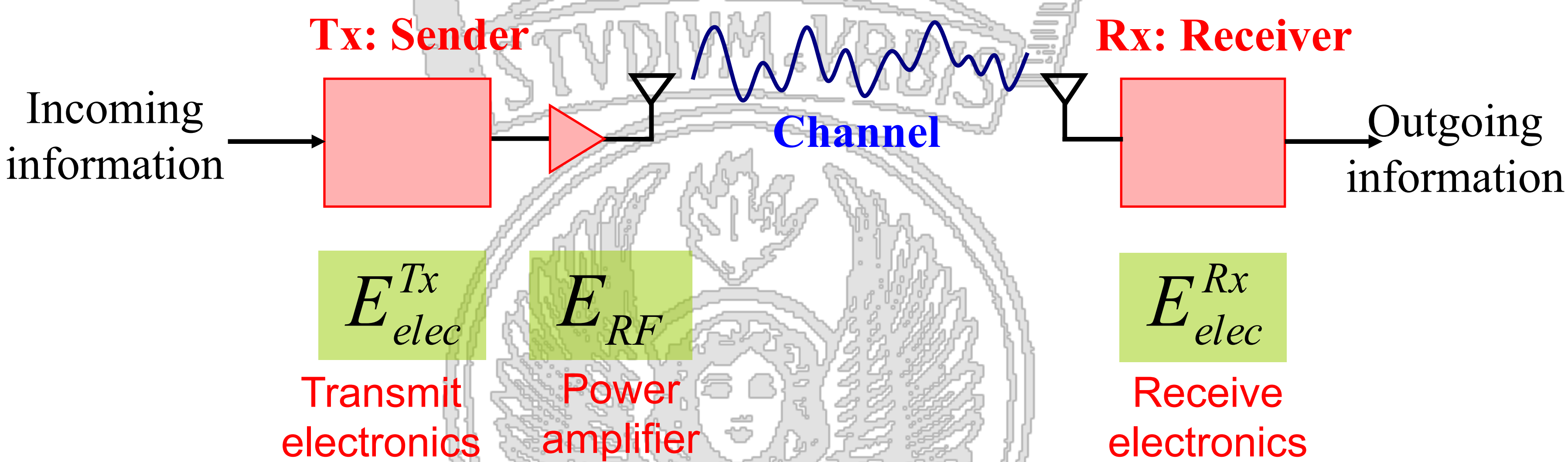


- General guidelines
  - PHY:
    - ✓ Power consumption is a function of the energy needed to activate the transceiver circuitry and of the emitted power → we can significantly decrease overall energy consumption in case of long range communication by applying power control (**Objective: minimizing transmission energy**)
    - ✓ Wireless technologies can dynamically change the modulation scheme used over time. Use of high data rate modulations reduce the time needed to transmit packets, thus the associated transmission energy consumption (**Objective: minimizing transmission energy**)
    - ✓ HW-dependent optimization and selection of HW: due to design choices standard compliant transceivers can have quite different performance in terms of energy consumption, BER and PER (Bit and Packet Error Rates). **HW selection can thus significantly impact the overall system energy consumption**.
    - ✓ Promiscous mode: several protocols proposed for ad hoc network routing exploit the idea of operating the wireless interface card in promiscous mode (→received packets are passed to higher layers and processed even if not addressed to the node) in order to gather information over the wireless broadcast channel which can be used to optimize the protocol operations.
    - ✓ Operating the wireless interface card in promiscous mode forces the interface card to stay in idle (instead of low power modes) for long periods of time, and leads to significant energy consumption due to processing of packets. Therefore, its use typically is a killer in terms of overall energy consumption.

Energy-efficient techniques

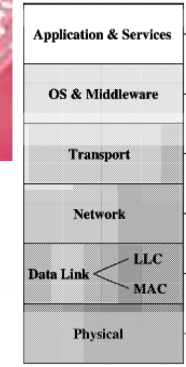


- General guidelines
  - PHY:
    - ✓ Power consumption is a function of the energy needed to activate the transceiver circuitry and of the emitted power → we can significantly decrease overall energy consumption in case of long range communication by applying power control (**Objective: minimizing transmission energy**)
    - ✓ Wireless technologies can dynamically change the modulation scheme used over time. Use of high data rate modulations reduce the time needed to transmit packets, thus the associated transmission energy consumption (**Objective: minimizing transmission energy**)
    - ✓ HW-dependent optimization and selection of HW: due to design choices standard compliant transceivers can have quite different performance in terms of energy consumption, BER and PER (Bit and Packet Error Rates). **HW selection can thus significantly impact the overall system energy consumption**.
    - ✓ Wireless transceiver should instead be switched to a low power 'sleep state' (where it cannot receive or transmit packets but the energy consumption is orders of magnitude lower) whenever a packet not addressed to the node or whenever information exchanged during a handshake make the node aware that the channel will be busy for the next future for transmitting packets not addressed to it
    - ✓ The transceiver should switch to low power mode for the whole time interval when it knows it will not be involved in communications.
      - This is also why destination address is the first field of the header
      - This is also why NAV field is part of RTS/CTS handshake in IEEE 802.11



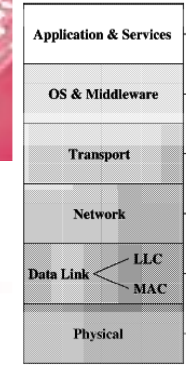
- Wireless communication subsystem consists of three components with substantially different characteristics
- Their relative importance depends on the **transmission range** of the radio

Energy-efficient techniques



- General guidelines
  - MAC
    - ✓ **Awake/asleep schedule:** Nodes alternate between
      - high energy consuming states (awake:transmit/receive/idle) in which the transceiver is ON and packets can be transmitted/received AND
      - states in which the transceiver is OFF, packets cannot be received or transmitted but the energy consumption is much lower.
      - Duty cycle=$T_{ON}/(T_{ON}+T_{OFF})$
      - Two possible classes of protocols:
        - » Synchronous:
        - nodes exchange information to coordinate on when to wake up;
        - periodic control message exchange ensures they know when their neighbors will wake up;
        - a packet is transmitted to a neighbor when it is ON.
        - » Asynchronous:
        - Awake/asleep schedule of neighbors is unknown;
        - No control overhead is needed to keep information updated;
        - To ensure reliable communications a sequence of packets must be sent until the destination node wakes up and answers (overhead when a packet has to be sent)
        - OR nodes must follow a cross-layering approach selecting one neighbors among the awake neighbors as relay.
    - ✓ Nodes not involved in communication should go to sleep till current information exchange completes (**Objective: avoid energy waste**).
    - ✓ Nodes should minimize collisions (**Objective: avoid energy waste**)
    - ✓ Header compression: By transmitting less bits the transceiver is ON for less time (**Objective: reducing transmission energy**)
    - ✓ Limit control information exchanged, aggregate redundant information (**Objective: reducing transmission energy**)

Tends to increase latency

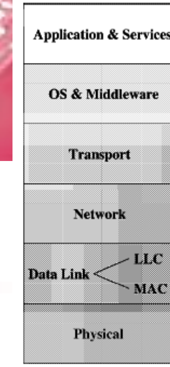


- General guidelines
  - MAC
    - ✓ **Awake/asleep schedule:** Nodes alternate between
      - high energy consuming states (awake:transmit/receive/idle) in which the transceiver is ON and packets can be transmitted/received AND
      - states in which the transceiver is OFF, packets cannot be received or transmitted but the energy consumption is much lower.
      - Duty cycle=$T_{ON}/(T_{ON}+T_{OFF})$
      - Two possible classes of protocols:
        - » Synchronous:
        - nodes exchange information to coordinate on when to wake up;
        - periodic control message exchange ensures they know when their neighbors will wake up;
        - a packet is transmitted to a neighbor when it is ON.
        - » Asynchronous:
        - Awake/asleep schedule of neighbors is unknown;
        - No control overhead is needed to keep information updated;
        - To ensure reliable communications a sequence of packets must be sent until the destination node wakes up and answers (overhead when a packet has to be sent)
        - OR nodes must follow a cross-layering approach selecting one neighbors among the awake neighbors as relay.
    - ✓ Nodes not involved in communication should go to sleep till current information exchange completes (**Objective: avoid energy waste**).
    - ✓ Nodes should minimize collisions (**Objective: avoid energy waste**)
    - ✓ Header compression: By transmitting less bits the transceiver is ON for less time (**Objective: reducing transmission energy**)
    - ✓ Limit control information exchanged, aggregate redundant information (**Objective: reducing transmission energy**)

Tends to increase latency

SAPIENZA UNIVERSITÀ DI ROMA

Energy-efficient techniques



- General guidelines
  - MAC
    - ✓ **Awake/asleep schedule:** Nodes alternate between
      - high energy consuming states (awake:transmit/receive/idle) in which the transceiver is ON and packets can be transmitted/received AND
      - states in which the transceiver is OFF, packets cannot be received or transmitted but the energy consumption is much lower.
      - Duty cycle=$T_{ON}/(T_{ON}+T_{OFF})$
      - Two possible classes of protocols:
        - » Synchronous:
        - nodes exchange information to coordinate on when to wake up;
        - periodic control message exchange ensures they know when their neighbors will wake up;
        - a packet is transmitted to a neighbor when it is ON.
        - » Asynchronous:
        - Awake/asleep schedule of neighbors is unknown;
        - No control overhead is needed to keep information updated;
        - To ensure reliable communications a sequence of packets must be sent until the destination node wakes up and answers (overhead when a packet has to be sent)
        - OR nodes must follow a cross-layering approach selecting one neighbors among the awake neighbors as relay.
    - ✓ Nodes not involved in communication should go to sleep till current information exchange completes (**Objective: avoid energy waste**).
    - ✓ Nodes should minimize collisions (**Objective: avoid energy waste**)
    - ✓ Header compression: By transmitting less bits the transceiver is ON for less time (**Objective: reducing transmission energy**)

Tends to increase latency

Energy consumption due to reception is typically >> than that for transmission, as it is not possible to predict when a packet will have to be received: ← wake-up radio

Transceiver states

- Transceiver can be in one of the following states

tx — ***Awake and transmitting***

rx — ***Awake and receiving***

idle — ***Awake, neither transmitting nor receiving***

asleep — ***Asleep: the transceiver is not operational but energy consumption is low. There can be several asleep states with different subsets of the circuitery switched OFF →different time to switch to such states, but also different energy consumption.***

There is a time and energy consumption associated to the switch whish should be accounted for when designing energy efficient protocols

SAPIENZA
UNIVERSITÀ DI ROMA

Transceiver energy consumption some examples

- Depend on technology, transmission range and phy layer solutions have an impact
- For the same type of device/technology significant changes in the energy consumption have occurred over time
  - As designers are striving to make technology more low power
  - Changes also occurred in the relative weight of different components of energy consumption (which has an impact on design of energy efficient protocols)
- Years: 90'-' 00
  - Message: transmitting costs >> receiving; idle cost costs significantly less, sleep mode 2-3 order of magnitudes lower
  - Examples:
    - ✓ Proxim RangeLAN2 2.4 GHz 1.6 Mbps PCMCIA card: 1.5 W in transmit, 0.75 W in receive, and 0.01 W in standby mode.
    - ✓ Lucent's 15 dBm 2.4 GHz 2 Mbps Wavelan PCMCIA card: 1.82 W in transmit mode, 1.80 W in receive mode, and 0.18 W in standby mode.

Transceiver energy consumption some examples

- Depend on technology, transmission range and phy layer solutions have an impact
- For the same type of device/technology significant changes in the energy consumption have occurred over time
  - As designers are striving to make technology more low power
  - Changes also occurred in the relative weight of different components of energy consumption (which has an impact on design of energy efficient protocols)
- Years: 90'-' 00
  - Message: transmitting costs >> receiving; idle cost costs significantly less, sleep mode 2-3 order of magnitudes lower
  - Examples:
    - ✓ Proxim RangeLAN2 2.4 GHz 1.6 Mbps PCMCIA card: 1.5 W in transmit, 0.75

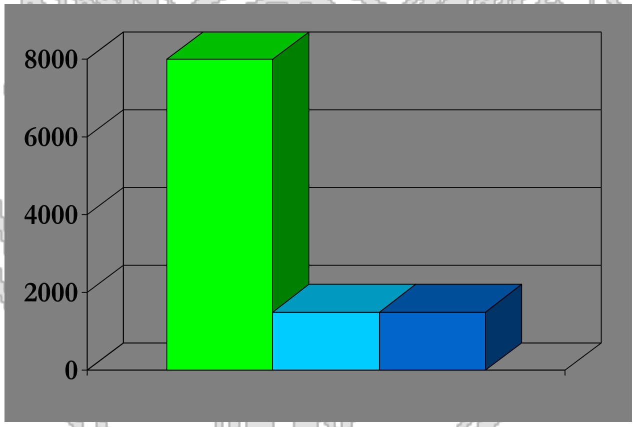
Nowadays for many low to medium range technologies transmission energy consumption is comparable to reception energy consumption and comparable to that of idle mode

Transceivers energy consumption-GSM

- When transmission range increases the percentage of transmitted energy due to emitted power increases

(data source: M. Srivastava, beginning of 2001)

