TDS Practice: Slotted Aloha Specification Protocol & Model

Dolors Sala

Aloha

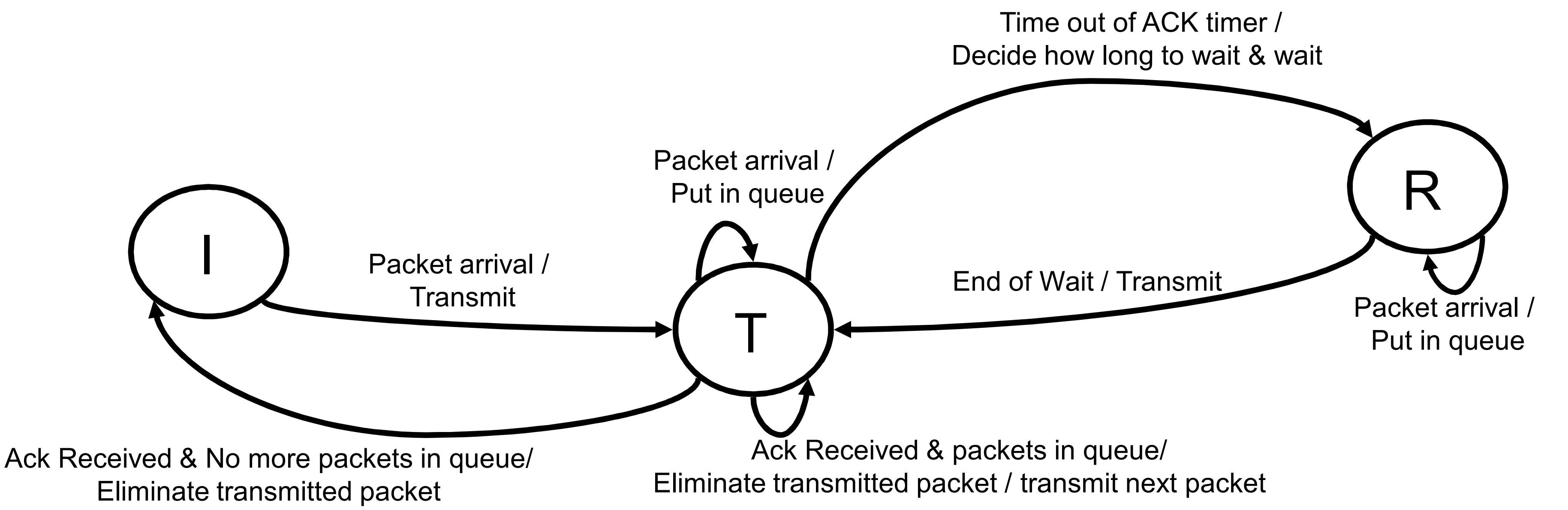
- Notation:
  - User: the station trying to communicate in the channel with another user/station
    - Wants to transmit packets
    - Wants to receive packets
  - Channel: transmission media that is broadcast and shared between all users to communicate
  - Sender: the user sending a packet to a destination user in the network
  - Destination: the user who has to receive the transmitted packet

Aloha : Protocol Definition

- Basic operation: Transmit a packet, the destination will send you an ack if received, if the ack reaches you the packet has been delivered correctly so it can be deleted and start again the process with the next packet in the queue
  - The sender waits for the ack for a time duration (set in a timer)
- Since the media is shared more than one user can transmit at the same time. In this case, the transmitted packets collide and the transmission cannot be received at the destination and sender will not receive the ack, and the ack timer expires which is the indication of a collision
- If collision happens users cannot transmit immediately all at the same time as the collision will repeat for ever
  - Each user has to wait (ideally) a different time before retransmitting
  - How long to wait is decided by the contention resolution algorithm (CRA)

Aloha : User State Diagram

- The user operation is implemented with a state diagram of 3 states:
  - Idle (I): initial state indicating that there is no packet with initiated transmission, so the user is doing nothing (Idle)
  - Transmit (T): A packet has been Transmitted and user is waiting for the ack
  - Resolution (R): A collision has occurred and is waiting for the Resolution time before the next Retransmission



Slotted Aloha : Protocol Definition

Universitat Pompeu Fabra Barcelona

- Aloha protocol reaches a maximum efficiency of 18% due to collisions and partial collisions
  - A collision can be between the last bit of current packet transmitting and the first bit of a new packet being transmitted (both are lost)
- Performance is higher (double) if collisions can happen only at the start of the packet transmission
- Slotted aloha defines the slot unit and transmissions can only happen at this point
  - All arrivals during an slot must wait transmission at the beginning of the next slot
- Everything else remains the same as aloha protocol previously defined

saloha Program: Model (I)

