Local area networks

Katia Jaffrès-Runser and Gentian Jakllari

{kjr,jakllari}-at-n7.fr

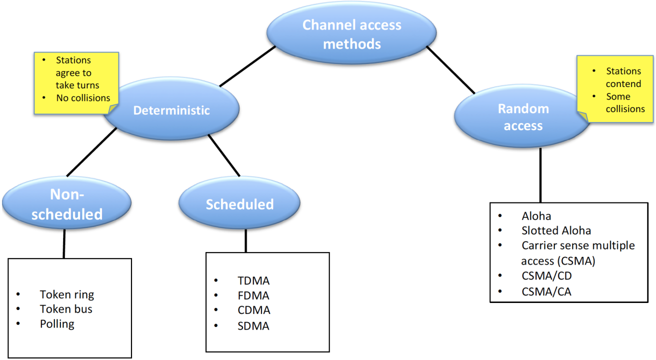
Toulouse INP - ENSEEIHT

Département Sciences du Numérique

1ère annèe

2017-2018

ALOHA
Carrier Sense Multiple Access



ALOHA networks

ALOHA: The origin of random access[1]

- Developed in the late 60's by Norman Abramson et al to allow the 7 campuses of the Univ. of Hawai'i, located on 4 different islands, to share computer resources on the main campus
- The first user terminals went into operation in June 1971
  - The communication protocol was implemented by a special-purpose piece of equipment – the terminal control unit (TCU)
    - Compare it to a wifi card...
  - A user terminal was attached to the TCU

[1] N. Abramson, "The AlohaNet - surfing for wireless data [History of Communications]," in IEEE Communications Magazine, vol. 47, no. 12, pp. 21-25, Dec. 2009.

ALOHA networks

- **Key decision:** Use the direct form of transmitting user information in a single high-speed packet burst in a *shared wireless channel*
  - Driven by the need for a simple design; throughput computed several weeks after the decision
  - Cost of memory for a packet buffer of 88 bytes was about $300
- Channel access philosophy : let collisions happen, detect when they occur and then try again.
  - Any station can send data at any time
  - If, while transmitting, any data is received concurrently, then there is a collision – will need to try again.

ALOHA networks

- **Key decision:** Use the direct form of transmitting user information in a single high-speed packet burst in a *shared wireless channel*
  - Driven by the need for a simple design; throughput computed several weeks after the decision
  - Cost of memory for a packet buffer of 88 bytes was about $300
- Channel access philosophy : let collisions happen, detect when they occur and then try again.
  - Any station can send data at any time
  - If, while transmitting, any data is received concurrently, then there is a collision – will need to try again.

How to try again ?

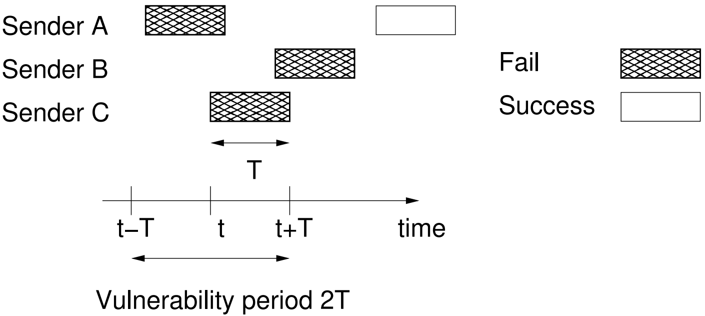
Re-send data after a random duration called the Backoff period

- Avoids repeated collisions.
- The way this random choice is made influences the overall performance.

ALOHA networks

Vulnerability period

The message transmitted at time t experiences a collision if any other message overlaps partially with its transmission.



If all messages have equal length T, then the vulnerability period is of size $2T$.