GSM

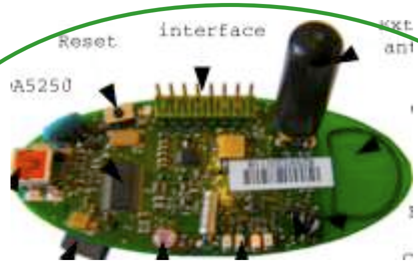


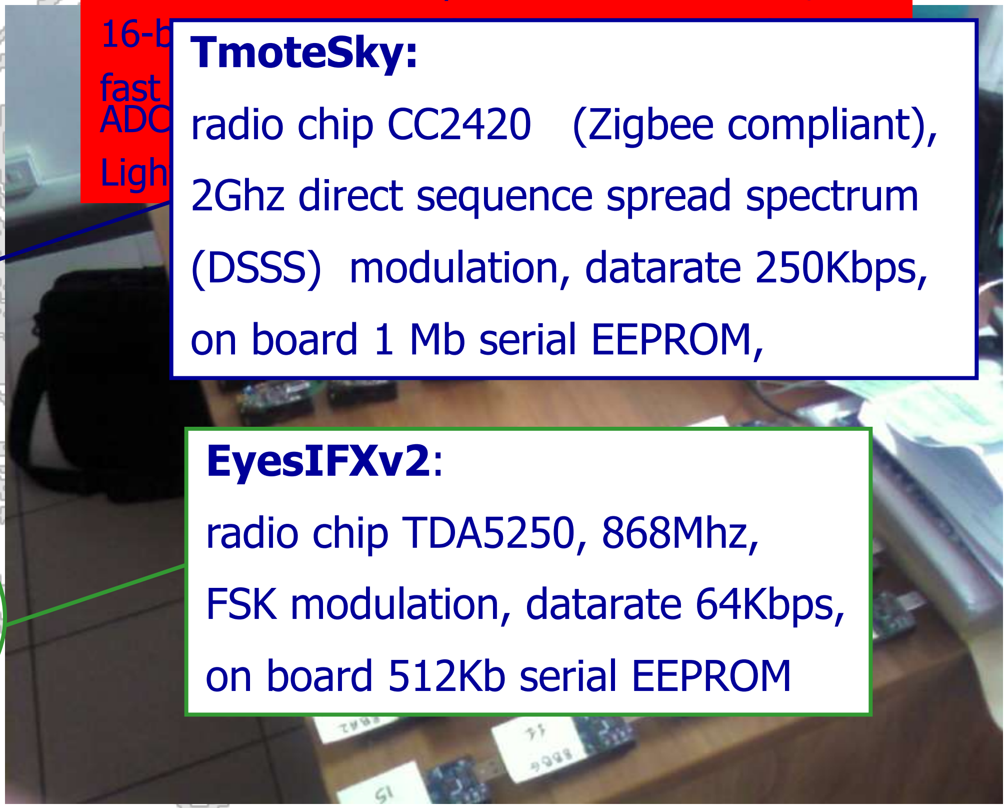
E_{RF} E^{Tx}_{elec} E^{Rx}_{elec}

~ 1 km

Long range communications

- TmoteSky, EYES v2.0 platforms





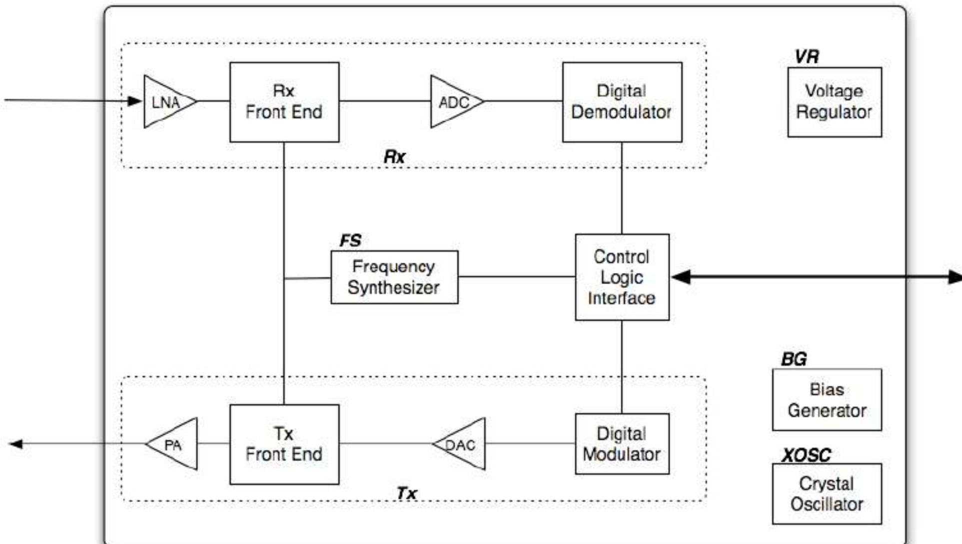
Texas Instruments Mps430 micro-controller,
16-b
fast
ADC
Ligh

TmoteSky:
radio chip CC2420 (Zigbee compliant),
2Ghz direct sequence spread spectrum
(DSSS) modulation, datarate 250Kbps,
on board 1 Mb serial EEPROM,

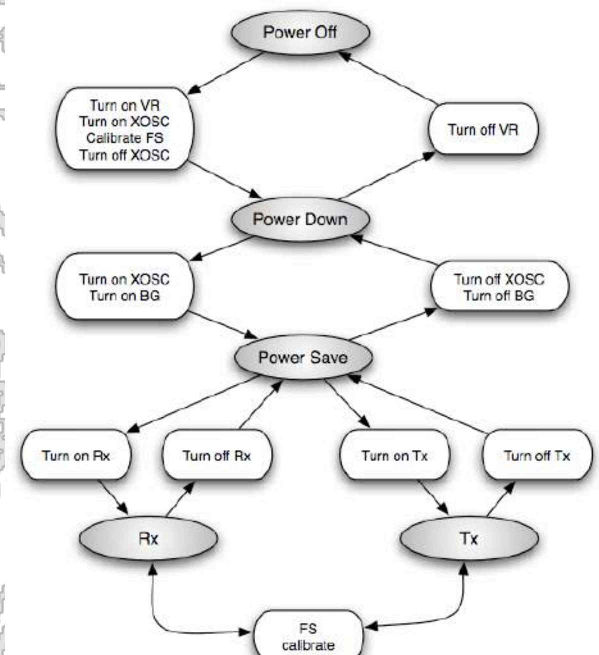
EyesIFXv2:
radio chip TDA5250, 868Mhz,
FSK modulation, datarate 64Kbps,
on board 512Kb serial EEPROM

TmoteSky Energy model

CC2420 Modules



Transceiver states



Energy model

State	Consumption (mA)
Rx	19.7
Tx	17.4
Save	0.45
Down	0.02
Off	0.001

Table 1: Consumption of CC2420 transceiver.

Switch Between (States)	Consumption (mA)	Duration (ms)
Rx/Tx	17.4	0.192
Tx/Rx	19.7	0.192
Tx/Save	17.4	0.192
Rx/Save	19.7	0.192
Save/Down	0.45	0.96
Down/Off	0.02	0.6

Table 2: Switch time of CC2420 transceiver.

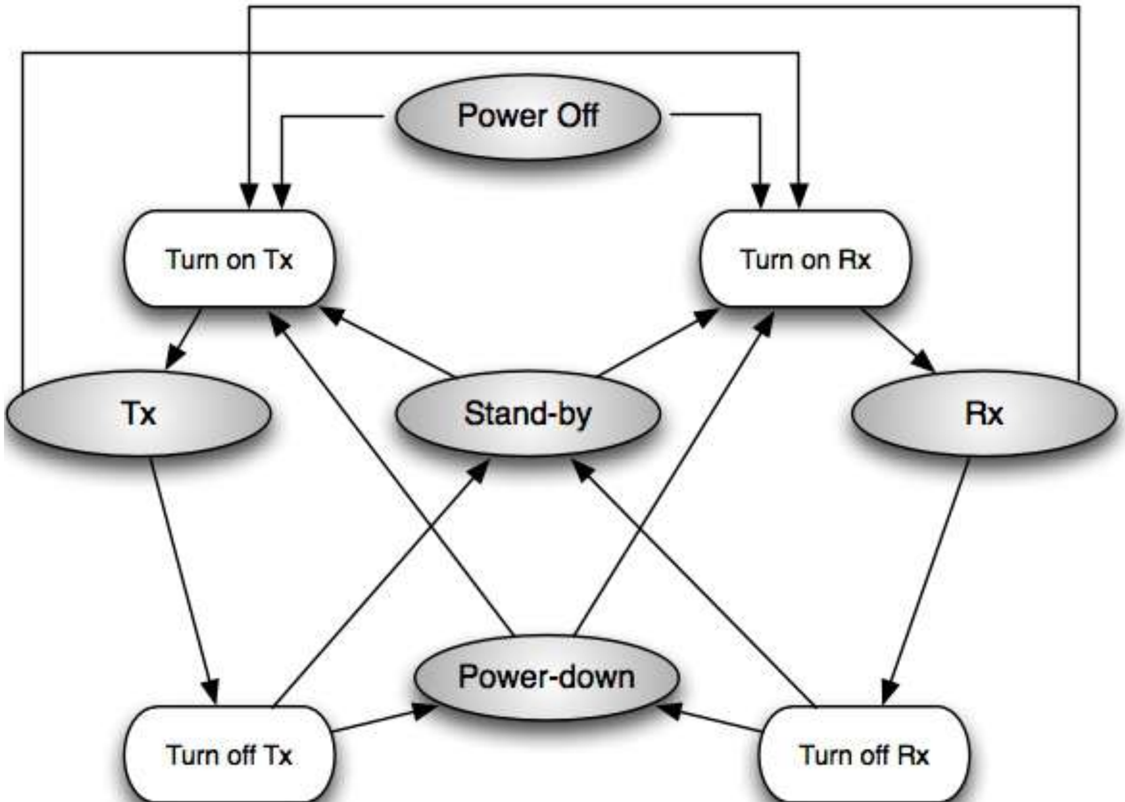
EYES IFXv2 Energy model

Energy model

State	Consumption (mA)
Rx	8.6
Tx	11.2
$Stand-by$	0.75
$Power-down$	0.009
$Power-off$	0

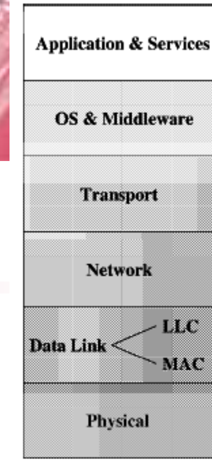
State	Consumption (mA)	Duration (ms)
$Power-off/Tx$	0.3085	9.1
$Power-off/Rx$	0.2632	10.2
Tx/Rx	0.0739	2.2
Rx/Tx	0.0369	1.1
$Power-down/Tx$	0.0025	1.1
$Power-down/Rx$	0.0568	2.2
$Stand-by/Tx$	0.0025	1.1
$Stand-by/Rx$	0.0568	2.2

Transceiver states



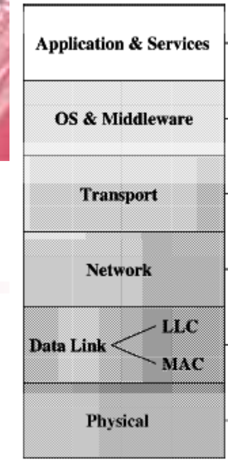
SAPIENZA
UNIVERSITÀ DI ROMA

Energy efficient techniques



- General guidelines
  - Data Link
    - ✓ If channel is in a bad (deep fade) state it is convenient to delay transmissions as it is very unlikely packets will be correctly received (**Objective: avoid waste**)
    - ✓ Energy efficient ARQ and FEC schemes have been studied to optimize energy consumption while ensuring reliable and timely communication (overhead vs. number of retransmissions trade-off; adaptive solutions depending on load, channel, application requirements).

Energy efficient techniques



- General guidelines
  - Routing
    - ✓ Depending on the scenario it can be more energy efficient to transmit over a higher number of shorter links or minimize the number of hops (**can you tell me when one option is better than the other? Suggestion: Long range vs. short range communication**)
    - ✓ Minimize the overhead associated to route discovery and maintenance
    - ✓ Load balancing of the energy consumption among nodes to increase the network lifetime;
    - ✓ Energy aware routing solutions which account for residual energy (and expected future availability of energy in case harvesting is an option) when selecting the best next hop relay.
    - ✓ Link quality aware relay selection to avoid retransmissions.
    - ✓ Relay selection which favors data fusion/aggregation.
    - ✓ All the above combined ← cross layer solutions.

- In the last few years there has been a change of devices used to access the Internet
  - From PC to smartphone
  - Novel Phy layer and more advanced transceiver features
  - How has the energy model been affected by changes in the device technology?
  - Can we still make the same assumptions or are there additional components to account for?

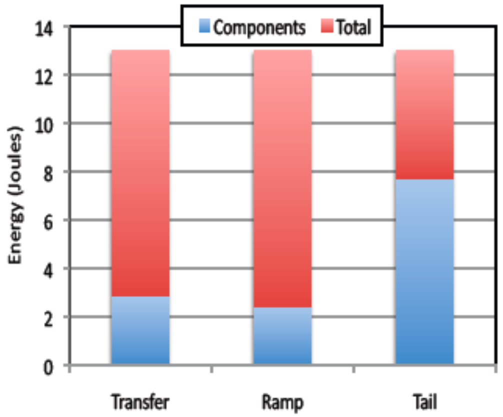
In the following the outcomes of:

N. Balasubramanian, A. Balasubramanian, A. Venkataramani “Energy consumption in mobile phones: A Measurement Study and Implications for Network Applications”, ACM IMC 2009. Observation: Workload impacts energy consumption of typical devices (cellular-GSM/3G; WiFi)

A. Garcia Saavedra, P. Serrano, A. Banchs, G. Bianchi “Energy Consumption Anatomy of 802.11 Devices and Its Implication on Modeling and Design” in Proceedings of Co-NEXT 2012 (on WiFi)

Device and standard-dependent optimizations

- Implementations and choices made for implementing standards make the difference:
  - N. Balasubramanian, A. Balasubramanian, A. Venkataramani "Energy consumption in mobile phones: A Measurement Study and Implications for Network Applications", ACM IMC 2009. Observation: Workload impacts energy consumption of typical devices (cellular-GSM/3G; WiFi)
    - ✓ In 3G, a large fraction (nearly 60%) of the energy, referred to as the *tail energy*, is wasted in high-power states after the completion of a typical transfer.



Typical 3G transfer

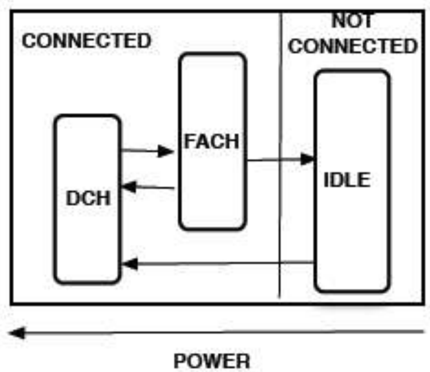
HTTP request issued to a remote server

50KB download

Nokia N95

Device and standard-dependent optimizations

- Implementations and choices made for implementing standards



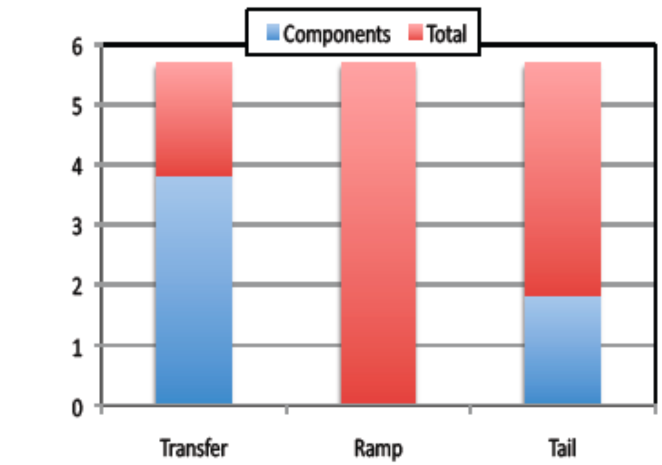
  – N. Balas ... kataramani “Energy ... rement Study and Imp ... IMC 2009. Observa ... ion of typical devices
    - ✓ In 3G, a large fraction (nearly 60%) of the energy, referred to as the *tail energy,* is wasted in high-power states after the completion of a typical transfer.
      - – Switching back from an active state is handled by means of inactivity timers often set to a few seconds.
    - ✓ Tail and ramp energies (more limited) are constants that amortize over larger transfer sizes or frequent successive transfers.

