Politecnico di Milano

Facoltà di Ingegneria dell'Informazione

5 – Multiple Access

Reti Mobili Distribuite

Prof. Antonio Capone

Packet access

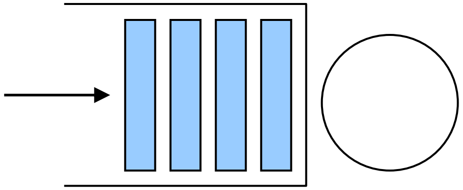
- At logical layer multiple access can be managed in a dynamic and distributed way using multiple access protocols
- First multiple access protocols have been designed for LANs
- Nowadays multiple access protocols are mainly used in wireless networks (no more shared medium wired LANs)

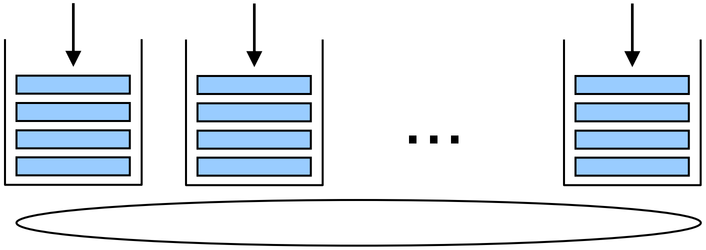
Packet access: Classification

- Scheduled access
  - Transmissions on the channel are sequential with no conflicts
  - Polling schemes
  - Centralized scheduling schemes
- Random access
  - Transmission are partially uncoordinated and can overlap (collision)
  - Conflicts are resolved using distributed procedures based on random retransmission delay

Packet access: Local and global queues

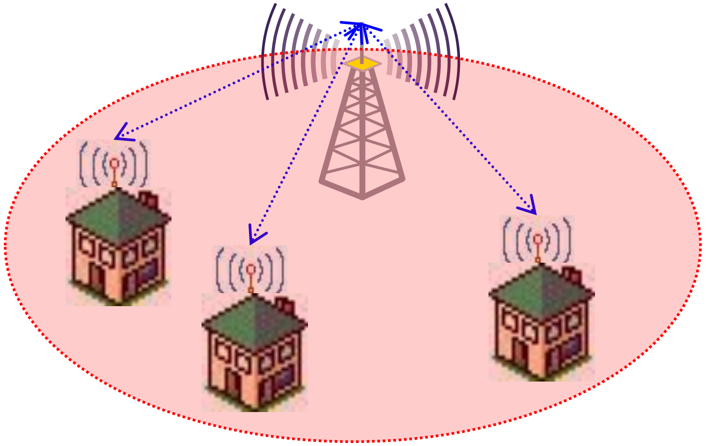
- In the case of multiplexing (single station) we have a single queue that is managed according to a scheduling algorithm
- In case of multiple access with *M* stations with local queues we still have the opportunity to use a single “virtual” queue at a central decision point that schedule access to the channel





Packet access: Local and global queues

- However, to inform the scheduler of the status of the local queues and provide access grants we have to use the channel (coordination signaling)
- The centralized scheduling approach is quite flexible but complex
- It is adopted in several wireless technologies like e.g. WiMax



Assumptions and notation

- In the following we drop the assumption of global coordination and analyze distributed mechanisms
- Let us assume that arrival times in the M local queues are described by a Poisson process with rate λ/M (λ global rate)
- The system status is described by vector

$$\mathbf{n} = \left(n_1, n_2, ..., n_M\right)$$

- Where n_i is the number of packets in queue i
- The system evolution is described by the process $N(t)$

Polling

- Polling schemes are scheduled access schemes where stations access the channel according to a cyclic order
- The polling message, or *token*, is the grant for access the channel
- The token can be distributed by a central station (roll-call polling) or passed from station to station (hub polling or token system)
- Let us assume that packet transmission time is T and that token passing time is h, both constant
- Polling schemes differentiate based on the service policy (exhaustive, gated, limited)

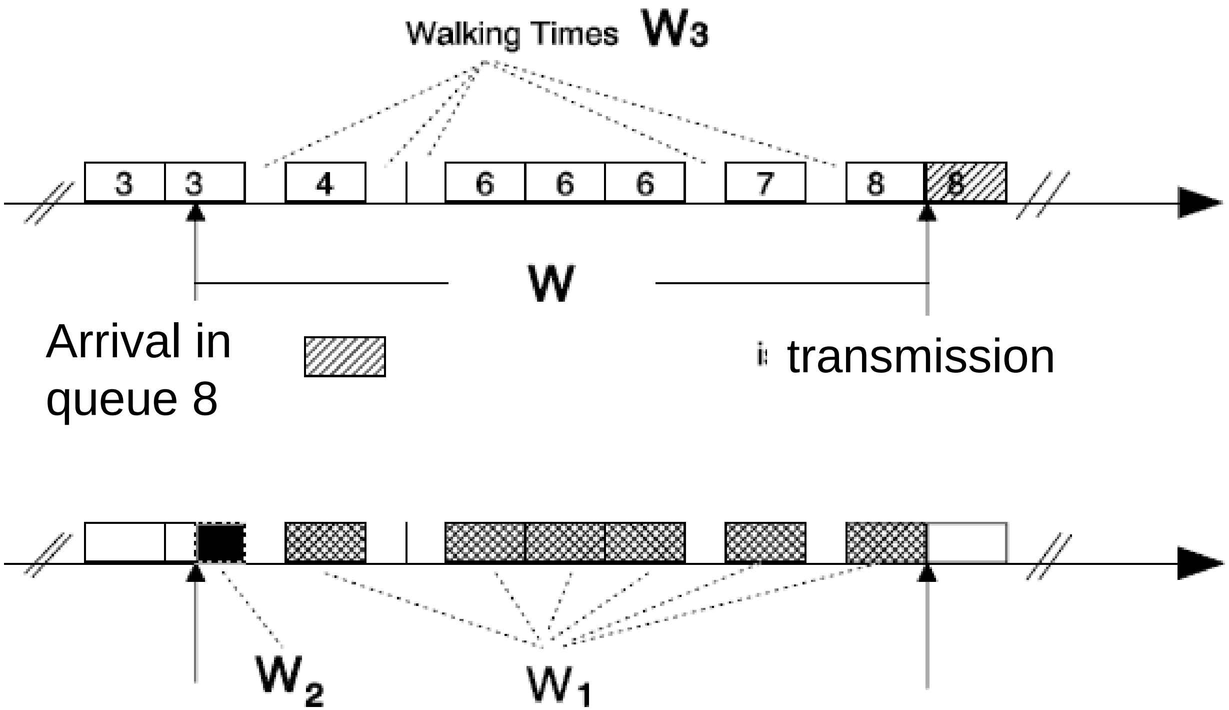
Exhaustive Polling

- With exhaustive polling, stations when receive the token transmit all packets in the queue before releasing it
- Let us analyze the behavior of this system
- The probability that the channel is transmitting a packet at a random time t is give by

$$\rho = \lambda T$$

Exhaustive Polling

- The average waiting time E[W] in the queue can be calculated considering three components $E[W] = W_1 + W_2 + W_3$

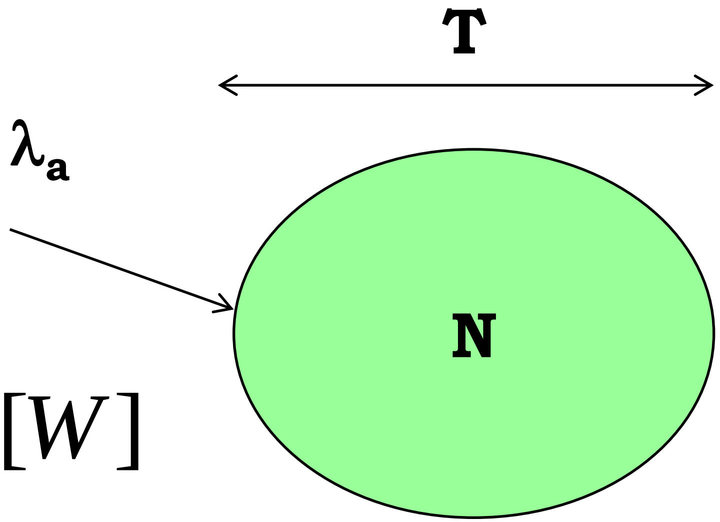


Exhaustive Polling

$$W_1 = E[N_c]T$$

- E[Nc] is the average number of packets transmitted before considered packet
- Using Little's result is can be expressed as:

$$E[N_c] = \lambda E[W]$$



- Therefore:

$$W_1 = \lambda T E[W] = \rho E[W]$$

Exhaustive Polling

$$W_2 = \rho \frac{T}{2} + (1-\rho)\frac{h}{2}$$

$$W_3 = \frac{M-1}{2} h$$

- The total average waiting time is given by:

$$E[W] = \rho E[W] + \rho \frac{T}{2} + (1-\rho)\frac{h}{2} + \frac{(M-1)}{2} h$$