ALOHA networks

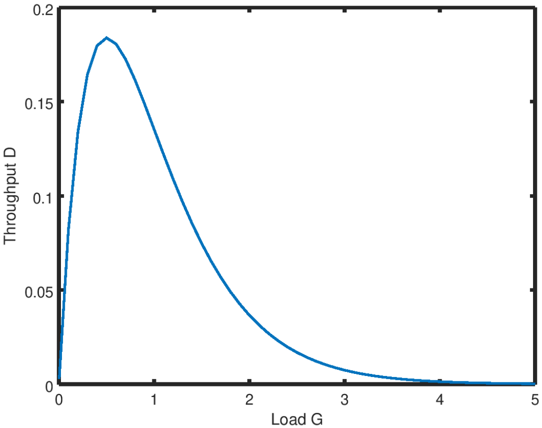
Throughput achieved by ALOHA

It can be derived as follow:

- Suppose that the number of transmission attempts per frame duration T follows a Poisson distribution of mean G.
  Thus the probability of having k attempts during T is: $\frac{G^k e^{-G}}{k!}$
- The probability of having no collision for the vulnerability period of 2T is given by e^{-2G}
- Thus, the throughput is the number G of attempts during T that don't experience any collision :

$$D = G \cdot e^{-2G}$$

Throughput for ALOHA



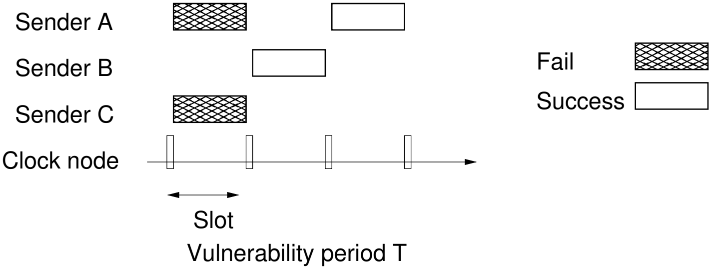
Maximum throughout is obtained for a load $G = 0.5$, that is $D = 0.5/e \simeq 0.184$.

- Very low! Only 18% of frames don't collide at best.

Slotted ALOHA

Increase the efficiency of ALOHA

Idea: reduce the vulnerability period duration by synchronizing transmissions



- All nodes are synchronized on a given slot duration of size T
- A transmission can only start at slot begin.
  $\rightarrow$ vulnerability period is reduced to T.

Slotted ALOHA

Efficiency of slotted ALOHA

As vulnerability period is reduced to T, the throughput increases to:

$$D = G \cdot e^{-G}$$

with e^{-G} the odds of experiencing 0 attempts during T for load G.

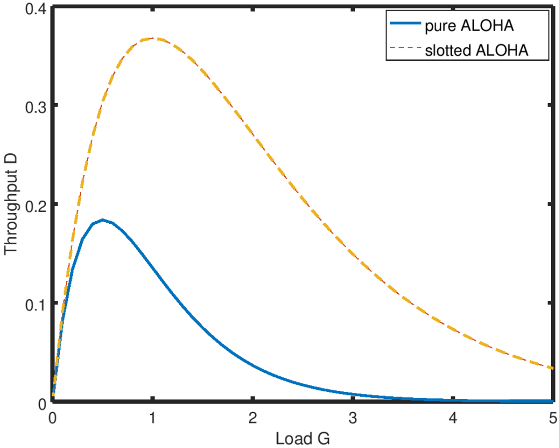
The number of transmissions E

to get a message through increases exponentially with G :

$$E = \sum_{k=1}^{k=\infty} kP_k = e^G$$

with P_k the probability to transmit a message after k attempts given by $P_k = e^{-G}(1 - e^{-G})^{k-1}$.

Slotted ALOHA vs pure ALOHA



Here, peak utilization is of $1/e$, $\simeq 36.8\%$ if one attempt per slot is made in average.

Towards better random access

Become a bit more polite

Become a bit more polite

- Listen before speaking,
- If someone speaks, defer transmission to a later date.

Towards better random access

Become a bit more polite

- Listen before speaking,
- If someone speaks, defer transmission to a later date.

Carrier sense

The node has to *sense the channel* to detect an ongoing transmission.

May collisions still happen then ?