SAPIENZA UNIVERSITÀ DI ROMA

Device and standard-dependent optimizations

- Implementations and choices made for implementing standards make the difference:
  - N. Balasubramanian, A. Balasubramanian, A. Venkataramani "Energy consumption in mobile phones: A Measurement Study and Implications for Network Applications", ACM IMC 2009. Observation: Workload impacts energy consumption of typical devices (cellular-GSM/3G; WiFi)
    - ✓ In 3G, a large fraction (nearly 60%) of the energy, referred to as the *tail energy,* is wasted in high-power states after the completion of a typical transfer.
      - Switching back from an active state is handled by means of inactivity timers often set to a few seconds.
    - ✓ Tail and ramp energies (more limited) are constants that amortize over larger transfer sizes or frequent successive transfers.

Device and standard-dependent optimizations

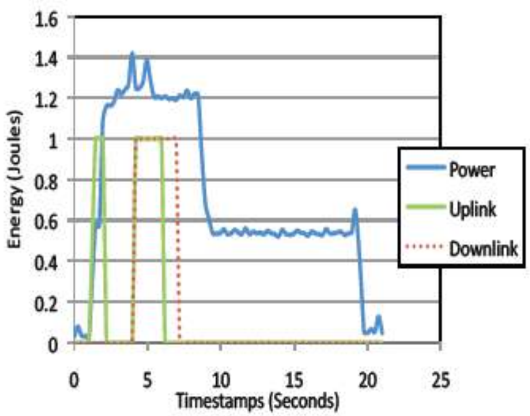


GSM shows a different trend

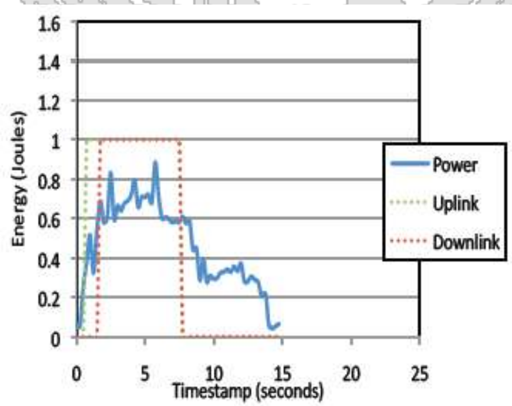
Lower power

More significant transmission energy

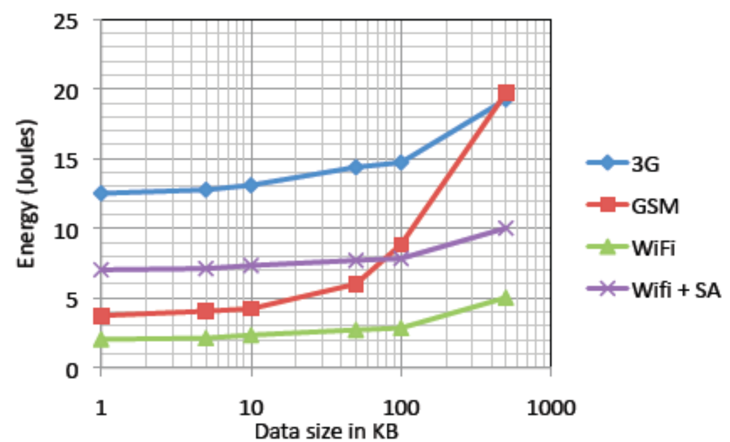
Less significant (even if present) tail energy effect



(a) 3G: Power Profile - 50K



(b) GSM: Power Profile - 50K



Solution:TailEnder

- Three ideas:
  - Combine use of 3G and WiFi (with prediction of WiFi availability)
  - For delay tolerant applications (news, emails) delay transfer if tolerable delay so to transfer batches
  - For web surfing applications design of energy-optimized prefechting techniques

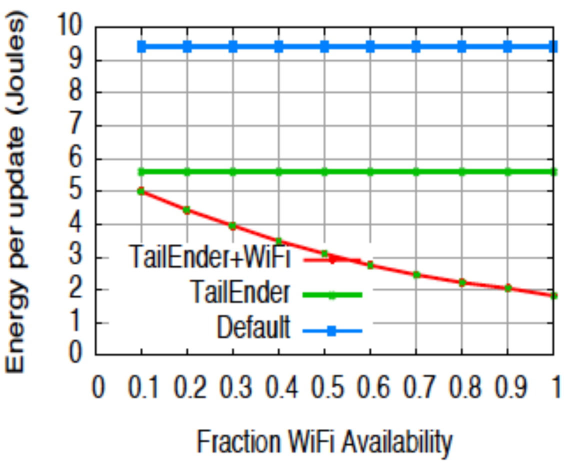


Figure 22: News feed. Average energy improvement when switching between WiFi and 3G networks.

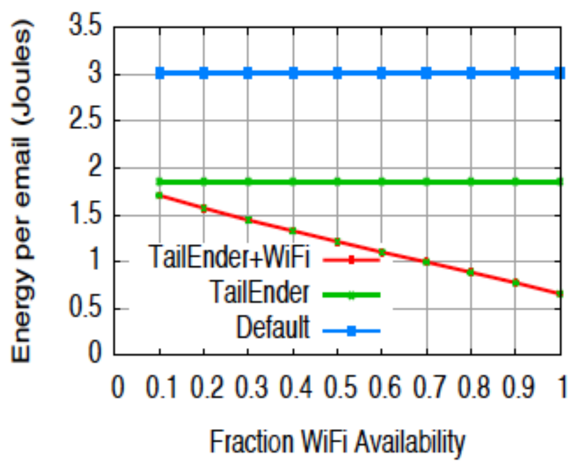


Figure 23: E-mail. Average energy improvement when switching between WiFi and 3G networks.

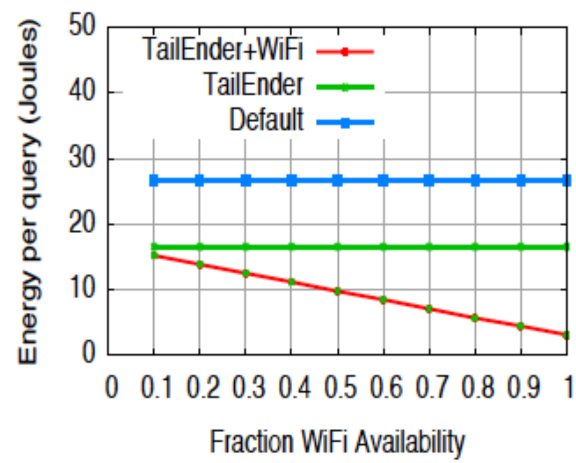


Figure 24: Web Search. Average energy improvement when switching between WiFi and 3G networks.

Energy Consumption Anatomy- IEEE 802.11

- Experiments and measurements on multiple commercial devices
  - Soekris net 4826-48 + Atheros 802.11a/b/g Mini-PCI card, configured to use the 802.11a PHY
  - Alix2d2 + Broadcom BCM4319 802.11b/g Mini-PCI card
  - Linksys WRT54GL + Broadcom BCM4320 802.11b/g Mini-PCI card
- Checking no interference (sniffers)
- Measuring energy consumption with high accuracy power meters
- Controlled traffic generation (mgen generates UDP packets)

A. Garcia Saavedra, P. Serrano, A. Banchs, G. Bianchi "Energy Consumption Anatomy of 802.11 Devices and Its Implication on Modeling and Design" in Proceedings of Co-NEXT 2012

Energy Consumption Anatomy IEEE 802.11

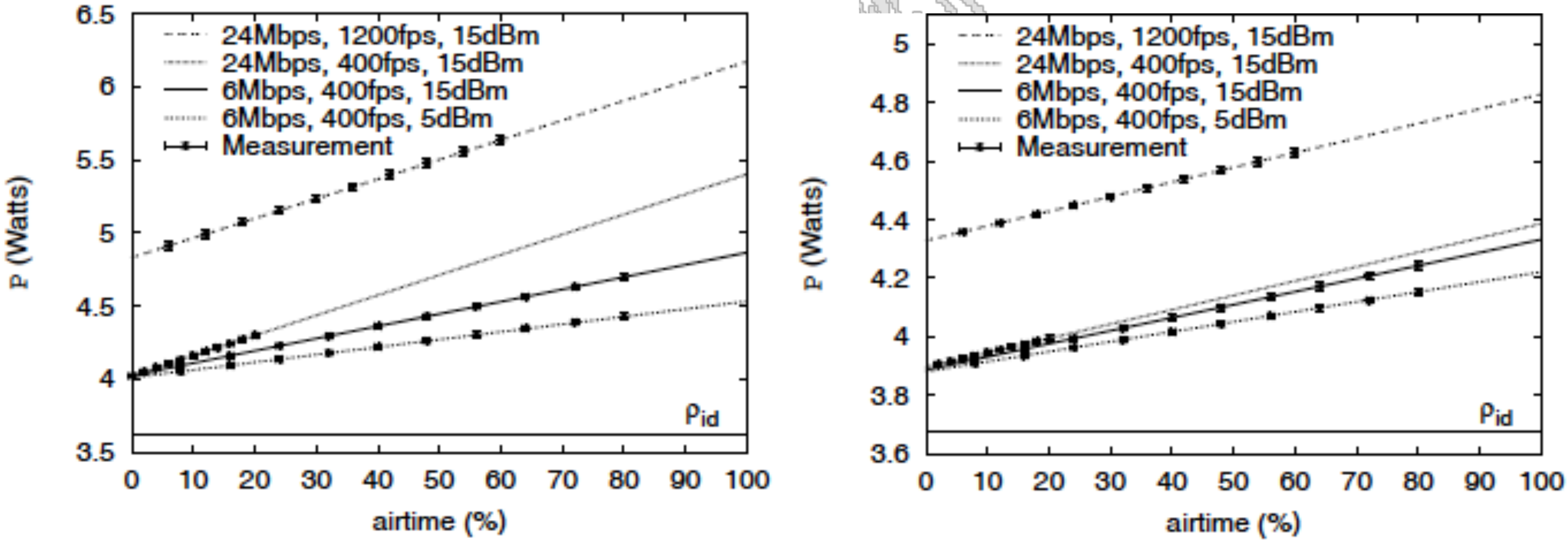
- Baseline energy consumption has been measured

Table 2: Soekris Baseline consumption profile

Config.	Description	Cons. (W)
w/o card	no NIC connected	$2.29 \pm 2.2\%$
WiFi off	NIC connected driver not loaded	$2.58 \pm 2.0\%$ (+0.29)
Idle (ρ_{id})	NIC activated+associated to AP no RX/TX besides beacons	$3.56 \pm 1.7\%$ (+0.98)

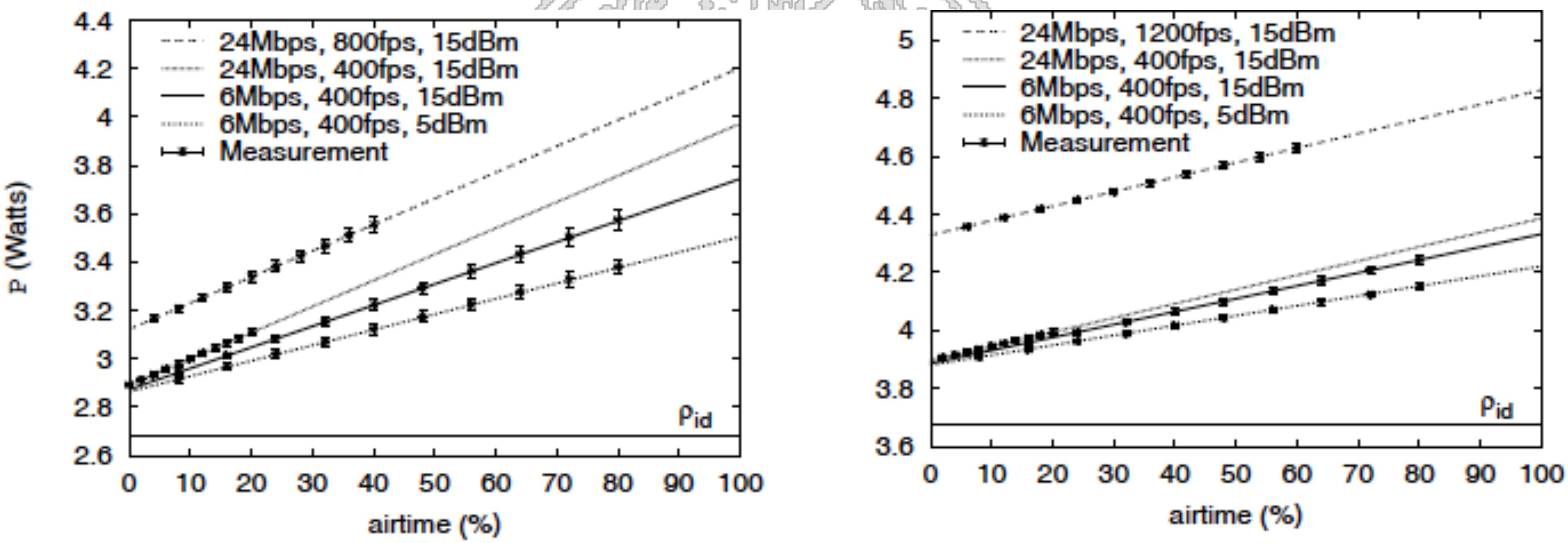
- Eenergy consumption of transmitting one packect without ACKs has been studied
- Impact on energy consumption of varying transmission power, packet length, type of modulation has been quantitatively studied

- Experimental results: Total power consumed by (unacknowledged) transmissions vs. airtime percentage



(a) Soekris (b) Alix

- Experimental results: Total power consumed by (unacknowledged) transmissions vs. airyime percentage



(c) Linksys

(b) Alix

SAPIENZA
UNIVERSITÀ DI ROMA

Energy Consumption IEEE 802.11

- Experimental results: Relationship between cross factor and traffic intensity

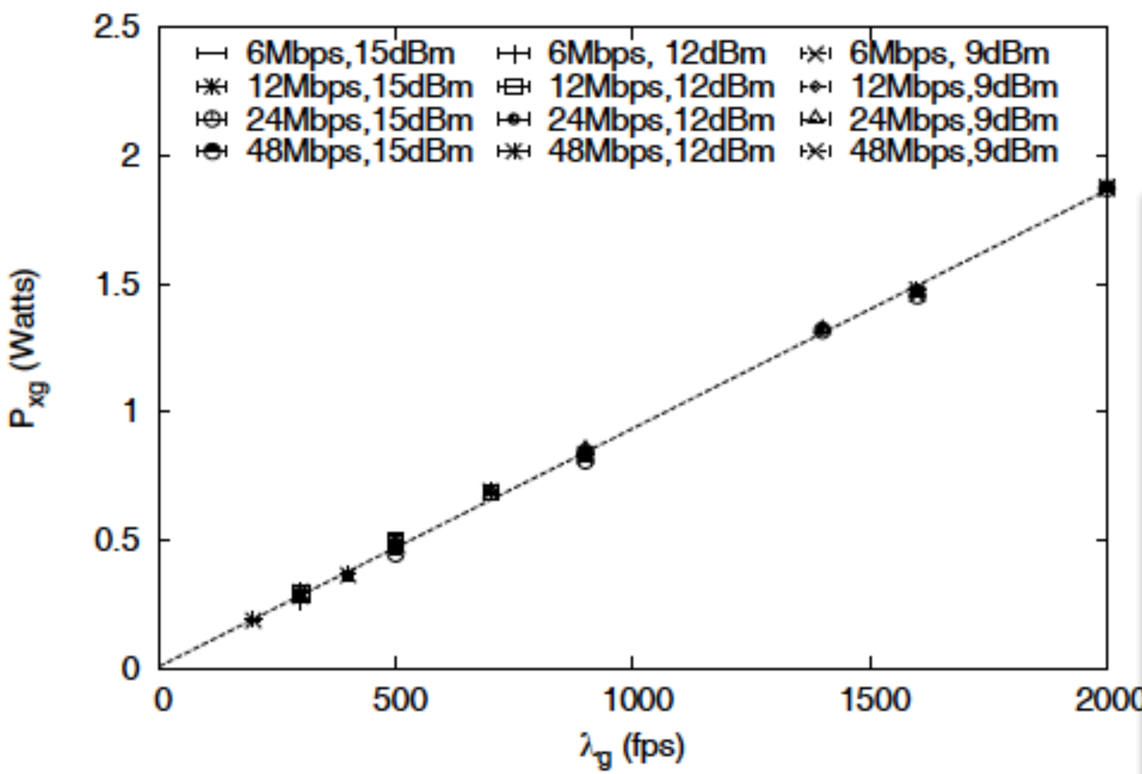


Figure 2: Relation between $P_{xg}(\lambda_g)$ and λ_g.