Exhaustive Polling

- Solving by E[W] we get:

$$E[W] = \underbrace{\frac{\rho}{2(1-\rho)}T}_{\text{Waiting time of a single queue (M/D/1)}} + \underbrace{\frac{M-\rho}{2(1-\rho)}h}_{\text{Additional waiting time due to token passing time}}$$

- Note that:

$$\rho_{\max} = 1$$

Exhaustive Polling

- The average token cycle time is given by the transmission time of all packets that arrive during a cycle plus the token passing time

$$E[C] = \lambda E[C]T + Mh$$

$$E[C] = \frac{Mh}{1-\rho}$$

Gated Polling

- With gated polling, stations when receive the token can transmit all packets that are in queue at the time when the token arrives
- The expression of the average waiting time is similar to previous case with an additional term
- This is the additional cycle the packet has to wait when it arrives when the token is already at the station

$$W_4 = \frac{\rho}{M} Mh = h\rho$$

Gated Polling

- Therefore we get:

$$E[W] = \frac{\rho}{2(1-\rho)} T + \frac{M+\rho}{2(1-\rho)} h$$

- Again $\rho_{\max} = 1$

Limited Polling

- With limited polling, stations when receive the token can transmit only up to *k* packets
- The special case of *k=1* is called *Round-Robin*
- Here we have one more additional term which are the additional cycles the packet has to wait, one per each packet in the queue at the arrival moment

$$W_5 = \frac{E[N_c]}{M} Mh = \lambda E[W]h$$

Limited Polling

- Therefore we get:

$$E[W] = \frac{\rho}{2(1-\rho\frac{h+T}{T})}T + \frac{M+\rho}{2(1-\rho\frac{h+T}{T})}h$$

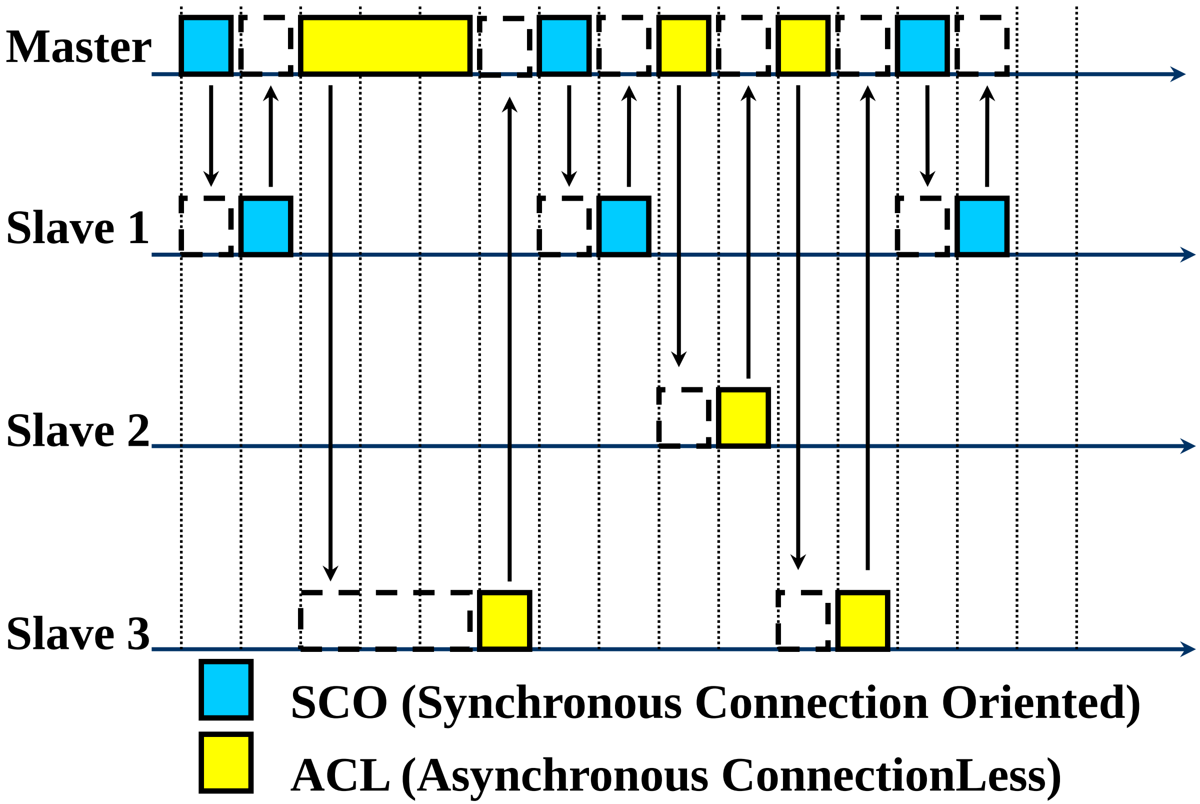
- Now we have:

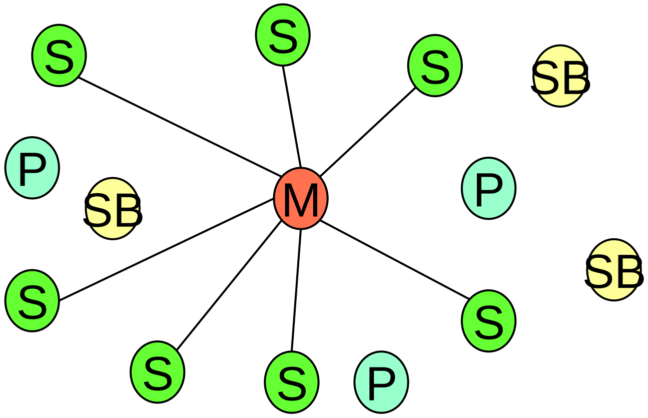
$$\rho_{\max} = \frac{T}{T+h}$$

Polling in real networks

- There are several examples where polling is used for regulating access to a channel in wireless technologies
  - WiFi (Point Coordination Function – PCF or HCF – Hybrid Coordination Function)
  - Bluetooth
- The main difference with simple schemes we considered so far is that the station sequence can be dynamically changed

Polling in Bluetooth





Polling in Bluetooth

- Some key characteristics of Bluetooth multiple access mechanism make the direct application of previously derived formulas not possible:
  - Queues are not visited in a sequential order (master queue is always visited in odd slots)
  - Token passing time is always one slot, but the slot is used for data transmission if the queue is not empty
  - Exhaustive service makes no sense for Bluetooth since after each packet transmission by the master/slave at least a slot is used by the slave/master
  - Bluetooth makes use of packets with different lengths (1, 3, or 5 slots)
- We derive expressions for the waiting time in two special cases

Polling in Bluetooth

- Let us assume the master has one separate queue per slave and all queues are visited according to a fixed sequence
- Arrival in the queues are independent Poisson processes

m: number of BT devices

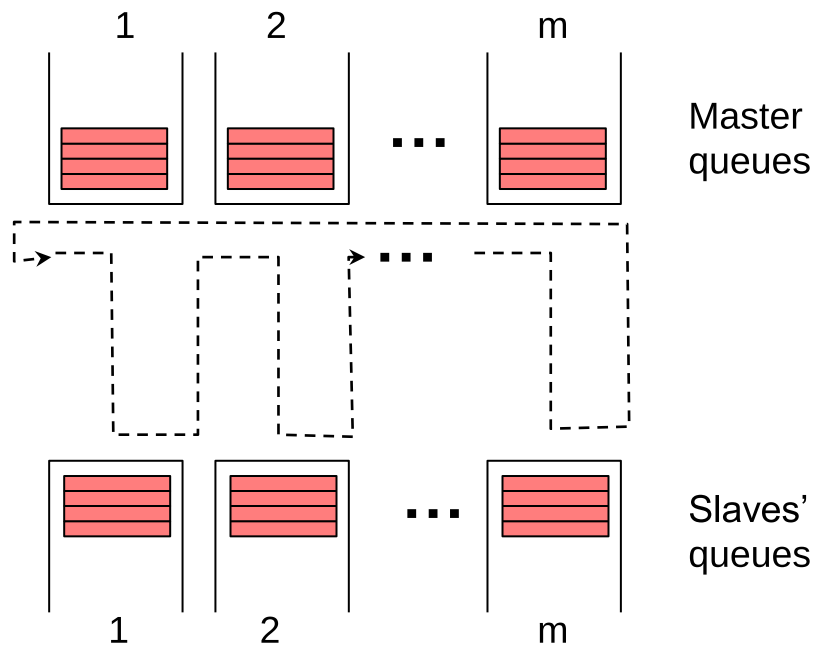
m-1: slaves

M=2(m-1): total queues

γ: arrival rate in each queue

λ: total arrival rate

T: slot duration

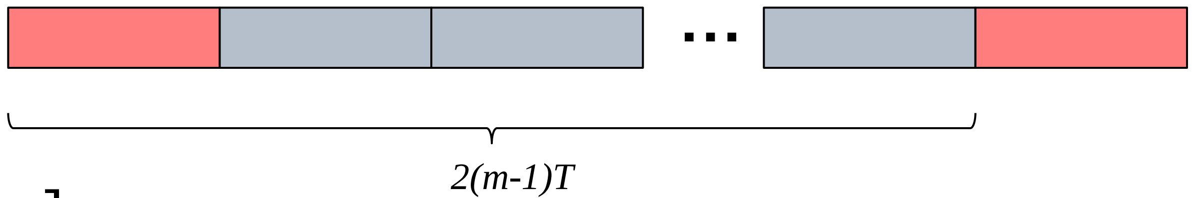


Polling in Bluetooth

- Case 1)
  - 1-limited service (round-robin)
  - 1-slot packets only
- We observe that
  - Cycle length is fixed and equal to 2(m-1) slots
  - System is equivalent to a TDMA with 2(m-1) slots per frame
  - There are several equivalent ways of calculating the waiting time
  - We use the same approach adopted for the general polling schemes