CSMA

A.k.a. *Carrier Sense Multiple Access* is a family of protocols where a node wanting to transmit a message :

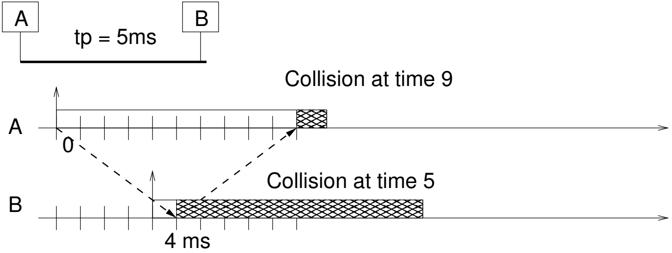
- Senses the channel
  - If the channel is busy, then she defers transmission
  - If the channel is idle, then she transmits

Whenever a node starts transmitting, **it sends the complete message**.

Carrier Sense Multiple Access

Can collisions still happen?

CSMA is very sensitive to propagation delay.



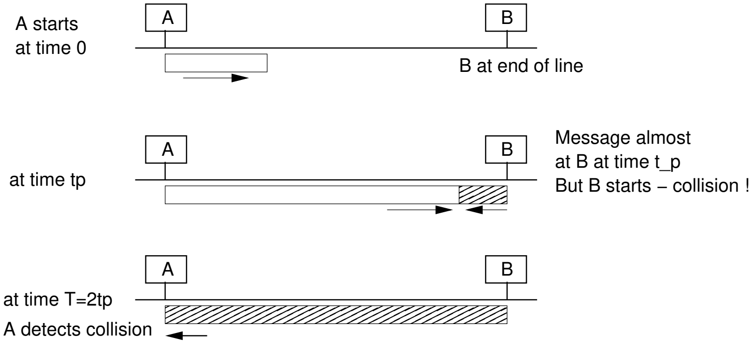
During 5ms the channel is seen as free for other nodes.
Node A can only see the collision after $2.t_p$ (round-trip duration).

Carrier Sense Multiple Access

Vulnerability period

In CSMA, the vulnerability period is the duration for the 1st bit to travel until the end of line and back, i.e.

$$T = 2t_p$$



Carrier Sense Multiple Access

Several variants of CSMA exist:

- 1-persistent CSMA
- non-persistent CSMA
- *p*-persistent CSMA
- CSMA/CD (collision detection)
- CSMA/CA (collision avoidance)
- CSMA/CR (collision resolution)

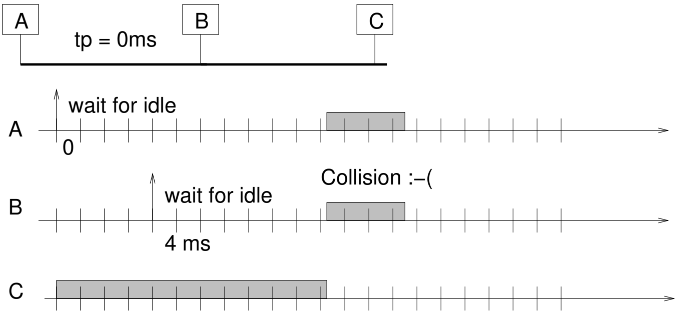
1-persistent CSMA

Algorithm for a ready-to-transmit node

- Sense the channel
  - If the channel is busy, then wait for end of ongoing transmission
  - If the channel is idle, then transmit immediately (*with probability 1*).

$\rightarrow$ **persistently listens** while transmitting
to detect idle state as soon as possible.

Issue with 1-persistent CSMA



Peak throughput is a bit better than for slotted ALOHA: $\simeq$ 52.9%

Non-persistent CSMA

Algorithm for a ready-to-transmit node

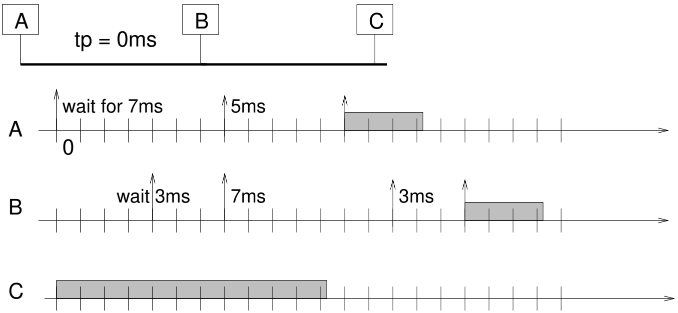
Here, the sender doesn't actively listen to detect the end of an ongoing transmission.