New energy model

$$P = \rho_{id} + P_{tx} + P_{xg}(\lambda_g)$$

ρ_{id} is the platform specific baseline power consumption

P_{tx} is the power consumption Associated to transmission (depends on airtime, tx power Modulation)

$P_{xg}(\lambda g)$ is the new cross factor

Cross factor analysis

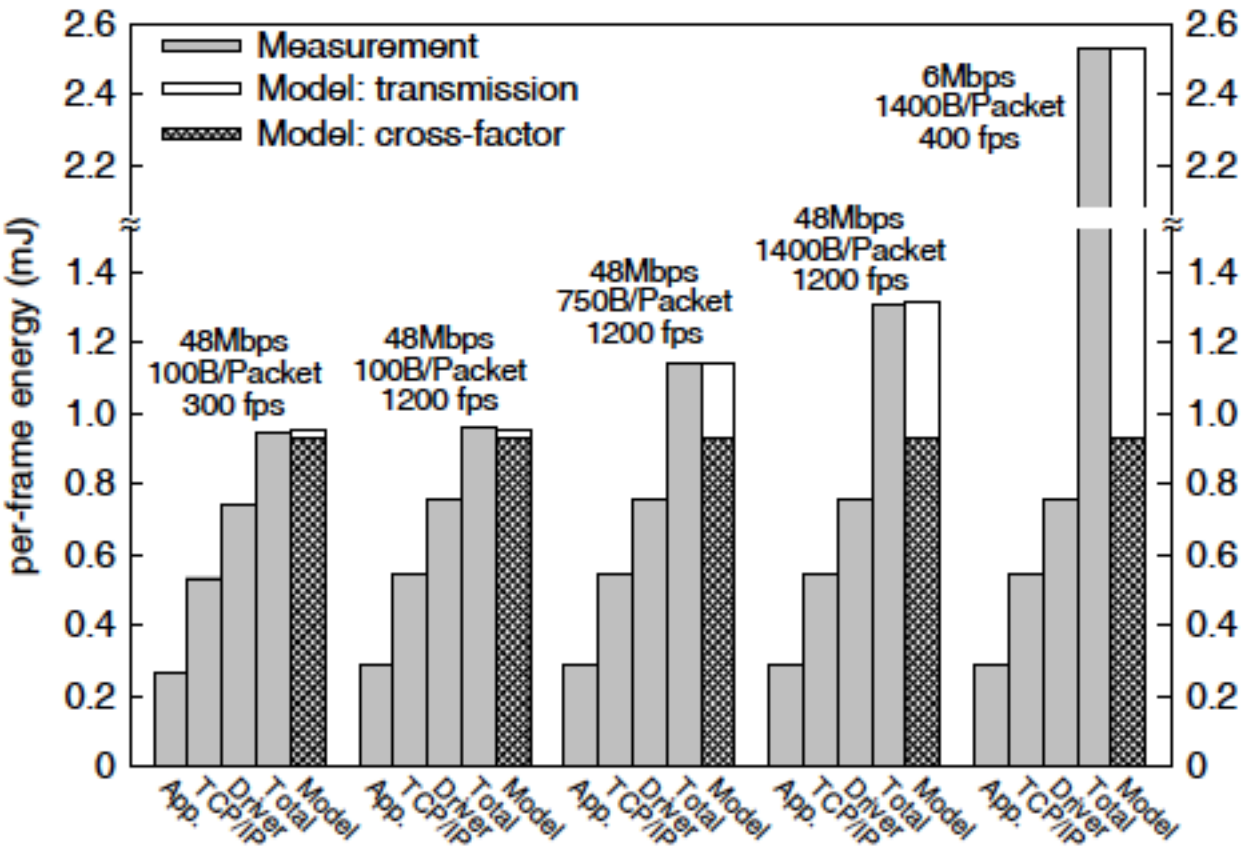


Figure 4: Per-frame energy cost in transmission.

New approaches are proposed for

- Packet relay selection
- Data compression
- Data transmission (back to back)
- Stack implementation

To offload or not to offload?

SAPIENZA UNIVERSITÀ DI ROMA

Number, type & OS	CPU	RAM
7×Samsung Galaxy S Plus (Android 2.3)	1.4 GHz Scorpion	512 MB
2×Samsung Galaxy S (Android 2.3)	1 GHz Cortex-A8	512 MB
1×Samsung Galaxy Note (Android 2.3)	1.4 GHz dual-core Cortex-A9	1 GB
1×Samsung Galaxy Nexus (Android 4.1)	1.2 GHz dual-core Cortex-A9	1 GB

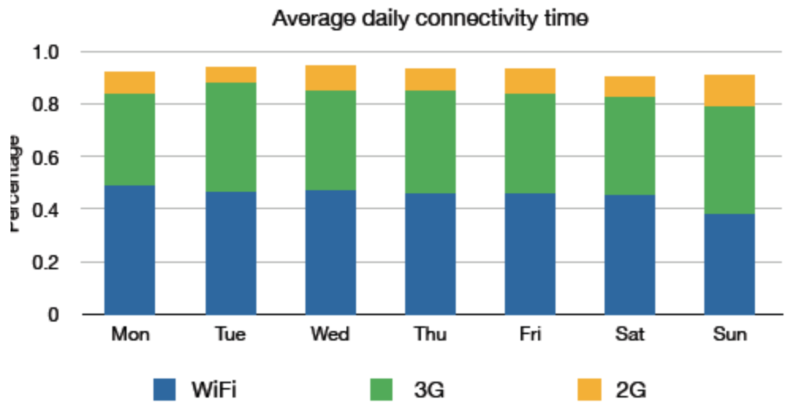
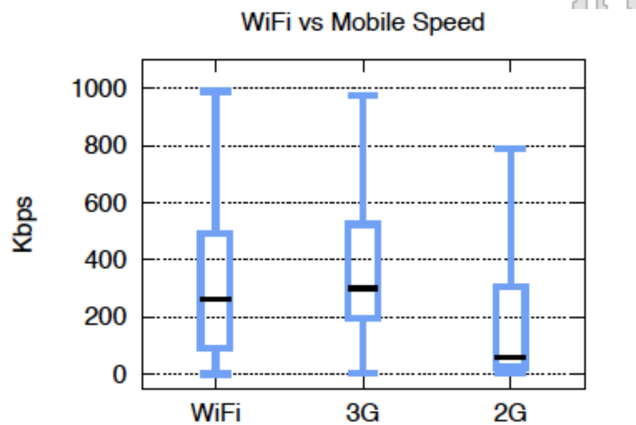
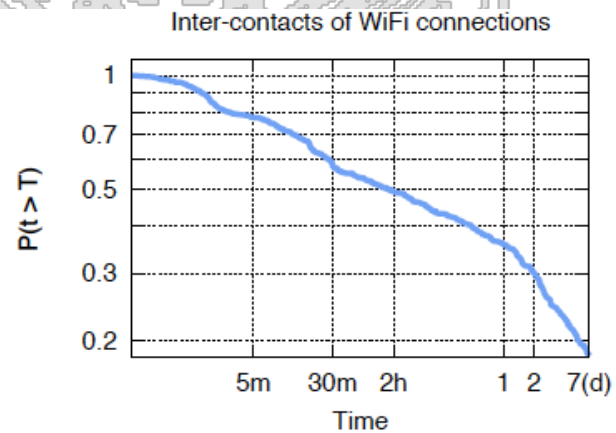


Fig. 1. Average daily connectivity percentage for various technologies.



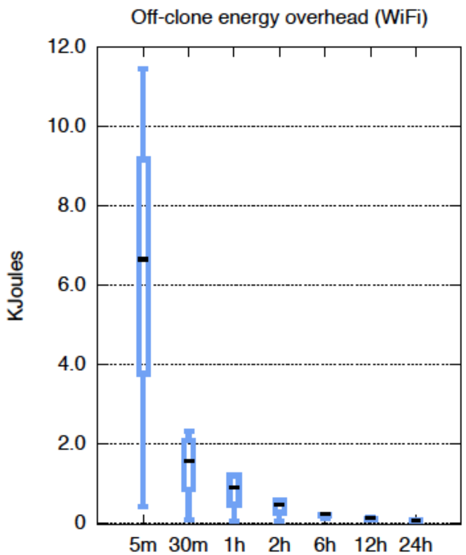
Average (per user) daily upload speed. The graphics lude the minimum and maximum speed value as well he 25^{th}, 50^{th} and 75^{th} quartile.



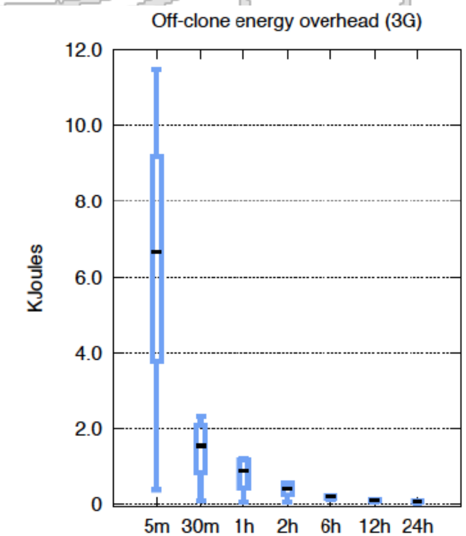
(b) Cumulative distribution of WiFi connection inter-contact times.

M.V. Barbera, S. Kosta, A. Mei, J. Stefa
To offload or not to offload? The bandwidth and Energy costs of mobile cloud computing
IEEE INFOCOM 2013

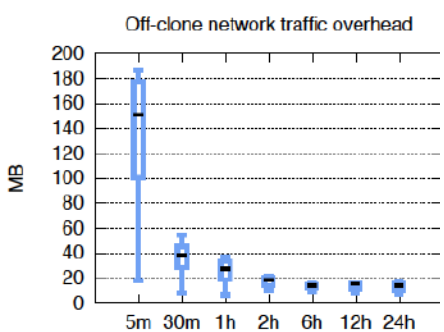
To offload or not to offload



(a) Off-clone energy overhead (WiFi) per day.



(c) Off-clone energy overhead (3G) per day.



(a) Average (per user) off-clone traffic overhead.

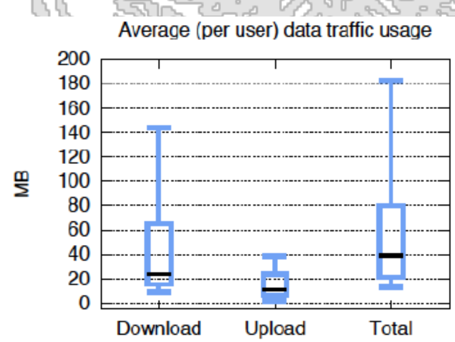


Fig. 4. Average (per user) data traffic sent/received per day.

Comparable energy performance when using WiFi or 3G (file diff computation dominates)

A high overhead is needed to maintain the clone over the cloud synch with the mobile application

Ad hoc Network Routing

IoT, a.a 2017/2018

Prof. Chiara Petrioli

Un. of Rome "La Sapienza"

Chiara Petrioli[†]

[†] *Department of Computer Science – University of Rome "Sapienza" – Italy*

What is an ad hoc network

- A wireless multi-hop infrastructure-less network whose devices act as source/ destination of messages & as relay for packets generated by a node s and addressed to a node z (iff they are on a s-z route)

- Pros: No need for infrastructure → low cost, enables communication where it is usually not needed or it is not viable

- Must be: Self-organizing, self-configuring, self-maintaining

Application scenarios

- Disaster recovery applications
- Military networks
- Personal Area Networks
- Home Networking
- Wireless Sensor Networks (WSNs)
- Inter-vehicular communication
- Mesh Networks (extension of WiFi standard)

Features of ad hoc networks

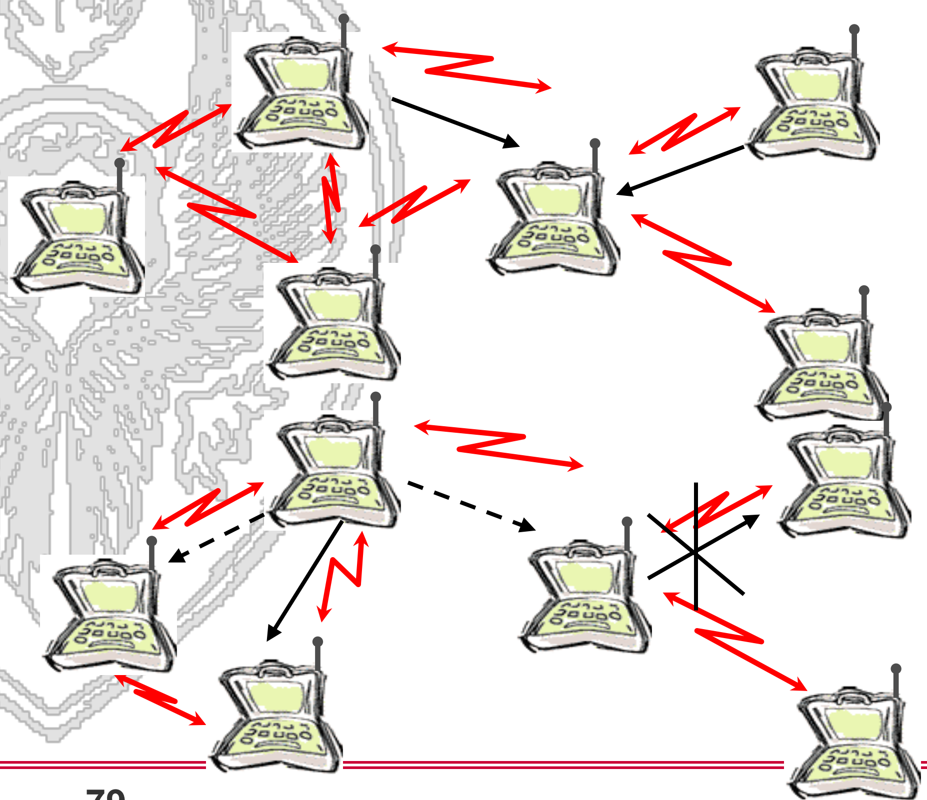
- Highly dynamic networks → device mobility, energy saving sleep/awake modes
- Need for low energy, low overhead, simple protocols
- Traffic:
  - All-pairs in general ad hoc networks
  - Can be low or high (multimedia) traffic
- Scale: Application dependent
  - 10-100 nodes in traditional ad hoc networks