Polling in Bluetooth

- Case 1)



$$W_1 = \gamma T E[W]$$

$$W_2 = \frac{2(m-1)T}{2} = (m-1)T$$

$$W_3 = 0$$

$$W_4 = 0$$

$$W_5 = \gamma T E[W][2(m-1)-1]$$

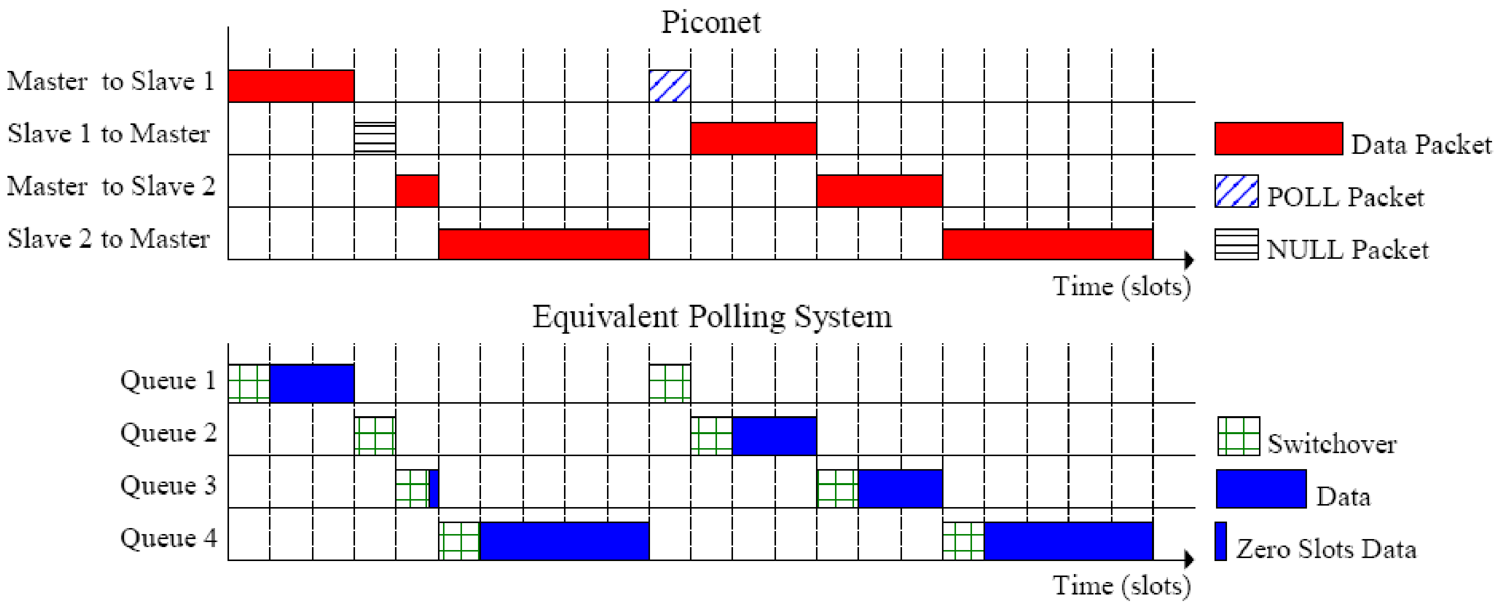
$$E[W] = \frac{(m-1)T}{1-\gamma T - 2(m-1)\gamma T + \gamma T} = \frac{(m-1)T}{1-2(m-1)\gamma T}$$

Polling in Bluetooth

- Case 2)
  - 1-limited service (round-robin)
  - 1, 3, and 5-slots packets
- We observe that:
  - The system is equivalent to a polling system with:
    - Token passing time equal to 1 slot
    - Service time equal to packet length minus one slot

Polling in Bluetooth

- Case 2)
  - 1-limited service (round-robin)
  - 1, 3, and 5-slots packets



Polling in Bluetooth

- Case 2)
  - Notation:

$$p_1 : \text{prob. of 1-slot packets}$$

$$p_3 : \text{prob. of 3-slot packets}$$

$$p_5 : \text{prob. of 5-slot packets}$$

$$L : \text{packet lenght}$$

$$\overline{L} = p_1 + 3p_3 + 5p_5$$

$$X : \text{service durantion in the equiv. system}$$

$$\overline{X} = 2p_3 + 4p_5$$

$$\overline{X}^2 = 4p_3 + 16p_5$$

$$E[z] = \frac{\overline{X}^2}{2\overline{X}}$$

Polling in Bluetooth

- Case 2)
  - Waiting time:

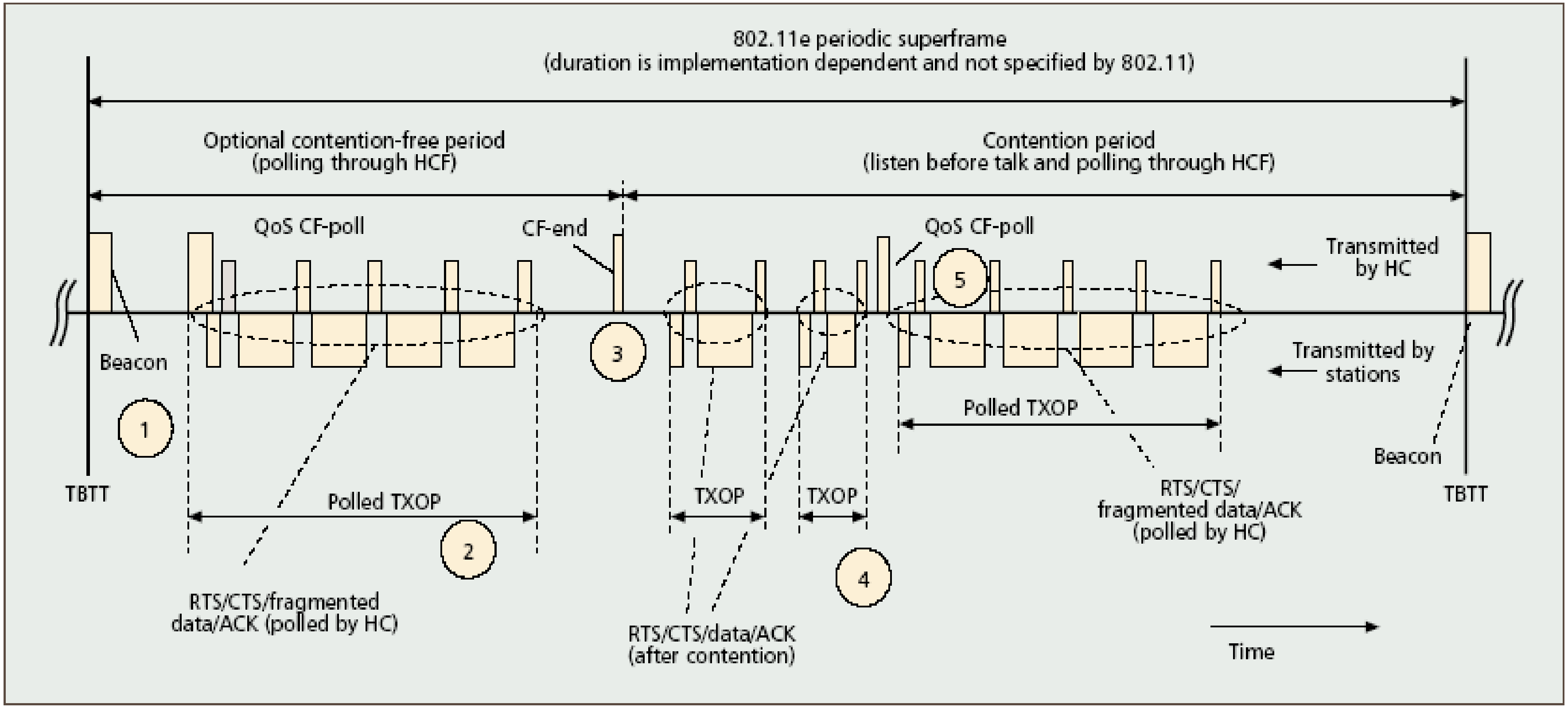
$$E[W] = \frac{\rho}{1-\frac{(L-1+1)T}{LT}\rho}E[z] + \frac{M+\rho}{2\left(1-\frac{(L-1+1)T}{LT}\rho\right)}T =$$

$$= \frac{\rho}{1-\frac{L}{L-1}\rho}E[z] + \frac{2(m-1)+\rho}{2\left(1-\frac{L}{L-1}\rho\right)}T =$$

$$= \frac{2(m-1)\gamma(L-1)T}{1-2(m-1)\gamma LT}\frac{\overline{X}^2}{2(L-1)} + \frac{2(m-1)+2(m-1)\gamma(L-1)T}{2\left(1-2(m-1)\gamma LT\right)}T =$$

$$= \frac{(m-1)\gamma}{1-2(m-1)\gamma L}\overline{X}^2 + \frac{(m-1)+(m-1)\gamma(L-1)T}{1-2(m-1)\gamma LT}T =$$