- Sense the channel
  - If the channel is busy, then **wait a random time** and sense channel again
  - If the channel is idle, then transmit immediately

$\rightarrow$ **No persistent listening** during transmission

Non-persistent CSMA

Example



Collisions are less likely to occur here, but time may be lost after the end of the ongoing transmission.
Peak throughput is a much better: $\simeq 81.5\%$.

p-persistent CSMA

Algorithm for a ready node

A compromise between 1-persistent CSMA and non-persistent CSMA.
Assume channels are **slotted** (but not globally synchronized).
A slot is long enough to detect for sure a collision (i.e. of duration $T = 2t_p$).

1. Sense the channel
   - If the channel is idle, then transmit with probability p
     - If message not transmitted, then wait **for one slot** and go to step 1.
   - If the channel is busy, then wait until channel becomes idle and go to step 1.

p-persistent CSMA

Performance

- Collisions are reduced, the peak throughput increases:

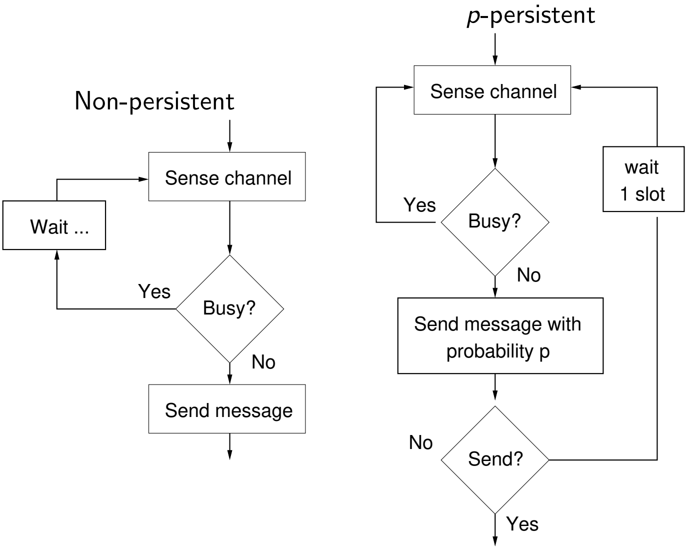
  For $p = 0.1$, $S \simeq 79.1\%$
  For $p = 0.03$, $S \simeq 82.7\%$

- But with lower p, the longer it takes to actually send a message. For given p, it takes

$$E[k] = \sum_{k=1}^{\infty} kp(1-p)^k = 1/p$$

  slots to send a message if channel is constantly idle.

Non-persistent vs. persistent



Performance of CSMA

Superiority of p-persistent/non-persistent over 1-persistent in terms of S.

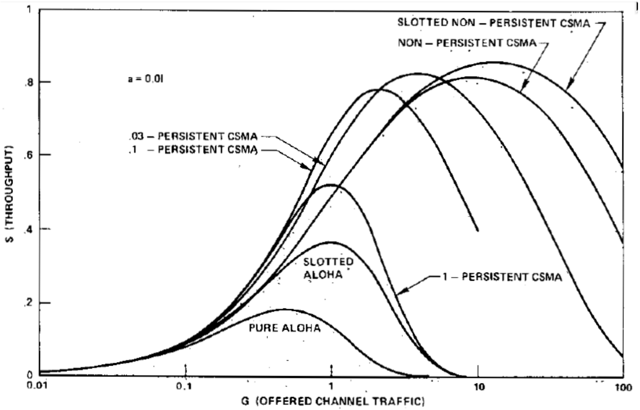


Fig. 9. Throughput for the various access modes ($a = 0.01$).

a = propagation delay / transmission delay.

Performance of CSMA[2]

But delay to transmit a message successfully increases exponentially with throughput S for non-persistent and p-persistent schemes.