Features of highly resource constrained ad hoc networks (WSNs)

- Highly dynamic networks → due to device mobility (only in some specific applications), to the fact the active node set changes in time for sake of energy saving (always to be considered)
- Need to design low energy protocols → very critical, energy consumption a real bottleneck, memory also a bottleneck
- Traffic from sensors to sink(s)
  - Even if opportunistic communication is also increasingly considered
- Scalability is a major issue (could go up to 1000 or 10000 nodes)
- Code must be simple (small storage capability, very simple, inexpensive, resource constrained devices)
- First solutions we will see for traditional ad hoc networks do not scale to high numbers and are not energy-saving, and are typically too complex for resource constrained embedded systems devices

Medium Access Control in Ad Hoc Networks

- CSMA-like approach, e.g. CSMA/CA
  - Why not TDMA like?
- Why not CSMA/CD? Nodes cannot received and transmit simultaneously
  - Hidden terminal
  - Exposed terminal



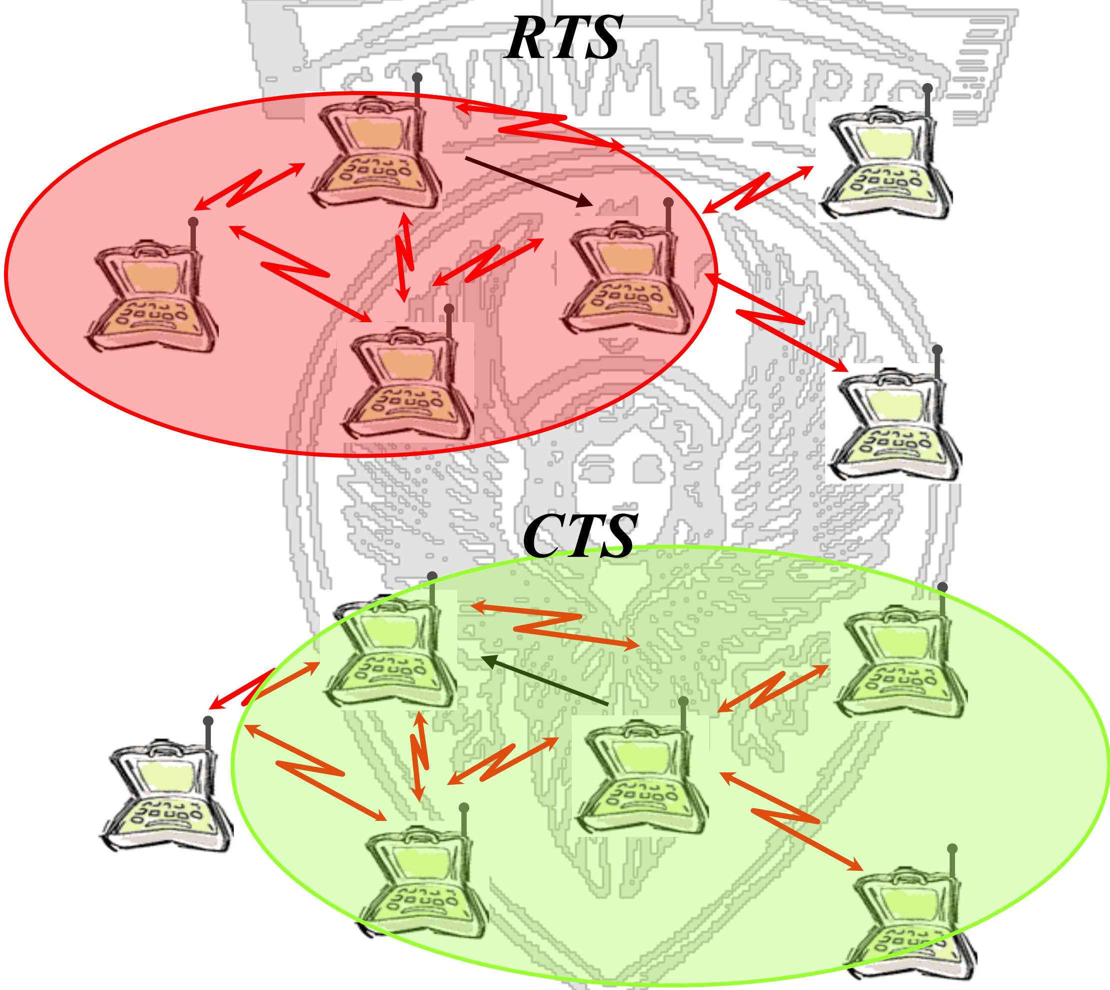
802.11 Distributed Coordination Function

- Based on CSMA/CA
- Before transmitting a frame the sender node x performs carrier sensing
- If the channel is free for a time called Distributed InterFrame Space (DIFS) node x transmits the packet
- Otherwise (being the channel in use) node x waits for the end of current transmission + a random backoff time.
  - The backoff timer count down is frozen whenever the node senses the channel busy (only when the channel is free for a DIFS the counter value is decreased) WHY?
  - When the backoff timer value is zero node x transmits the packet.
  - The backoff timer value is randomly picked within a window interval of CW slots. At the first transmission attempt CW is set to the minimum value (16 slots). An exponential backoff is used. At each retranmission attemp CW is doubled till a maximum value equal to 1024 slots.
- How can node x decide whether the transmitted packet has been successfully received? By means of an explicit ACK the receiving nodes transmits (if the packet is correctly received after a Short InterFrame Space (SIFS) time, <u>SIFS<DIFS)</u>

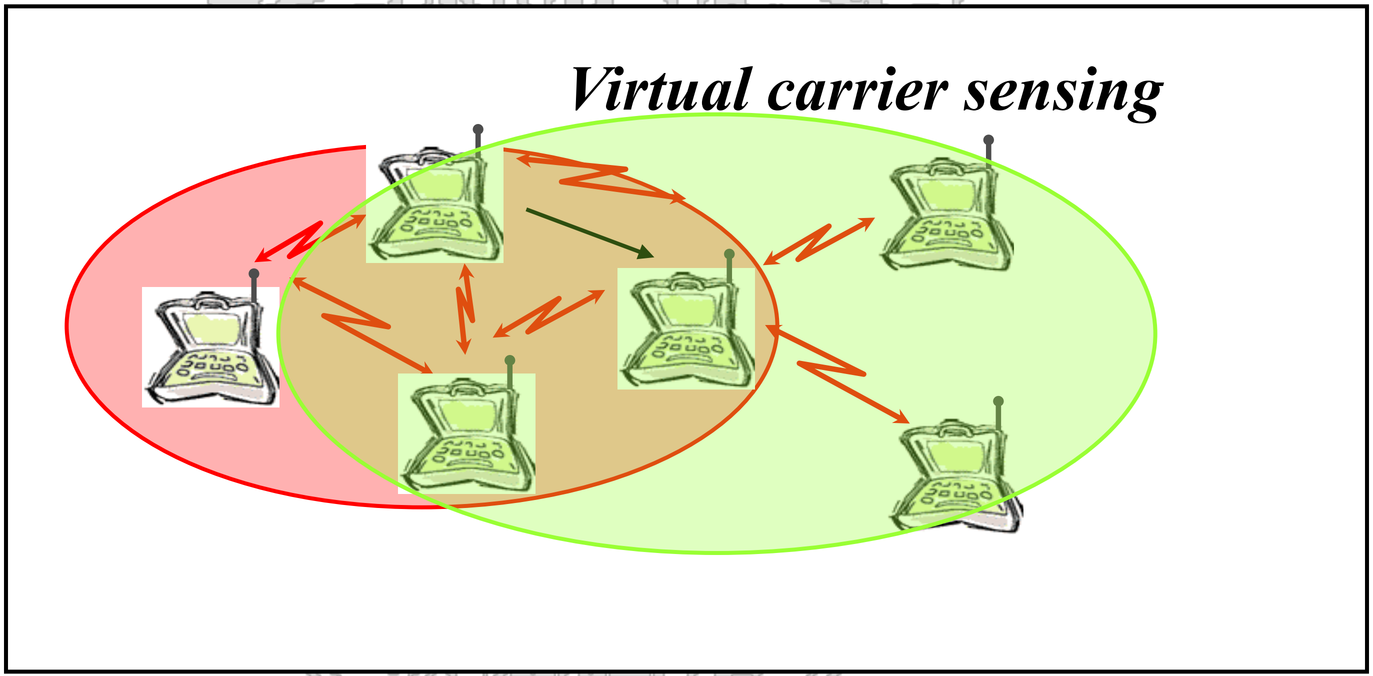
802.11 DCF
Virtual carrier sensing

- To mitigate performance impairments due to the hidden terminal phenomenon DCF uses virtual carrier sensing
- Before transmitting a frame the sender node performs carrier sensing, waits for a time called Distributed InterFrame Space (DIFS) and then transmits a short control packet named Request To Send (RTS), informing its neighbors that it is going to transmit a packet towards the destination
- RTS includes a NAV (Network Allocation Vector) field whose value expresses the time from the RTS reception to the end of the handshake (ACK reception)
  - By receiving the RTS and looking at the NAV value all nodes can estimate when not to transmit in order not to interfere with the on-going transmission
- If the receiver correctly receives an RTS it waits for a SIFS and then transmits a Clear To Send (CTS) message
  - All destination neighbors will know that the channel is busy, and for how long
  - CTS also includes a NAV field
- Upon receiving a CTS, the sender transmits the DATA packet (after SIFS)
- Upon receiving the DATA packet, the destination waits for a SIFS and then sends the ACK
- If the handshake is not correctly completed the node performs a retransmission attempt after an exponential backoff

Virtual Carrier Sensing



Virtual Carrier Sensing



SAPIENZA
UNIVERSITÀ DI ROMA

Other approaches?

- Can TDMA be an option?
- Or an hybrid solution?

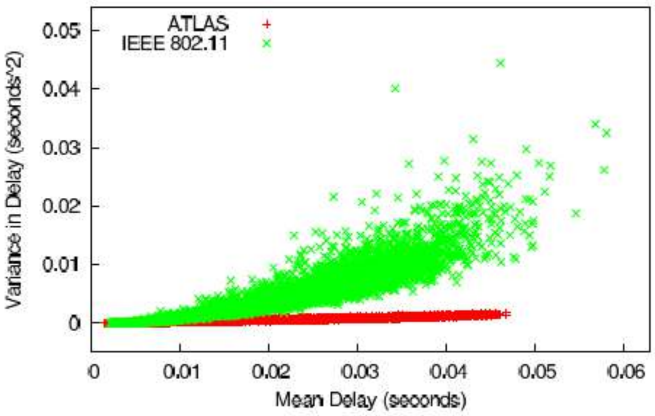
- Several proposals in the literature
  - Hybrid or TDMA approaches used in sensor networks
  - Demonstrated promising for MANET based on simulations

Other approaches?

- Can TDMA be an option?
  - Synchronization in higly dynamic environments
  - Static allocation of resources is a limit
  - Efficiency depends on traffic (control overhead balanced by increased efficiency in high traffic scenarios);
- Or an hybrid solution?
- Several proposals in the literature
  - Hybrid or TDMA approaches used in sensor networks
  - Demonstrated promising for MANET based on simulations, in case of adaptive schemes, also ensuring some level of fairness

Other approaches?

- Can TDMA be an option?
- Or an hybrid solution?
- Several proposals in the literature
  - Hybrid or TDMA approaches used in sensor networks
  - Demonstrated promising for MANET based on simulations
- Jonathan Lutz, Charles J. Colbourn, Violet R. Syrotiuk: ATLAS: Adaptive Topology- andLoad-Aware Scheduling. IEEE Trans. Mob. Comput. 13(10): 2255-2268 (2014)



Routing-Background

Routing-Traditional approach (from Reti 1)

- Intra-AS routing in the Internet
  - Link State Approaches

    (info on the topology graph gathered at nodes which run shortest path algorithms-Dijkstra- to decide the routes to the different destinations –e.g. OSPF routing protocol)
  - Distance Vector approaches (e.g. RIP)

Bellman-Ford

Given a graph G=(N,A) and a node *s* find the shortest path from *s* to every node in N.

A shortest walk from *s* to i subject to the constraint that the walk contains at most h arcs and goes through node *s* only once, is denoted **shortest(<=h) walk and its length is D^h_i.**

<u>Bellman-Ford rule:</u>

Initiatilization $D^h_s=0$, for all h; $c_{i,k}$ = infinity if (i,k) NOT in A; $c_{k,k}=0$; **D^0_i=infinity for all i!=s.**

<u>Iteration:</u>

$$D^{h+1}_i=\min_k [c_{i,k} + D^h_k]$$

Assumption: non negative cycles **(this is the case in a network!!)**

The Bellman-Ford algorithm first finds the one-arc shortest walk lengths, then the two-arc shortest walk length, then the three-arc…etc. →distributed version used for routing

Bellman-Ford

$$D^{h+1}_i = \min_k [c_{i,k} + D^h_k]$$

Can be computed locally.
What do I need?

For each neighbor *k*, I need to know
-the cost of the link to it (known info)
-The cost of the best route from the neighbor *k* to the destination (←this is an info that each of my neighbor has to send to me via messages)

In the real world: I need to know the best routes among each pair of nodes → we apply distributed Bellman Ford to get the best route for each of the possible destinations iterative:

- continues until no nodes exchange info.
- *self-terminating*: no "signal" to stop

asynchronous:

- nodes need *not* exchange info/iterate in lock step!