Polling in WiFi (IEEE 802.11e)

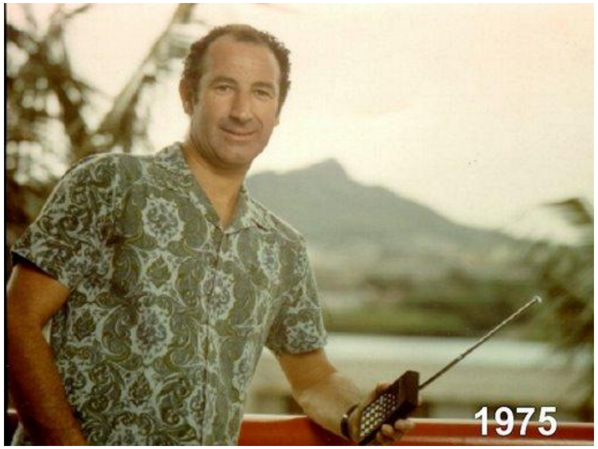


Random access

- With random access there are possible conflics on the channel (collisions)
- Conflicts are resolved using the channel feedback and some procedure to select a random waiting time
- The minimum channel feedback a station need to have is the information if its transmission was sccessful or not
- The first and simplest random access protocol is **Aloha** which uses just this minimum feedback

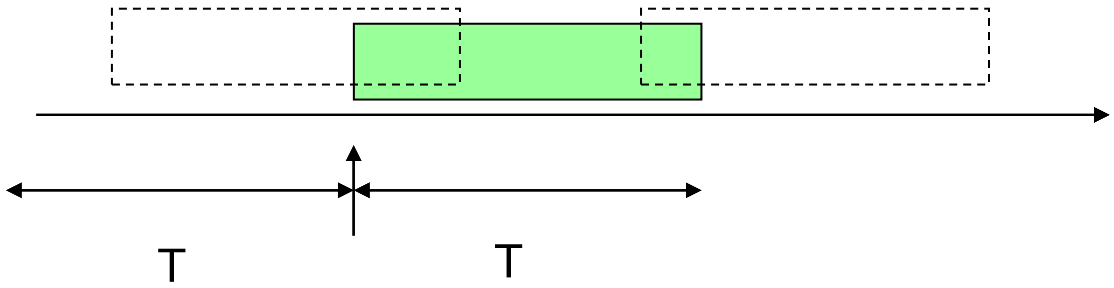
AlohaNet

- The ALOHA network was created at the University of Hawaii in 1970 under the leadership of Prof. Norman Abramson
- It was the first wireless network!



Aloha

- The access mechanism is very simple:
  - When there is a packet to be transmitted, just transmit it.
  - If transmission fails, wait for a random time and retransmit



Aloha

- Let us assume the transmission starting times on channel are a Poisson process with rate λ
- Let us consider the normalized rate $G=\lambda T$
- The success probability is given by the probability that there is no other transmission in a $2T$ interval

$$P_s = e^{-2G}$$

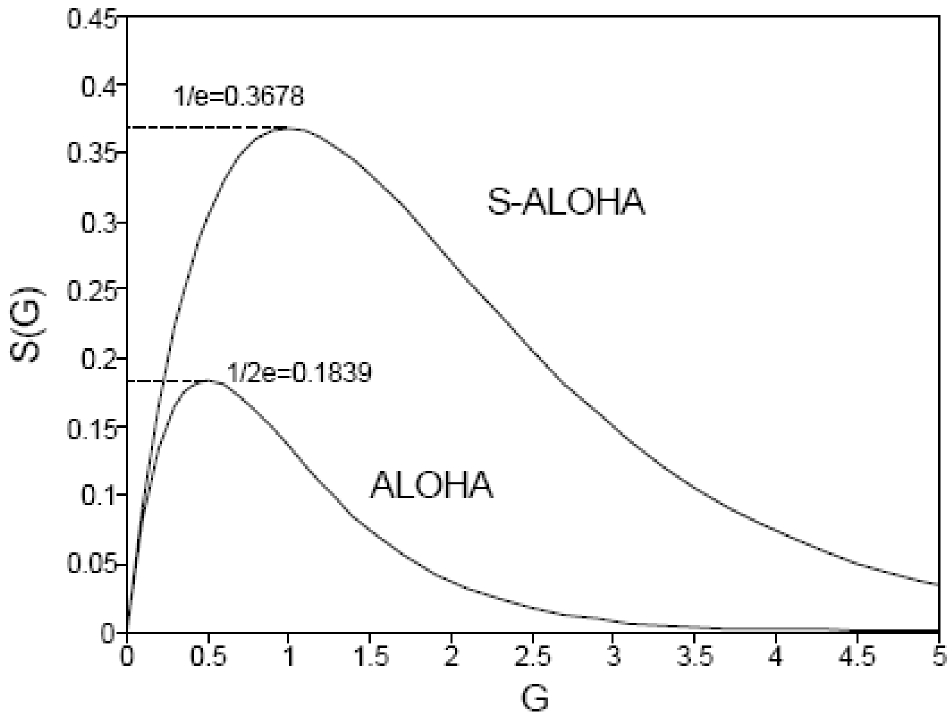
- The normalized throughput S is therefore given by:

$$S = Ge^{-2G}$$

Aloha

- If transmissions are somehow synchronized (slotted Aloha) the vulnerability period reduces to T and therefore

$$S = Ge^{-G}$$



Infinite population model

Aloha: Single buffer

- Unfortunately, the traffic on the channel is the combination of new transmissions and retransmissions and it can increase if throughput reduces
- To evaluate the dynamic behavior of Aloha let us consider an enhanced model
- Let us assume we have *M* stations with a single transmission buffer (max 1 packet)
- Channel is slotted
- If buffer is empty a new packet arrive and is immediately transmitted in the slot with probability α
- If buffer is full, packet is retransmitted with probability β

Aloha: Single buffer

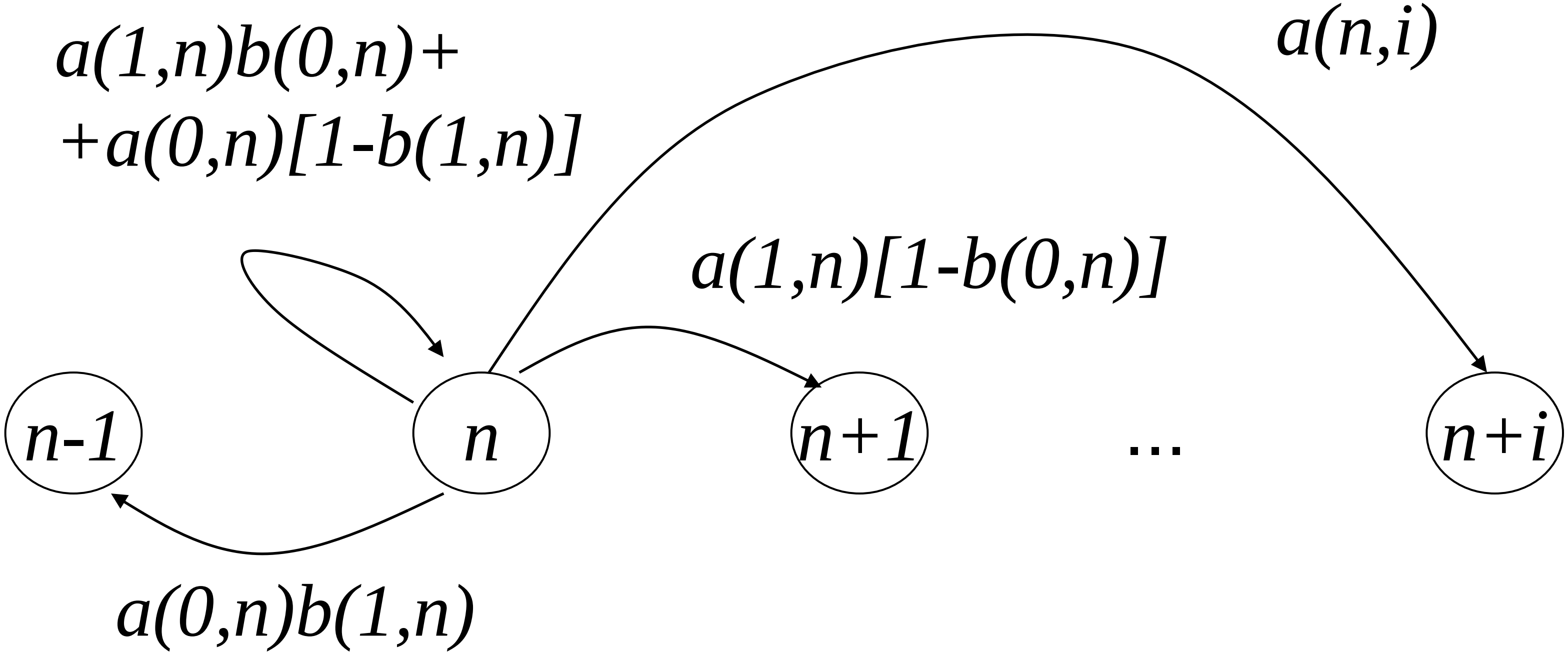
- The system status is given by $n(t)$, the number of full buffers at time t
- $n(t)$ is a discrete Markov chain
- The probability that in a slot there are i arrivals given n buffers are full is:

$$a(i,n) = \binom{M-n}{i} \alpha^i (1-\alpha)^{M-n-i}$$

- While the probability that there are i retransmission is:

$$b(i,n) = \binom{n}{i} \beta^i (1-\beta)^{n-i}$$

Aloha: Single buffer



Aloha: Single buffer

- We can solve (numerically) the chain and get the stationary state probability π_n
- The throughput S is given by

$$S = E[s(n)] = \sum_{n=0}^{M} \pi_n s(n)$$

$$s(n) = a(1,n)b(0,n) + a(0,n)b(1,n)$$

Aloha: Single buffer

- The traffic on the channel is:

$$g(n) = (M - n)\alpha + n\beta$$

- and the new arrivals:

$$a(n) = (M - n)\alpha$$

- We can express *n* as a function of *g*:

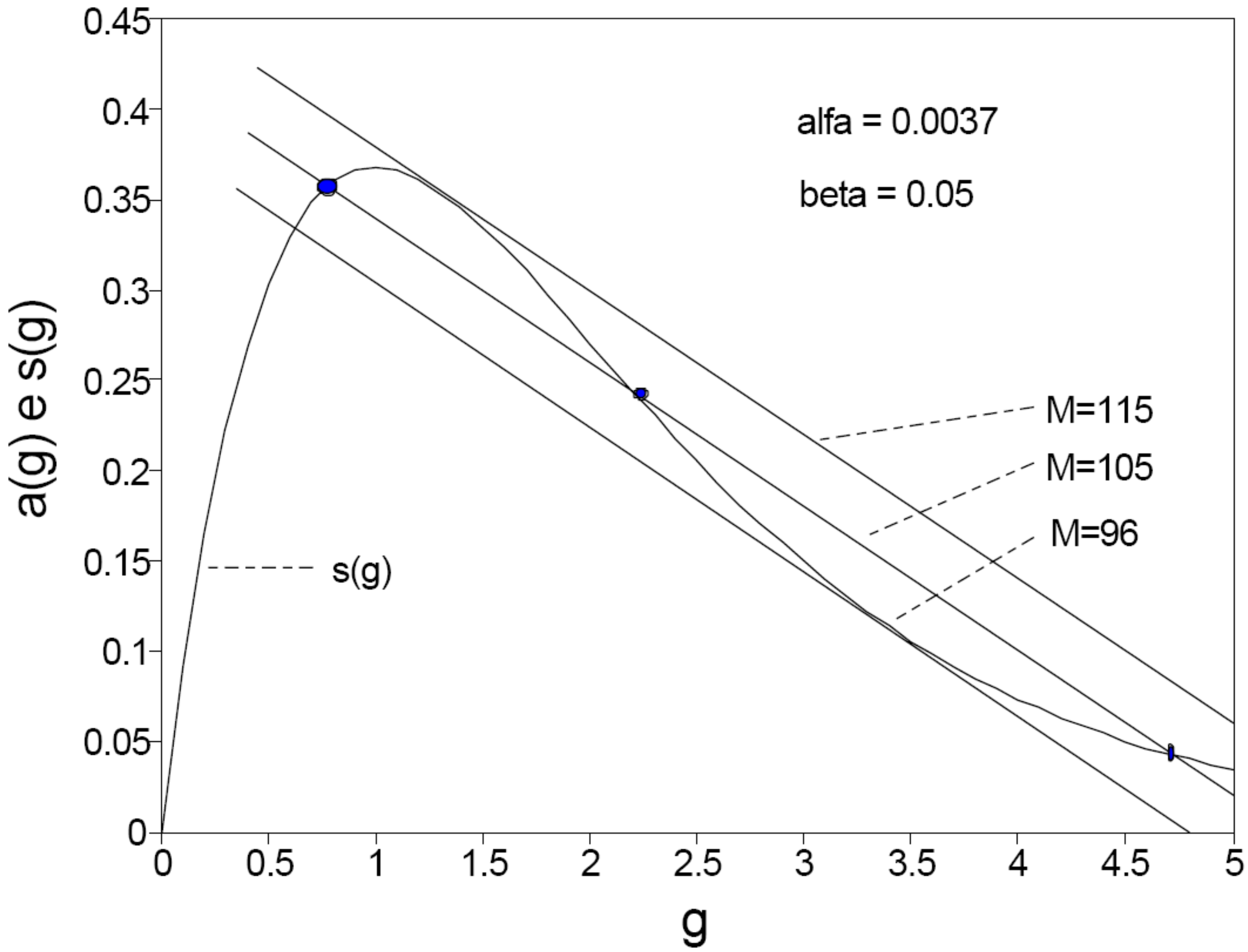
$$n = \frac{g - M\alpha}{\beta - \alpha}$$

- And calculate $a(g) = M\alpha - (g - M\alpha)\dfrac{\alpha}{\beta - \alpha}$

Aloha: Single buffer

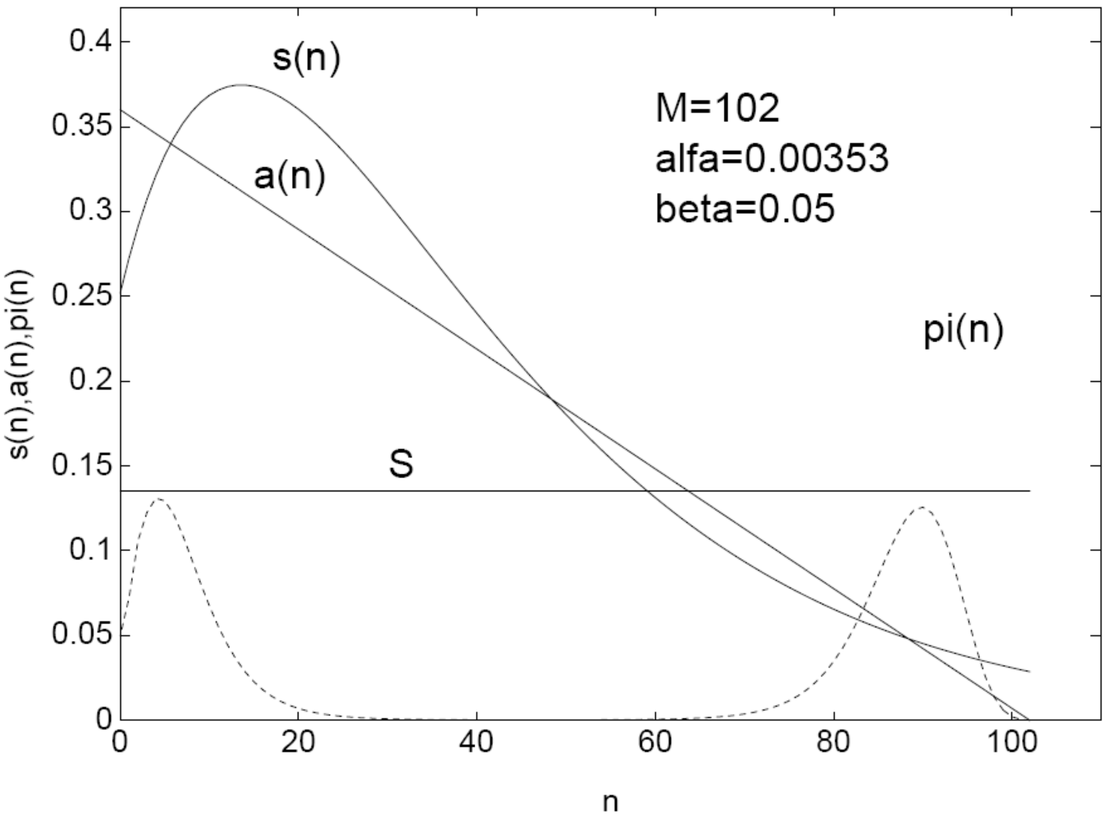
- Similarly, we can also get *s(g)* using stationary probabilities (the curve is very close to that of the infinite population model)
- We can consider points where *s(g)=a(g)* as equilibrium points of the process
- Equilibrium points can be "stable" or "unstable"