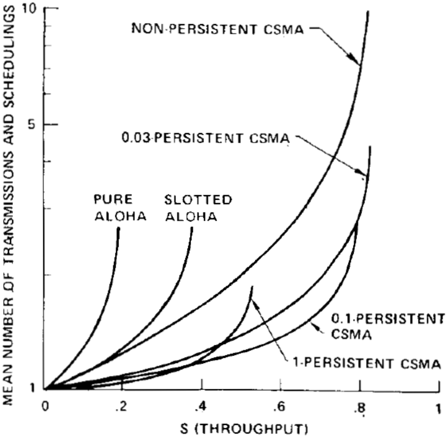


Fig. 11. G/S versus throughput ($a = 0.01$).

[2] L. Kleinrock and F. Tobagi, "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics," in IEEE Transactions on Communications, vol. 23, no. 12, pp. 1400-1416, December 1975.

Advanced CSMA protocols

CSMA/CD

A 1-persistent CSMA with an advanced collisions detection (CD) mechanism.

- 1-persistent CSMA: low peak throughput S but fast channel access in average.
- Improve throughput with CD thanks to
  - Detection of collision at sender
  - Stop transmission if collision is detected
  - Random back-off duration before new transmission attempt.

Will be detailed in Ethernet lecture.

Advanced CSMA protocols

CSMA/CA

A p-persistent CSMA with an advanced collisions avoidance (CA) mechanism made for wireless systems.

- p-persistent CSMA: larger peak throughput S but slower channel access time in average.
- Reduce channel access time with CA operations:
  - Detection of collision **at receiver**
  - Acknowledgement message to notify sender
  - Random back-off duration before new transmission attempt, with back-off freeze.

Will be detailed in WiFi lecture.

Advanced CSMA protocols

CSMA/CR for priority-based channel access

A CSMA where the contention resolution procedure elects the **highest priority message**.

- Each message gets a unique identifier (ID) representative of its priority. The lower the ID, the higher the priority.
  - If two messages of different IDs are sent concurrently, the one with the lowest ID wins channel access.
  - The sender of the lower priority message defers transmission until channel gets idle again.

Runs on any car you're driving...

Advanced CSMA protocols

CSMA/CR for priority-based channel access

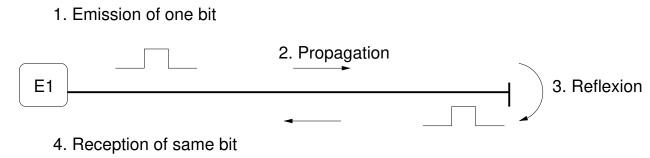
How does it work ?

- All senders are synchronized at bit-level.
- A bit can either be **recessive** or **dominant**.
  Dominant bit wins over recessive bit.
- Logical values
  - Recessive $\rightarrow$ bit value 1
  - Dominant $\rightarrow$ bit value 0

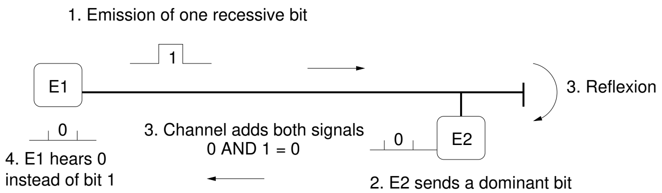
CSMA/CR

Bit-level contention resolution

For each transmitted bit, sender checks whether it stayed unchanged.



If bit unchanged, then sender keeps sending, else it stops.

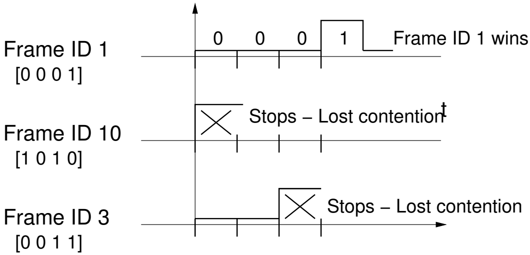


E2 keeps sending and E1 stops.

CSMA/CR

Priority with contention resolution

Each node sends its ID (i.e. priority) in the header of the message, with big-endian encoding (most significant bit first).



Highest priority message never waits. Others wait for higher priority messages to be transmitted first.
Throughput is limited by the maximum length of the wire.