Distributed, based on local info:

- each node communicates *only* with directly-attached neighbors

Distance Table data structure

each node has its own

- row for each possible destination
- column for each directly-attached neighbor to node
- example: in node X, for dest. Y via neighbor Z:

Cost associated to the (X,Z) link

$D^X(Y,Z)$ = distance *from* X *to* Y, *via* Z as next hop

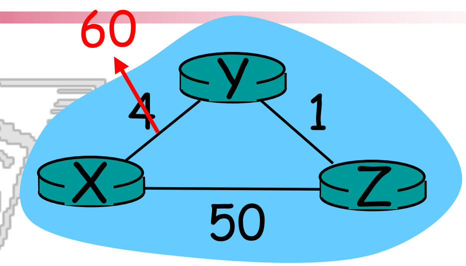
$= c(X,Z) + \min_w \{D^Z(Y,w)\}$

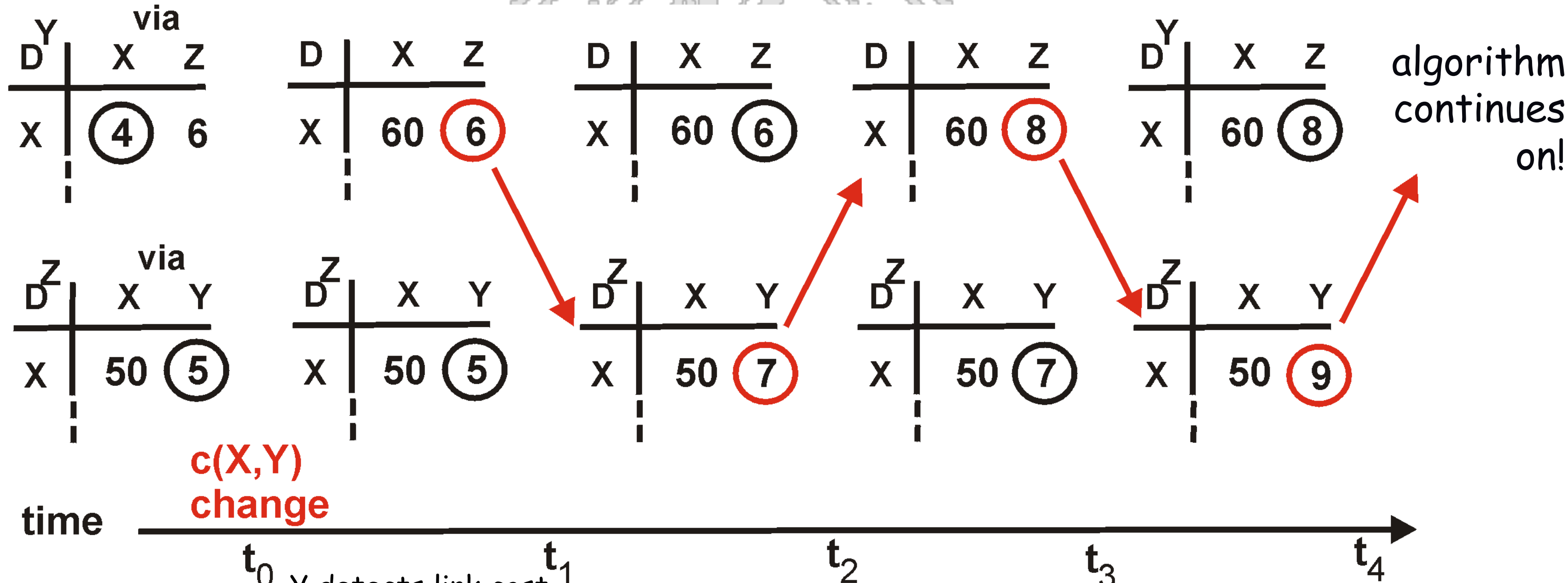
Info maintained at Z. Min must be communicated

Distance Vector: link cost changes

Link cost changes:

- good news travels fast
- ***bad news travels slow*** - "count to infinity" problem!





Y detects link cost Increase but think can Reach X through Z at a total cost of 6 (wrong!!)

The path is Y-Z-Y-X

Count-to-infinity – an everyday life example

Which is the problem here?

the info exchanged by the protocol!! 'the best route to X I have has the following cost…' (no additional info on the route)

A Roman example…

-assumption: there is only one route going from Colosseo to Altare della Patria: Via dei Fori Imperiali. Let us now consider a network, whose nodes are Colosseo., Altare della Patria, Piazza del Popolo

Count-to-infinity – everyday life example (2/2)

The Colosseo. and Alt. Patria nodes exchange the following info

- Colosseo says ‘the shortest route from me to P. Popolo is 2 Km’
- Alt. Patria says ‘the shortest path from me to P. Popolo is 1Km’

Based on this exchange from Colosseo you go to Al. Patria, and from there to Piazza del Popolo OK **Now due to the big dig they close Via del Corso (Al. Patria—P.Popolo)**

- Al. Patria thinks ‘I have to find another route from me to P.Popolo.

Look there is a route from Colosseo to P.Popolo that takes 2Km, I can be at Colosseo in 1Km → I have found a 3Km route from me to P.Popolo!!’ Communicates the new cost to Colosseo that updates ‘OK I can go to P.Popolo via Al. Patria in 4Km’

VERY WRONG!! Why is it so? I didn’t know that the route from Colosseo to P.Popolo was going through Via del Corso from Al.Patria to P.Popolo (which is closed)!!

Count to infinity-solutions

- Bounded network diameter (RIP)
  - It is possible to use a TTL and discard all packets that have traversed more than x hops.
  - If network diameter is limited (15 in RIP networks) convergence in case of count to infinity is fast.
- Split horizon with poison reverse
  - Limits transmitted information. If A uses information received by B to select the route towards D (in other words if A's next hop relay to reach D is B), A will not communicate valid route lengths to B, or it will communicate infinity
    - ✓ Broadcast cannot be used to send updates
    - ✓ Does not solve all loop situations
- Trigger Updates (to fasten convergence)
  - Instead of sending periodic updates, updates can be transmitted immediately in case route lengths change

Link State Approach

- Each node periodically sends information on its neighbors (and the associated cost on the links to them) to all other nodes in the network
  - Via flooding or a variant of flooding
- Updates can be sent also in case changes are detected
- As each node has a complete view of the network topology it can locally compute the best route towards the destination
  - Running Dijkstra
- The node then populates its rouing table accordingly

Ad Hoc Networks Routing

Routing in ad hoc networks- Goals

- Multi-hop path routing capability
- Dynamic topology maintenance
- "No loops"
- Minimal control overhead
- Low processing overhead
- Self-starting

Primary Approaches

- Proactive
  - Based on traditional distance-vector and link-state protocols
  - Each node maintains route to each other network node
  - Periodic and/or event triggered routing update exchange
  - Higher overhead in most scenarios
  - Longer route convergence time
  - Examples: DSDV, OLSR

Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing

- Proactive, distance vector approach (uses distributed asynchronous Bellman Ford). Updates on route costs transmitted periodically or when significant new information is available.
- Difference wrt Bellman Ford: in ad hoc networks there are frequent changes in the topology, solutions must try to avoid loops (approaches such as Poison reverse are not effective in broadcast channels, we seek solutions which are simple and fully distributed)
- Metrics: fresh routes better than stale routes, number of hops used to select among the fresh routes
- How to identify fresh routes? By means of sequence numbers identifying the freshness of the communicated information. When changes occur, the sequence number increase.

Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing

- Periodically destination nodes transmit updates with a new sequence number (and such updates are propagated by the other nodes).
- Updates periodically sent by nodes contain information on the costs to achieve the different destinations and the freshness of the route
- Data broadcast include multiple entries each with:
  - Destination address
  - Number of hops required to reach the destination
  - Sequence number of the information received regarding that destination as originally stamped by the destination
- In the header the data broadcast also include:
  - Address (HW address/Net address) of the sender of the message
  - Sequence number created by the transmitter
- Two types of updates (full dump or incremental-only changes- to decrease bandwidth consumption.

Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing

- How can the costs be modified? Cost=number of hops, target: using fresh routes as short as possible → a link cost changes from 1 to inf and from inf to 1
- How do we detect that a link is 'broken'? At layer 2 (no hello messages received for some time, or attempts to retransmit a frame exceeds the MAC protocol threshold) or at layer 3 (do not receive periodic updates by a neighbor)
- Link cost increase (1→ inf):
  - The nodes incident to that link (A,B) discover it (see above)
  - Routes going through that link get assigned an inf cost in nodes A and B routing tables
  - A new sequence number is generated by the mobile node. Mobile nodes different from the destination use odd SN, the destination even SN.
  - Updates with routes with infinite cost are **immediately transmitted** by nodes
- Link cost decrease (inf→1):
  - Immediately transmits updates

Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing

- When a node receives updates it sees if costs to reach the different destinations can be improved:
  - routes with more recent sequence numbers to a given destination are used
  - if more routes available with the same SN the shortest is used
- Newly recorded routes are scheduled for immediate advertisement (inf→ finite value)
- Routes with improved metric are scheduled for advertisement at a time which depends on the estimated average settling time for routes to that particular destination (based on previous history) → delayed advertisements to decrease the overall overhead
- As soon as a route cost changes the node may delay informing its neighbors but immediately starts using the new information for its forwarding

Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing-Correctness

- Assuming routing tables are stable and a change occurs
  - let G(x) denotes the routes graph from the sources to x BEFORE the change (assume no loop)
  - change occurs at *i* when 1) the link from *i* to its parent p(*i*) in G(x) breaks → *i* sets to inf that route (no loop can occur) 2) node *i* receives from one of its neighbors k a route to x with sequence number SN^x_k and metric m which is selected to replace the current metric *i* is using to select the route to x (this occurs only if SN^x_k greater than the previous SN I had stored Sn^x_i or if the two SN are equal but the new route has a lower hop cost → in the first case if selecting k leads to a loop then $SN^x_k <= Sn^x_i$ which is a contradiction, in the second case the claim comes from the observation that in presence of static or decreasing link weights distance- vector algorithms always maintain loop-free paths).

OLSR

- Optimized Link State Routing (OLSR) is a link state protocol for MANETs
  - suited for large and dense ad hoc networks
- The key concept is to decrease the overhead of flooding by means of identifying a subset of nodes (multipoint relays) in charge of forwarding the information during the flooding process
  - **Multipoint relay Y**: a node selected by at least one of its 1-hop neighbors (say node X) to relay all valid broadcast information it receives from X (the broadcast information is valid if it has not expired and not duplicate).
    - ✓ MPR(X)=set of multipoint relays of node X
    - ✓ neighbors of node X which are not in MPR(X) receive and store the broadcast messages transmitted by X but DO NOT retransmit them
  - A node X which has selected a neighbor Y as multi-point relay is called a **multipoint relay selector** of node Y
- Requires only partial link state to be flooded
  - links from MPR to their selectors must be declared
    - ✓ enough to ensure routes to each destination can be found
  - additional link state information MAYBE advertised

OLSR- other features

- The protocol is fully distributed
- Proactive approach: routes are always available when needed
- Other features:
  - Time between updates can be tuned to increase reactivity to topological changes
  - does not require reliable transmission
    - ✓ some losses are tolerated ← needed info are periodically transmitted
      - OLSR control packets are embedded in UDP datagrams
    - ✓ sequenced delivery of the messages is not needed ← proper recostruction of the sequence is possible due to use of sequence numbers
  - support to other MANET related issues
    - ✓ Sleep mode operation
    - ✓ Multicasting

Packet format

in bytes

Time for which the info is valid

Es: Hello messages

Topology declaration messages

Message source

Num. Max di hop che un messaggio può attraversare

Included by originator And increased each time It transmits a bew message

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Packet Length         |    Packet Sequence Number     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Message Type |     Vtime     |         Message Size          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Originator Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time To Live |   Hop Count   |    Message Sequence Number    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                            MESSAGE                            :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Message Type |     Vtime     |         Message Size          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Originator Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time To Live |   Hop Count   |    Message Sequence Number    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                            MESSAGE                            :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Each node maintains triples <originator address, sequence number, if the message has already been transmitted> for messages recently received. This allows to discard duplicates.

Packets with TTL=0 or inconsistent with OLSR specifications are also discarded.

Hello messages

- Hello messages are used to
  - verify if links are up and running (link sensing)
    - ✓ if no hello message is received by a neighbor in a given interval a timeout occurs and the link is assumed down
    - ✓ to exchange with neighbors neighborhood information (piggybacked in the hello messages)
      - allows to compute two hop neighborhood
        - » Which in turn is needed to select multipoint relays

How MPRs are selected

- Each node X selects its MPRs among its one hop neighbors
- The set is selected to cover node X 2-hop neighborhood
  - MPR(X) is an arbitrary subset of node X one hop neighbors such that each node z in node X' s two hop neighborhood have a neighbor in MPR(X)
    - ✓ can be selected with a greedy protocol
      - MPR(X)=null, C(X)= two hop neighborhoos of X
      - For each neighbor Y of X, its degree D (Y) is computed without considering X and its neighbors
      - Y is included in MPR(X) if its the only neighbor of X able to cover a two hop neighbor
        - » C(X)=C(X) \ {nodes covered by Y}
      - till (X)=null
        - » Include in MPR(X) the neighbor of X which allows to cover more uncovered nodes in in C(X) (ties broken based on degree D)
        - » C(X)=C(X) \ {nodes covered by selected neighbor}
  - The smaller MPR(X) the less control overhead exchanged

Upon receiving a message *m* at node Y

- If the received message is not a duplicate, is valid and has a non zero TTL
  - if it is received by an MPR selector of Y
    - ✓ retransmit ***m***
      - ***reduce by one the message TTL***
      - ***increase by one the message hop count***
      - ***broadcast on all node Y interfaces***
    - ✓ update or create the entry for the message in the duplicate set (→upon receiving the same message the node can identify it was already received and retransmitted→ can be discarded)

How are data packets forwarded

- Information on topology are disseminated to all the network nodes
  - in an efficient way
    - ✓ exploiting the backbone of multipoint relays
    - ✓ limiting as much as possible topology information
- Each node then locally runs a shortest path algorithm to determine paths to the different destination
  - fills a routing table
  - upon reception of a data packet forwarding is performed according to the routing table

Ad hoc Network Routing

IoT, a.a 2017/2018

Prof. Chiara Petrioli

Un. of Rome "La Sapienza"

Chiara Petrioli[†]

[†] *Department of Computer Science – University of Rome "Sapienza" – Italy*

Why traditional approaches have limits?

- Proactive protocols are costly in terms of overhead (the bandwidth and energy are critical resources)
- The cost of maintaining routes updated may not make sense in an environment in which
  - Medium-high mobility
  - Medium-high dynamicity (awake/asleep states)

  Motivate frequent changes in the the optimum route (requiring updates) while
  - Traffic is generally low (so the cost of maintaining always updated routes is not balanced by their use)

  If this is the scenario what can we do?

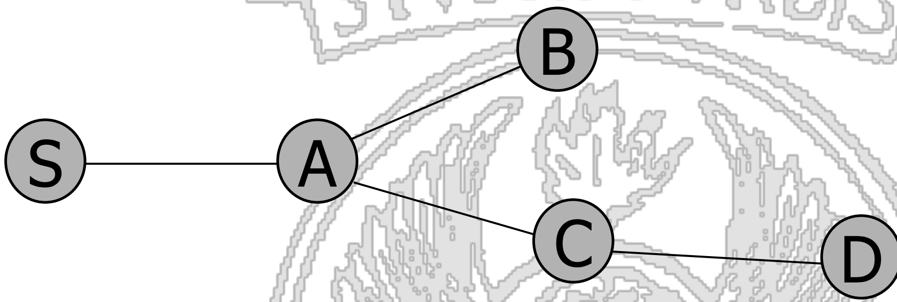
2 Primary Approaches (cont.)