- Assumptions made in the model are to simplify the implementation
  - They are good enough for our study (purpose)
- All packets are assumed same size
- The slot size is equal to the time to transmit a packet
- Time unit in simulation is the slot (time) → time driven simulation
  - Operations happen at slot times (finer time granularity is neglected)

saloha Program: Model (II)

- The channel is modeled with ONE slot
  - All operations happening in this slot are performed together and before the clock moves to next slot
  - Therefore no need to keep channel history when time passes to next slot

- All stations (0.. nstns-1) are sender users implementing the previous state diagram

- There is an extra station (stns[nstns]) that is the destination user of all packets sent by all stations
  - It generates the ack and delivers them to the corresponding sender station

- The contention resolution algorithm implemented is the determinist

saloha Program: Model (III)

- The acks are not created instead a the destination is directly informed (it never collides)
  - It is not transmitted in the channel as in protocol rules
  - Notification is received immediately by the sender (same slot that the packet is transmitted)
    - A packet can have a transmission delay of zero, this is inaccurate, it has to be at least one
    - The delay can be corrected by just adding one, so no need to make the model more complex for an operation that can be reduced to add a constant to a metric result

Determinist (D) CRA

- This mechanism is not an algorithm per se
  - We use it as the simplest form of not colliding indefinitely in our basic initial implementation
  - It will be used in a CRA study together with typical algorithms
- Definition:
  - Each station has an identification number `n` which corresponds to the index /position of the station in the stations vector
  - The algorithm D is called once the collision is identified
  - It returns the number of slots to wait before trying to transmit again
  - The returned value is a constant value for each station equal to the station identification number `n`
    - `station 0 waits 0, station 1 waits 1, …`

Universitat Pompeu Fabra Barcelona

Contention Resolution Algorithms (CRAs): Brief description for the assignments

Determinist (D) CRA

- This mechanism is not an algorithm per se
  - We use it as the simplest form of not colliding indefinitely in our basic initial implementation
  - It will be used in a CRA study together with typical algorithms
- Definition:
  - Each station has an identification number `n` which corresponds to the index /position of the station in the stations vector
  - The algorithm D is called once the collision is identified
  - It returns the number of slots to wait before trying to transmit again
  - The returned value is a constant value for each station equal to the station identification number `n`
    - `station 0 waits 0, station 1 waits 1, …`

Truncated Binary Exponential Backoff (B)

- This mechanism increments the waiting time of a retransmission of a packet in relation with the number of collisions occurred by the packet
  - More collisions means that currently there are more packets trying to be transmitted
  - New packets of other stations join with low backoff and colliding packets keep incrementing the backoff (unfair!)
- It applies an exponential dispersion up to a maximum to avoid an ever growing delay
- The algorithm is defined as follows:
  - Given `m` as the number of collisions suffered by a packet
  - The waiting time of a collided packet `w` (en slots) is computed as a random value between 0.. 2^t where `t` is the minimum between `m` and a límit `c` to this `m`, this is, `t = min(m,c)` where c is a determined constant value
    - Typical value of c is 10 (maximum delay of 1024)

P-persistence (P)

- The algorithm p-persistence is equally applied to the first attempt of transmission of the packet and all following retransmissions
- An station with packets in the queue decides whether to (re)transmit or wait for the following slot following a probabiliby p [0..1]
  - It generates a randon number u between 0..1. If $u < p$ it transmits otherwise it does not transmit and repeats the process the next slot
- In the implementation the P version it assumes a fix p value for all stations, and this value is an input parameter
- Note: this algorithm is different from others in two aspects:
  - The waiting is applied once the collisions has occurred (as others) but it is also applied in the first attempt of a packet
  - It does not compute how many slots the station has to wait instead it tests in every slot to transmit or not
    - If p is always constant we could compute how how many slots to wait by generating as many u numbers until is smaller than p. The number of times would be the number of slots to wait. This is not possible to compute is p is different (see next).