Aloha: Single buffer

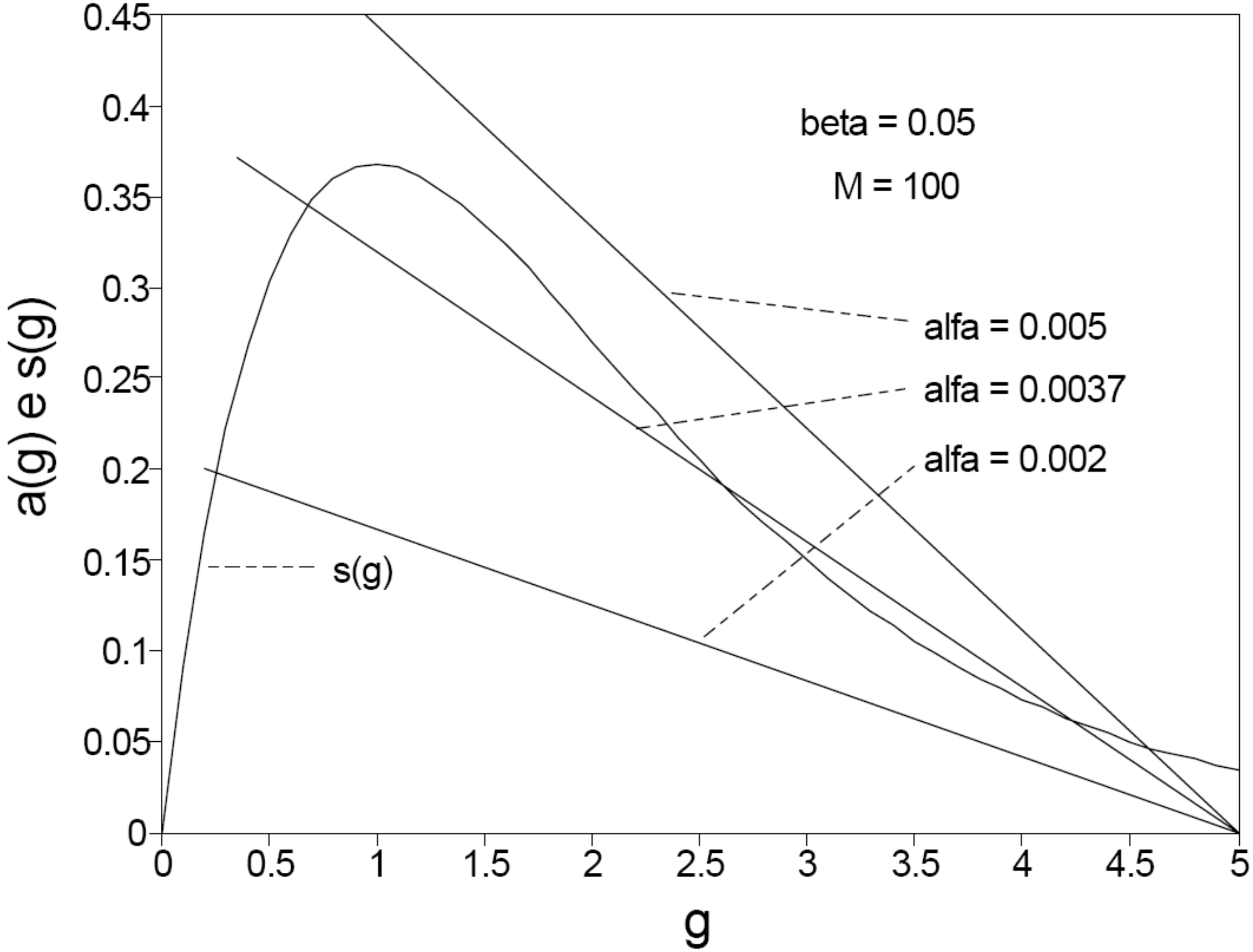


Aloha: Single buffer

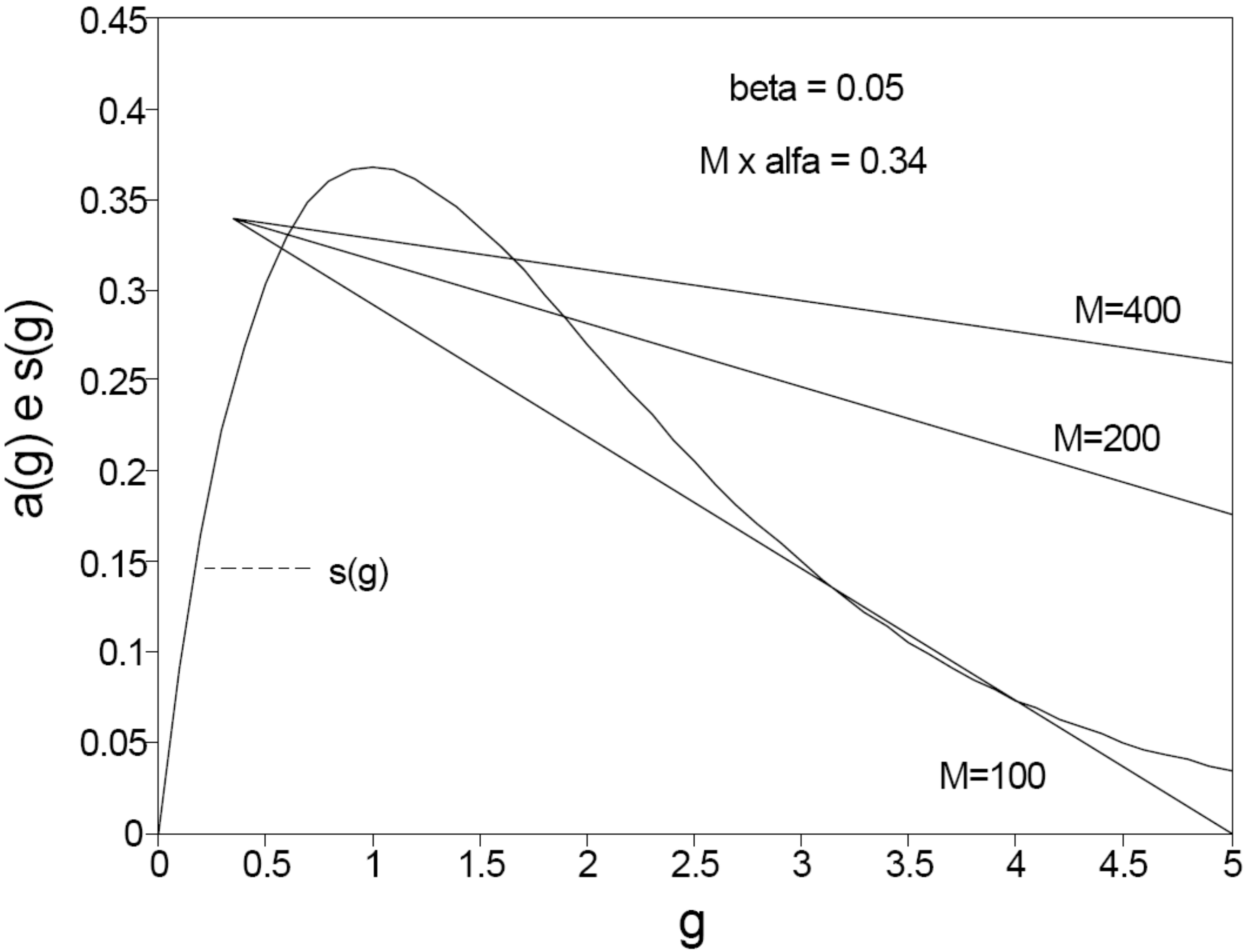
- What are really equilibrium points?