- Reactive (on-demand)
  - Source build routes on-demand by “flooding”
  - Maintain only active routes
  - Route discovery cycle
  - Typically, less control overhead, better scaling properties
  - Drawback: route acquisition latency
  - Example: AODV, DSR

Ad hoc On-Demand Distance Vector Routing

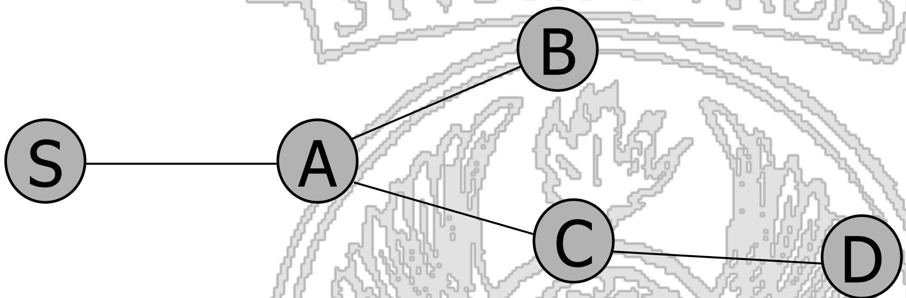
- AODV: Reactive (nodes that do not lie on active paths neither maintain any routing information nor participate in any periodic routing table exchange; a node does not have to discover/maintain a route to a destination till it is on a path to it or has to send messages to it)
- *Route discovery cycle* used for route finding
- Maintenance of *active* routes
- Sequence numbers used for loop prevention and as route freshness criteria
- Descendant of DSDV (standard distance vector approach mapped to ad hoc networks), in AODV no periodic updates but pure on-demand operation.
- Provides unicast and multicast communication

AODV: Route Discovery



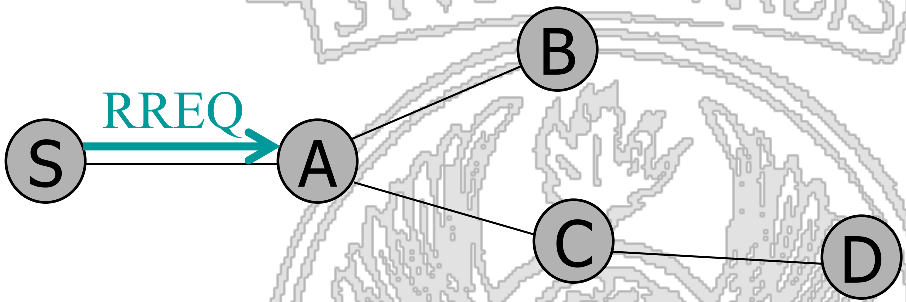
1. Node *S* needs a route to *D AND does not have routing info for it in its table*

AODV: Route Discovery



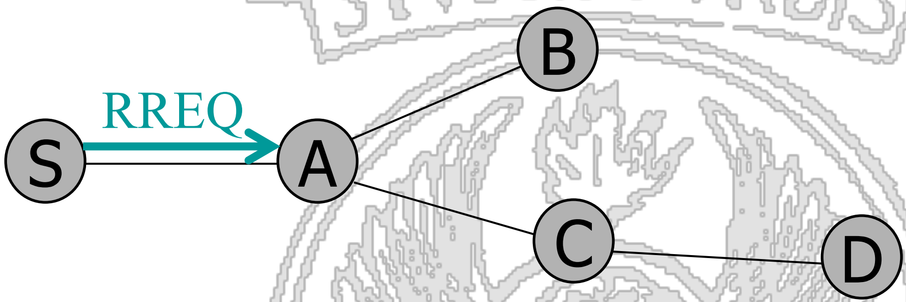
1. Node *S* needs a route to *D*
2. Creates a Route Request (RREQ)
   - Enters *D*'s IP addr, seq#,
   - *S*'s IP addr, seq#
   - hopcount (=0), broadcast ID

AODV: Route Discovery



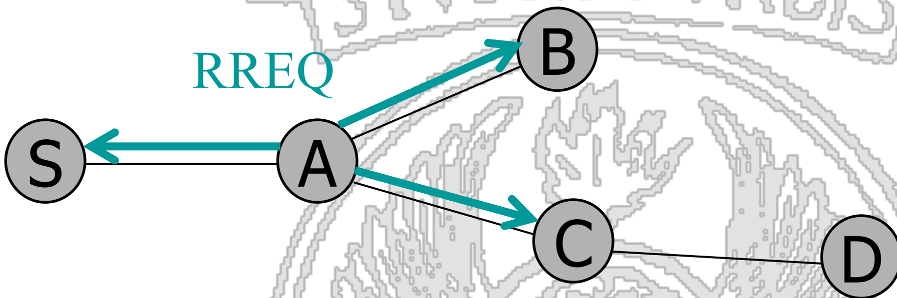
1. Node *S* needs a route to *D*
2. Creates a Route Request (RREQ)
   Enters *D*'s IP addr, seq#,
   S's IP addr, seq#
   hopcount (=0), broadcast ID
3. Node *S* broadcasts RREQ to neighbors

AODV: Route Discovery



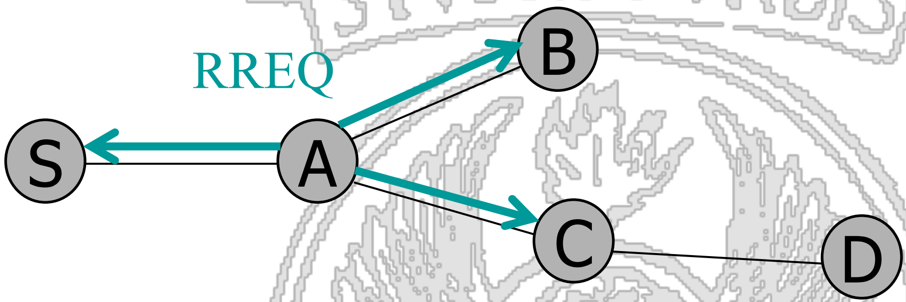
4. Node *A* receives RREQ
   - Makes reverse route entry for *S*
     dest=*S*, nexthop=*S*, hopcnt=1,expiration time for reverse path
   Source node, Source node SN,D,broadcastID also maintained
   - It has no route to *D*, so it rebroadcasts RREQ (hopcount increased)
   - If it has already received that request (same source and broadcast ID) it discards the RREQ
   - if it knows a valid path to D it will send back a reply to the source

AODV: Route Discovery



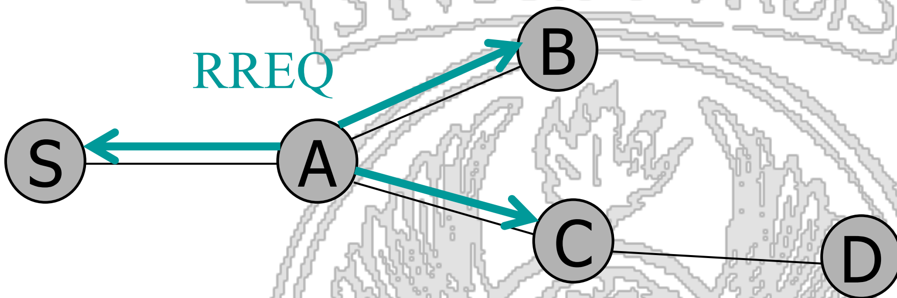
4. Node *A* receives RREQ
   - Makes reverse route entry for *S*
     dest=*S*, nexthop=*S*, hopcnt=1
   - It has no route to *D*, so it rebroadcasts RREQ

AODV: Route Discovery



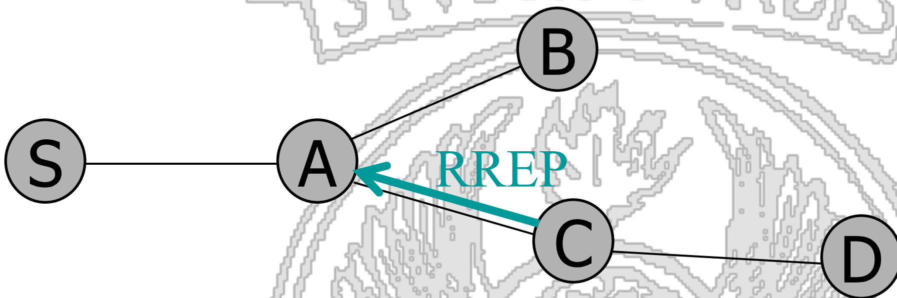
5. Node *C* receives RREQ
   - Makes reverse route entry for *S*
     dest=*S*, nexthop=*A*, hopcnt=2
   - It has a route to *D*, and the seq# for route to *D* is >= *D*'s seq# in RREQ

AODV: Route Discovery



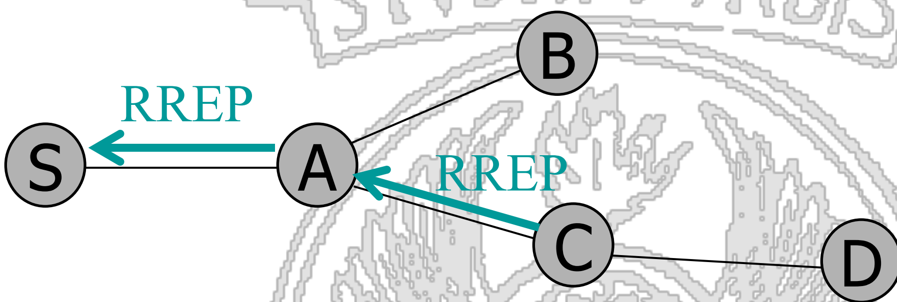
5. Node *C* receives RREQ (cont.)
   - *C* creates a Route Reply (RREP)
     Enters *D*'s IP addr, seq#
     S's IP addr, hopcount to *D* (= 1), lifetime of the forward route
   - Unicasts RREP to *A*

AODV: Route Discovery



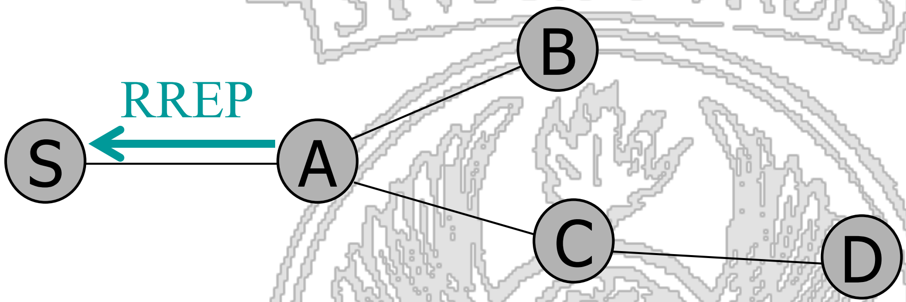
5. Node *C* receives RREQ (cont.)
   - *C* creates a Route Reply (RREP)
     Enters *D*'s IP addr, seq#
     S's IP addr, hopcount to *D* (= 1)....
   - Unicasts RREP to *A*

AODV: Route Discovery



6. Node *A* receives RREP
   - Makes forward route entry to *D*
     dest = *D*, nexthop = *C*, hopcount = 2
   - Unicasts RREP to *S*

AODV: Route Discovery

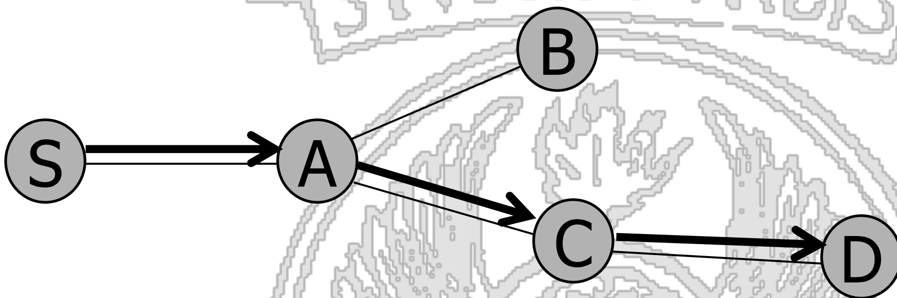


7. Node *S* receives RREP
   – Makes forward route entry to *D*
     dest = *D*, nexthop = *A*, hopcount = 3

Also the latest SN of the destination is updated when receiving the RREP

Nodes not along the path determined by the RREP will timeout after ACTIVE_ROUTE_TIMEOUT (3000ms) and will delete the reverse pointer

AODV: Route Discovery



7. Node *S* receives RREP
   – Makes forward route entry to *D*
     dest = *D*, nexthop = *A*, hopcount = 3
   – Sends data packets on route to *D*

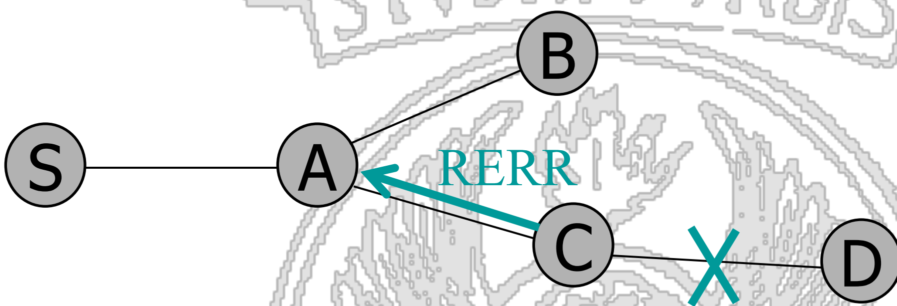
What if….

- A node receives further RREPs for the same request? (e.g. more neighbors of a node had paths to the destination in cache)?
  - upon reception of another RREP the node updates its routing information and propagates the RREP only if the RREP contains either a greater destination SN, or the same destination SN with a smaller hopcount

Other info maintained

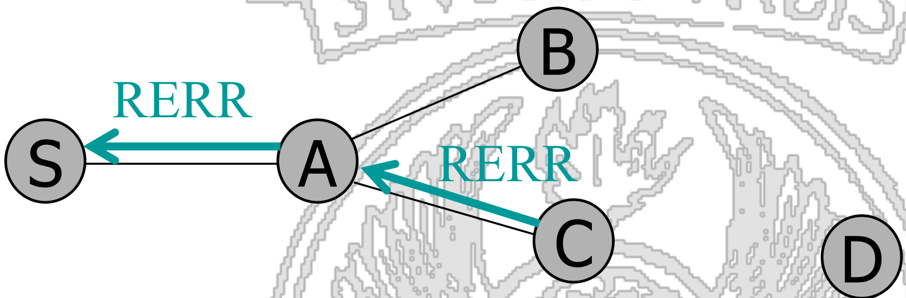
- Each node maintains the list of active neighbors, neighbors sending to a given destination through it
  - useful for route maintenance
- Routing table entries: dest,next hop, hopcount, dest SN, active neighbors for this route, expiration time for route table entry (updates each time the route is used for transmitting data → routes entries are maintained if the route is active)

AODV: Route Maintenance



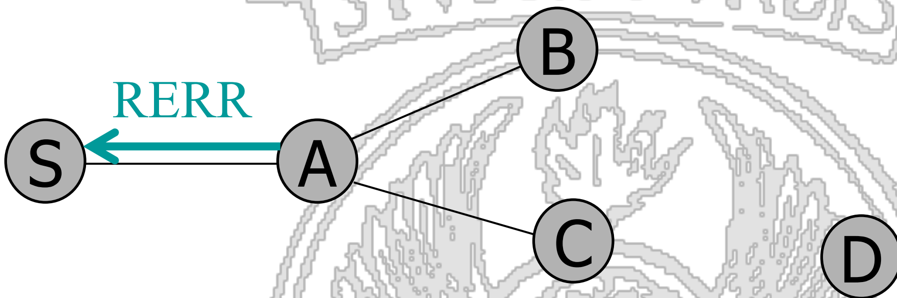
1. Link between *C* and *D* breaks
2. Node *C* invalidates route to *D* in route table
3. Node *C* creates Route Error (RERR) message
   - Lists all destinations which are now unreachable
   - Sends to upstream neighbors
   - Increases of one the SN of the destination

AODV: Route Maintenance



4. Node *A* receives RERR
   - Checks whether *C* is its next hop on route to *D*
   - Deletes route to *D*
   - Forwards RERR to *S*

AODV: Route Maintenance



5. Node *S* receives RERR
   - Checks whether *A* is its next hop on route to *D*
   - Deletes route to *D*
   - Rediscovers route if still needed (in that case it sends a RREQ with a SN which is equal to the last known destination Sequence Number +1)

AODV: Optimizations

- Expanding Ring Search
  - Prevents flooding of network during route discovery
  - Control Time To Live (TTL) of RREQ to search incrementally larger areas of network
  - Advantage: Less overhead when successful
  - Disadvantage: Longer delay if route not found immediately

AODV: Optimizations (cont.)