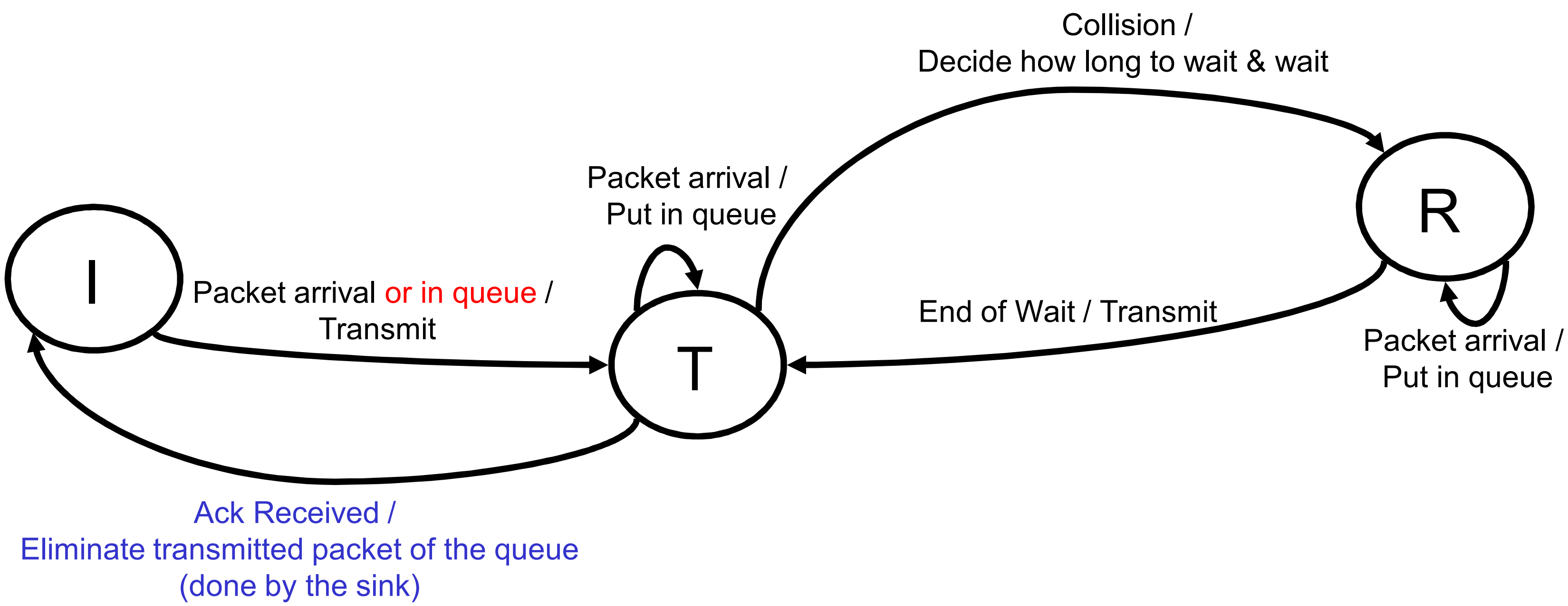
Optimal, or Ideal, p-persistence (O)

- The optimal probability p to use in p-persistence is 1/N where N is the number of stations trying to transmit in this same slot
- Therefore, the best performance is obtained when p is adapted to this optimal value
- The difficulty is to know how many stations want to transmit (N) so that to compute p in every slot
- This information is available in the simulation, but it is not known in the real operation of the protocol, hence the name of ideal and optimal as it is not achievable
- Note: When p is adapted by the number of stations transmitting in the slot the simulation must make the decision at every slot and the possibility to compute how many slots to wait according to p does not work

Slotted aloha Implementation

Aloha : User State Diagram Implemented

- The user operation is implemented with a state diagram of 3 states:
  - Idle (I): initial state indicating that there is no packet with initiated transmission, so the user is doing nothing (Idle)
  - Transmit (T): A packet has been Transmitted and user is waiting for the ack
  - Resolution (R): A collision has occurred and is waiting for the Resolution time before the next Retransmission



saloha Code Structure: General

init_stats   init_traf

gen_new_slot

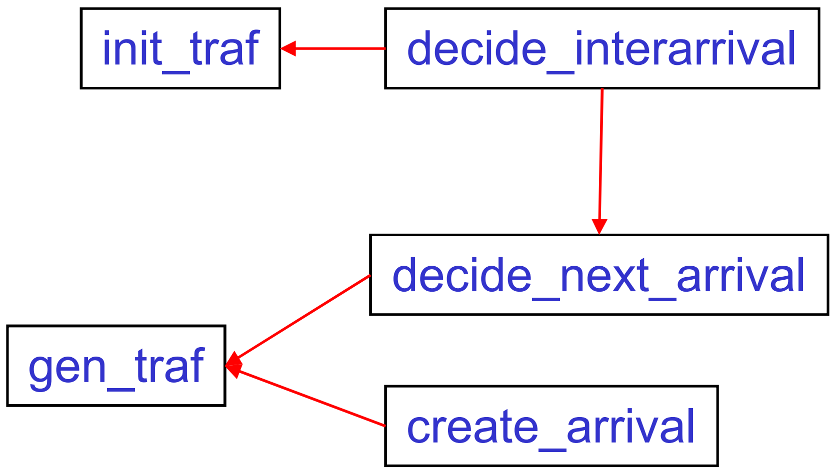
gen_traf

station

run_sink

collect_stats

saloha Code Structure: Packet Generation



saloha Code Structure: Station