Aloha: Single buffer



Aloha: Single buffer



Aloha stability

- If more than one equilibrium point exists, we cannot guarantee the system has a stable throughput
- Il traffic is low (number of stations) throughput is stable
- To stabilize Aloha the only way is to limit traffic on the channel (keeping it out of the system if necessary)
- With the minimum feedback considered so far, stabilizing the system is not possible

Stabilized Aloha

- Let’s assume a richer feedback from the channel: stations know at the end of each transmission interval if it was empty(E), if there was a successful transmission (S), or a collision (C)
- Let’s also assume the slotted Aloha case for simplicity and the single buffer model
- In order to control traffic on the channel we need to use transmission probability
- Let’s assume the same probability β for new transmissions and retransmissions

Stabilized Aloha

- The traffic on the channel is given by $G=n\beta$
- Assuming we know the number n of busy buffers, the best strategy to limit traffic is to set dynamically $\beta=1/n$ so that $G=1$ and throughput is maximized
- However, the number n of busy buffers cannot be known directly by stations and needs to be estimated at each slot k
- Let's denote with $\widehat{n}_k$ the estimated value and with $\beta_k = 1/\hat{n}_k$ the probability value at slot k

Stabilized Aloha

- Let's assume n_k is distributed according to a Poisson distribution and let's set its average value to $\widehat{n}_k$

$$P(n_k = i) = \frac{\widehat{n}_k^{\,i}}{i!} e^{-\widehat{n}_k}$$

- This is the 'a priori' distribution of n_k
- Due to channel feedback, after a slot, the distribution 'a posteriori' is modified
- This is obviously different depending on the observed feedback

Stabilized Aloha

- If we observe a successful transmission (S) the obvious update rule is:

$$\widehat{n}_{k-1} = \widehat{n}_k - 1 + S$$

- Since 1 packet leaved the system and, on average, S new arrivals increased the number of busy buffers

Stabilized Aloha

- If we observe an empty slot (E), we have:

$$P(n_k = i / E) = \frac{P(n_k = i; E)}{P(E)} = \frac{(1-\beta)^i \dfrac{\hat{n}_k^i}{i!} e^{-\hat{n}_k}}{e^{-G}}$$

- Since the mechanism keeps $G=1$ with $\beta_k = 1/\hat{n}_k$

$$P(n_k = i / E) = \frac{(\hat{n}_k - 1)^i}{i!} e^{-(\hat{n}_k - 1)}$$

- which is equal to the 'a priori' distribution with an average value of $\hat{n}_k - 1$
- So the update rule is again: $\hat{n}_{k-1} = \hat{n}_k - 1 + S$

Stabilized Aloha

- Since the update rule is the same we can consider a single event type, we can call no-collision (NC)
- Therefore the feedback we have is just

$$[C; NC]$$

- This is usually called binary channel feedback

Stabilized Aloha

- If we observe a collision (C), we could calculate the ‘a posteriori’ distribution as in previous case.
- However, since the system need to be in equilibrium we can simply write that:

$$P(E)x_e + P(S)x_s + P(C)x_c = -1/e$$

- Where x are the variation in the estimated n_k (without including new arrivals)

Stabilized Aloha

- We have

$$x_e = x_s = -1$$

$$P(E) = e^{-G}\Big|_{G=1} = e^{-1}$$

$$P(S) = Ge^{-G}\Big|_{G=1} = e^{-1}$$

$$P(C) = 1 - e^{-G} - Ge^{-G}\Big|_{G=1} = 1 - 2e^{-1}$$

$$\text{therefore: } x_c = \frac{1}{e-2}$$

- and the updated rule is $$\widehat{n}_{k-1} = \widehat{n}_k + \frac{1}{e-2} + S$$

Stabilized Aloha

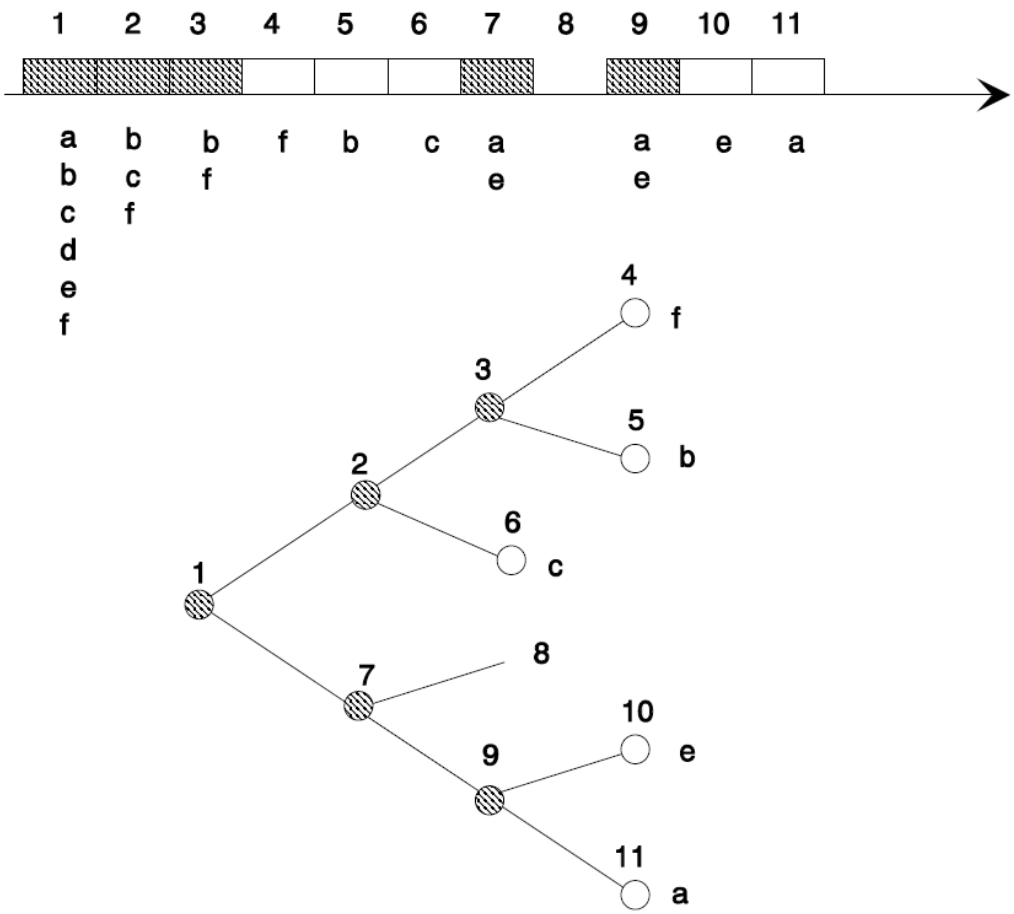
- It is possible to show that the system is stable and that the most convenient value for S is

$$S = e^{-1}$$

Collision Resolution Algorithms

- Another very popular way of stabilizing Aloha that provides a performance (throughput) much higher than in previous case is based on the so called stack algorithms
- The idea is to solve the collision among a group of stations (all packets successfully transmitted) before moving on with the access mechanism
- The simplest version of CRA is based on binary trees
- After a collision stations are divided in two groups randomly
- The first group retransmits while the other waits until all the packets of the first group are successfully transmitted

Collision Resolution Algorithms

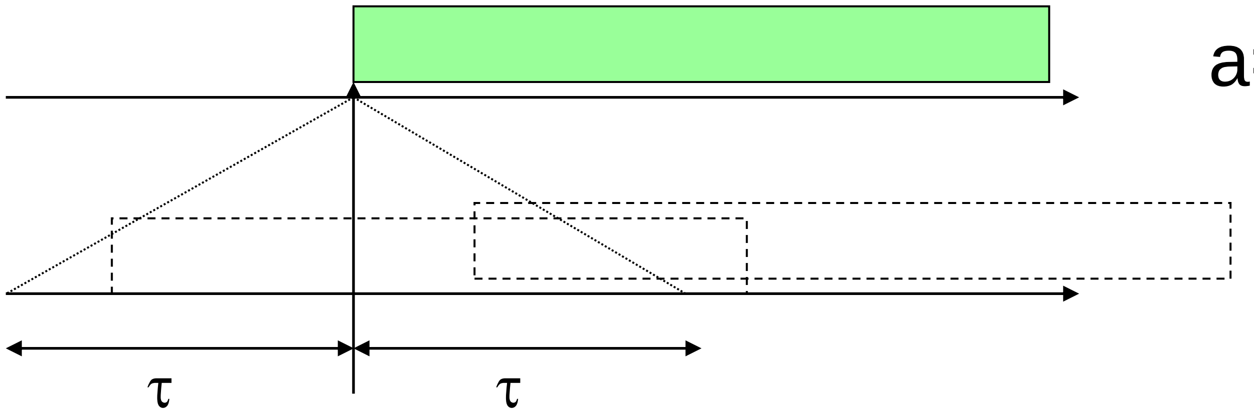


Aloha in real networks

- There are several technologies where Aloha is still adopted including:
  - Random access signaling channel of cellular systems (like e.g. GSM)
  - Reservation channel of WiMax
  - RFID

Carrier Sense Multiple Access (CSMA)