- Local Repair
  - Repair breaks in active routes locally instead of notifying source
  - Use small TTL because destination probably hasn't moved far
  - If first repair attempt is unsuccessful, send RERR to source
  - Advantage: repair links with less overhead, delay and packet loss
  - Disadvantage: longer delay and greater packet loss when unsuccessful

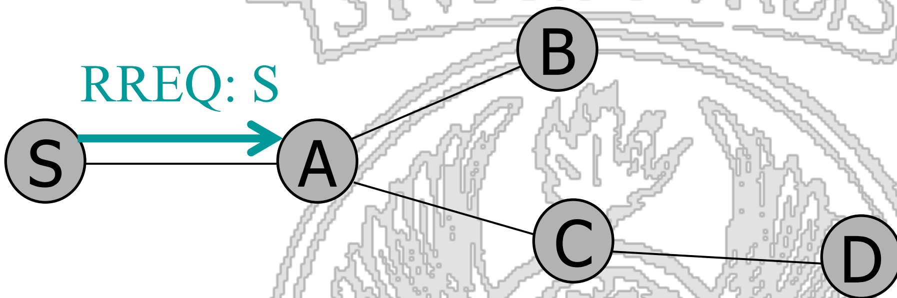
AODV: Summary

- Reactive/on-demand
- Sequence numbers used for route freshness and loop prevention
- Route discovery cycle
- Maintain only active routes
- Optimizations can be used to reduce overhead and increase scalability

Dynamic Source Routing (DSR)

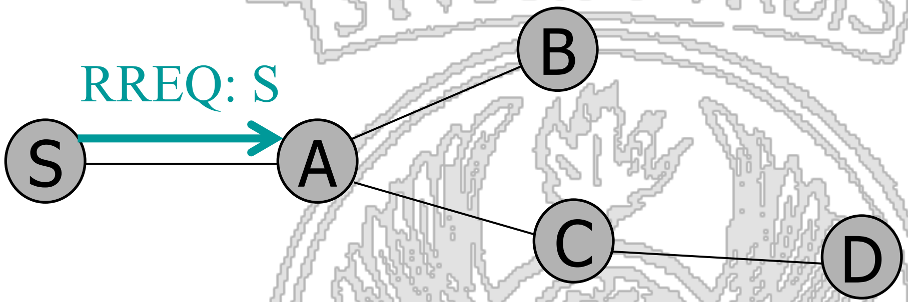
- Reactive
- *Route discovery cycle* used for route finding
- Maintenance of *active* routes
- Utilizes *source routing*

DSR: Route Discovery



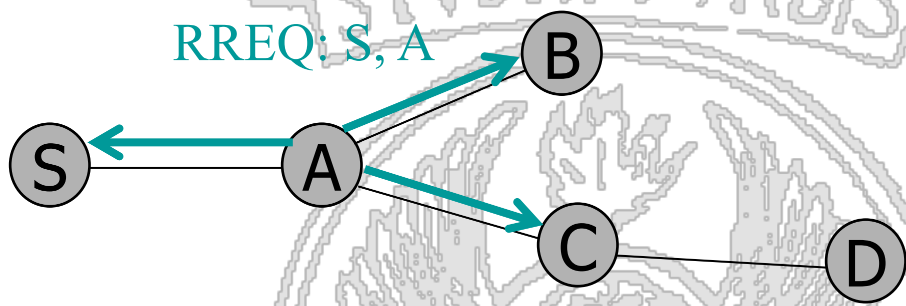
1. Node *S* needs a route to *D*
2. Broadcasts RREQ packet
   1. RREQ identifies the target of the route discovery, contains a route record in which the traversed route is accumulated, contains a pair <initiator, request id> uniquely identifying the request

DSR: Route Discovery



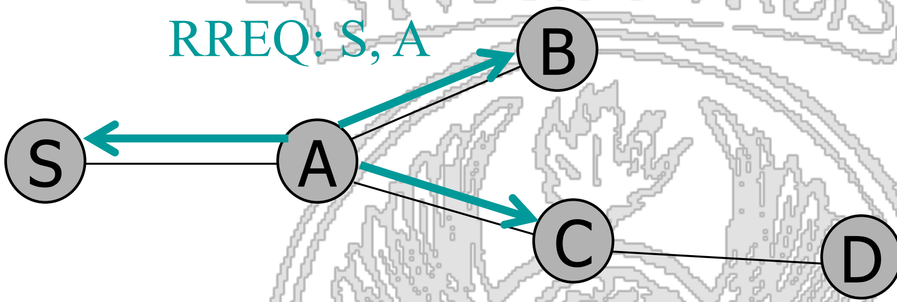
1. Node *S* needs a route to *D*
2. Broadcasts RREQ packet
3. Node *A* receives packet, has no route to *D AND is NOT D*
   - Rebroadcasts packet after adding its address to source route

DSR: Route Discovery



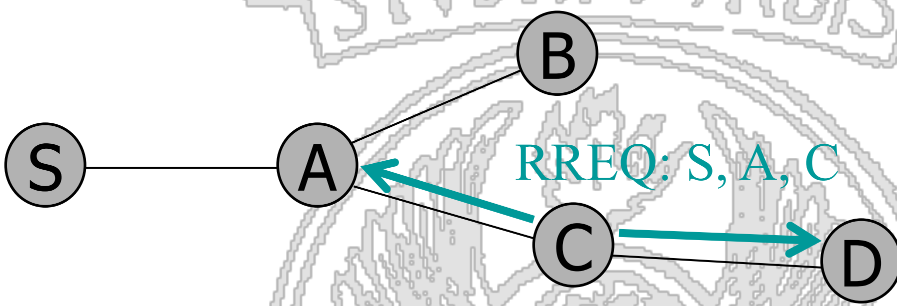
1. Node *S* needs a route to *D*
2. Broadcasts RREQ packet
3. Node *A* receives packet, has no route to *D*
   - Rebroadcasts packet after adding its address to source route

DSR: Route Discovery



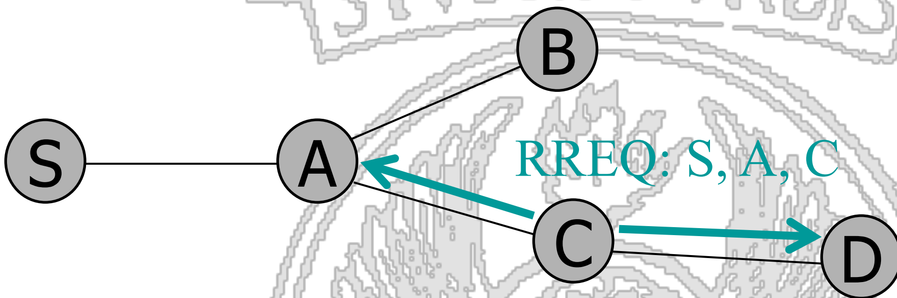
4. Node *C* receives RREQ, has no route to *D*
   – Rebroadcasts packet after adding its address to source route

DSR: Route Discovery



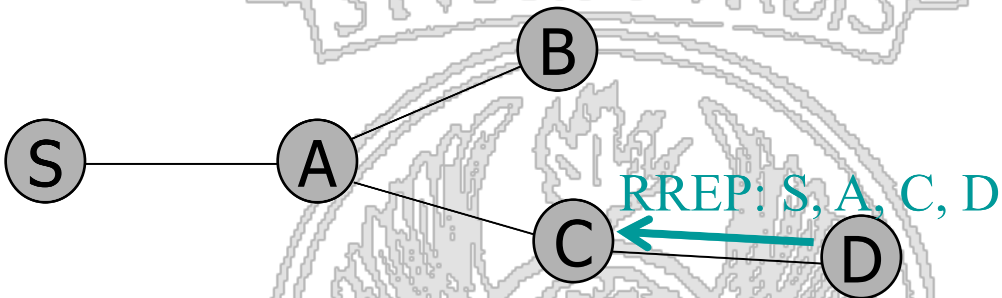
4. Node *C* receives RREQ, has no route to *D*
   - Rebroadcasts packet after adding its address to source route

DSR: Route Discovery



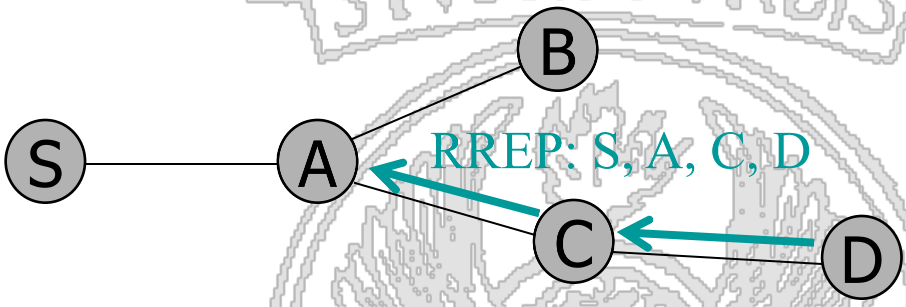
4. Node *C* receives RREQ, has no route to *D*
   – Rebroadcasts packet after adding its address to source route
5. Node *D* receives RREQ, unicasts RREP to *C*
   – Puts source route accumulated in RREQ into RREP

DSR: Route Discovery



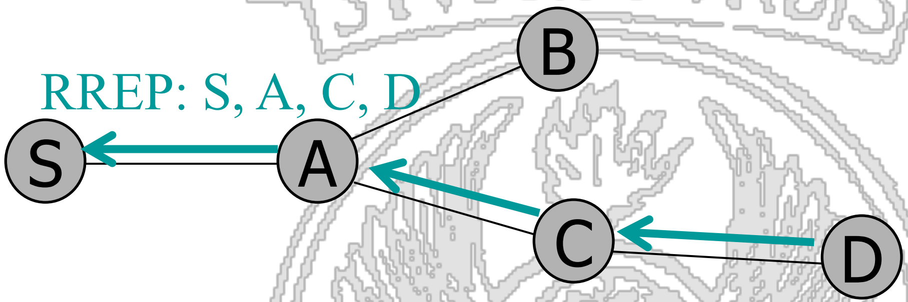
4. Node *C* receives RREQ, has no route to *D*
   – Rebroadcasts packet after adding its address to source route
5. Node *D* receives RREQ, unicasts RREP to *C*
   – Puts source route accumulated in RREQ into RREP

DSR: Route Discovery



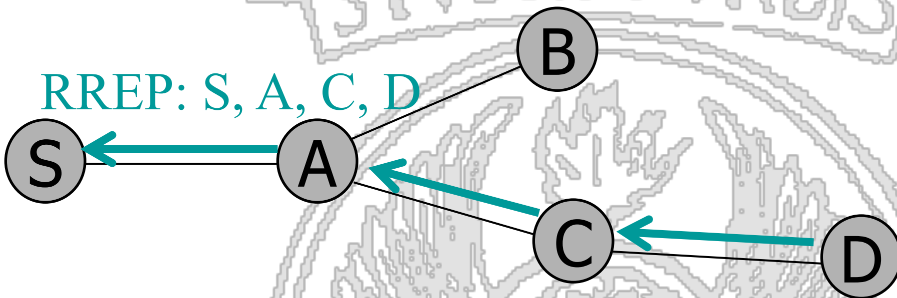
6. Node *C* receives RREP
   - Unicasts to *A*

DSR: Route Discovery



6. Node *A* receives RREP
   - Unicasts to *S*

DSR: Route Discovery



8. Node *S* receives RREP
   – Uses route for data packet transmissions

General node operation upon receiving RREQ

- If the pair <initiator address, request ID> has recently been seen, DISCARD
- If the node ID is already listed in the source route DISCARD→ avoids loops
- If I'm the destination, send a RREP
- Otrherwise, append my ID in the source route and rebroadcast (orange cases already seen in the previous slides)

Route maintenance

- The two endpoints of a failed link are transmitted to the source in a route error packet
- Upon a receiving a RERR packet a node invalidates all the routes going through that link
- If the route is invalidated and it is needed, a new route must be discovered

SAPIENZA
UNIVERSITÀ DI ROMA

Optimizations (1)

- Extensive use of caching (caching source routes means that I already know all the route to intermediate destinations; discovery of a better route to an intermediate destination also brings me to improve the route to the final destination). Transmitting packets or sending back replies make me learn routes.

- A node that knows a route to a given destination (has a source route in cache) can immediately answer a RREQ
  - Broadcast storm? Each nodes waits for a time which is C*(h-1+r), r random in (0,1), h length of the route I'm advertising. Only if I haven't received other routes –listen to other routes tx in the meanwhile-I transmit mine.

Optimization (2)

- Operation in promiscuous mode (I keep discovering new routes by transmission of routes by my neighbours)
- RREQ overhead minimization: first set a TTL=1, if I do not get answer I set it to infinity
- Path shortening: if Y receives a packet by node X but in the source route we have X, B,…,C,Y, Y signals the path can be shortened (unsolicited RREP)
- What if the network is disconnected? Exponential back-off to decrease the quantity of RREQ sent

AODV and DSR Differences

- DSR uses source routing; AODV uses next hop entry
- DSR uses route cache; AODV uses route table
- DSR route cache entries do not have lifetimes;
  AODV route table entries do have lifetimes

Proactive Solutions: Drawbacks

- Updates overhead, especially in presence of high mobility
- Overhead for enforcing loop freedom
- Large routing tables
- Low *scalability*

➢ Is it really necessary to maintain a consistent view of the network topology?

Reactive Protocols: Drawbacks

- The discovery phase introduces long delays
- Route discovery and maintenance is very sensitive to node mobility
- Route caching is memory greedy
- The size of the header of a data packet can become cumbersome in approaches such as DSR (no scalability)
- Operating in promiscuous mode is energy-consuming.
- Relying on flooding based route discovery is resource consuming.
- ➢ Is the dependency on the network topology avoidable?