- If the channel feedback is richer, more efficient random access mechanisms can be adopted
- If the propagation time is short wrt to transmission time we can sense the channel status
- CSMA: like Aloha but transmit only when you sense the channel free



$a=\tau/T$

Carrier Sense Multiple Access (CSMA)

- On the channel we have cycles of Busy (at least one station sense the channel as busy) and Idle (all stations sense the channel free) periods
- The throughput S can be given by:

$$S = \frac{\alpha}{B + I}$$

- where B and I are the average busy and idle periods and α is the probability that there is a success transmission in a busy period

Carrier Sense Multiple Access (CSMA)

- Making the same assumptions of the aloha infinite population model we have:

$$\alpha = e^{-aG}$$

$$I = \frac{1}{G}$$

$$B = e^{-aG}(1+a) + (1-e^{-aG})(1+a+Z)$$

- where Z is the time when colliding transmission partially overlap

Carrier Sense Multiple Access (CSMA)

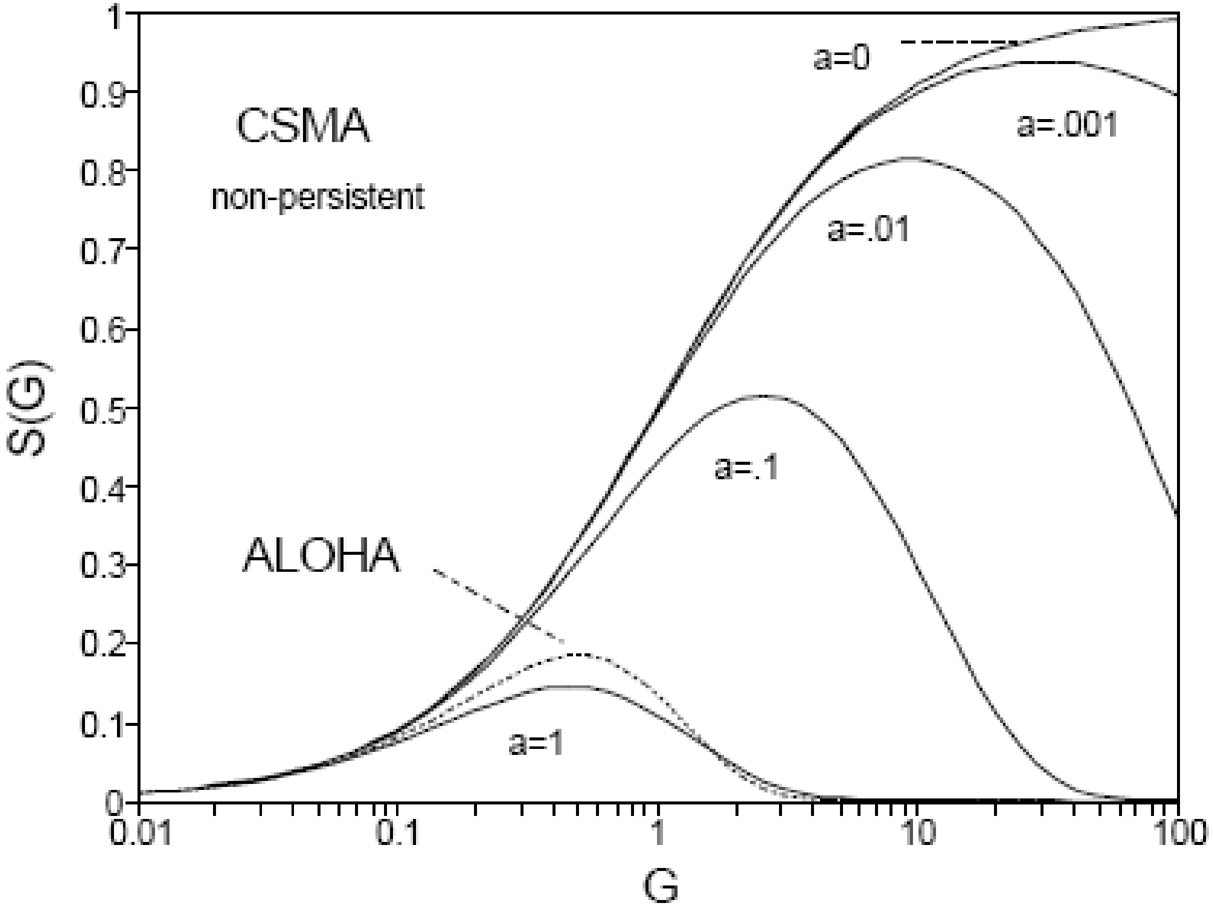
- It can be shown that:

$$Z = a + \frac{ae^{-aG}}{1-e^{-aG}} - \frac{1}{G}$$

- Therefore we get:

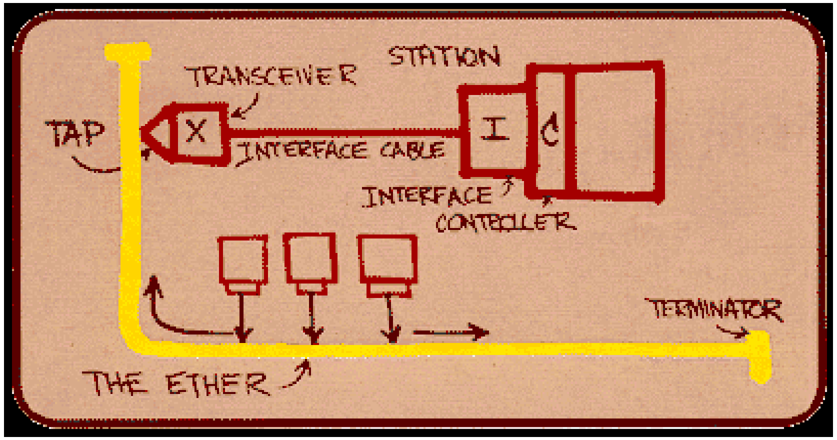
$$S = \frac{Ge^{-aG}}{G(1+2a)+e^{-aG}}$$

Carrier Sense Multiple Access (CSMA)



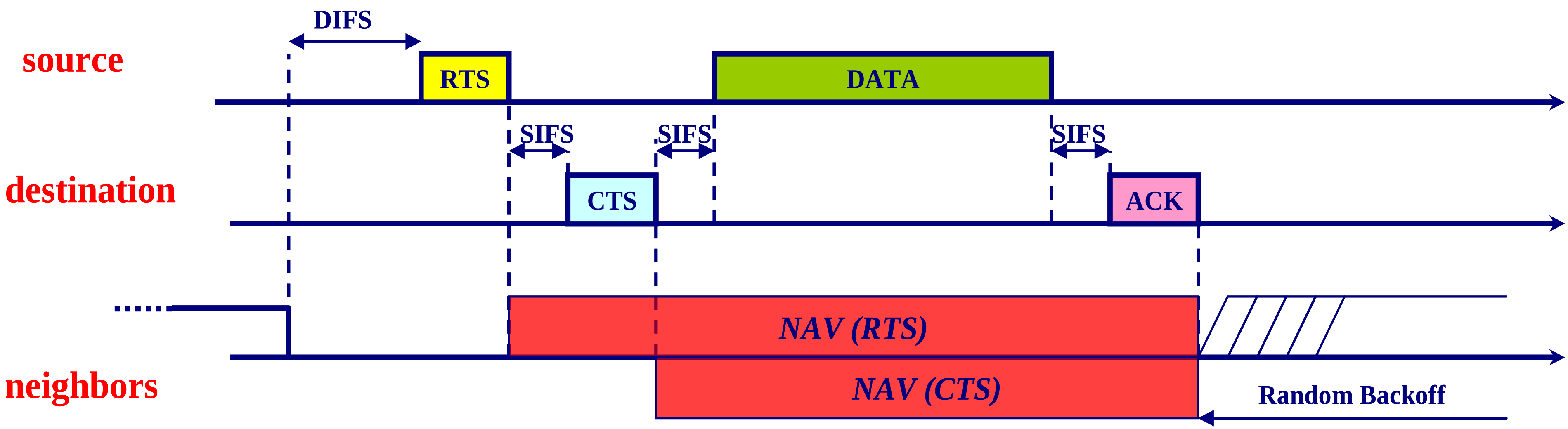
CSMA in real networks

- There are several technologies that are based on variants of the CSMA protocol
- Including Ethernet



- Today the most famous and widely used one is WiFi

IEEE 802.11 random access



IEEE 802.11 random access

- Similarly to the general model we can derive a model for WiFi

$$\alpha = e^{-aG}$$

a = interframe space
b = duration of RTS and CTS

$$I = \frac{1}{G}$$

$$B = e^{-aG}(1+3a+2b) + (1-e^{-aG})(b+a+Z)$$

$$S = \frac{Ge^{-aG}}{G(1+2a) + e^{-aG} - G(1-b)(1-e^{-aG}) + (2a+2b)Ge^{-aG}}$$