Oltre le collisioni

- In tutti i sistemi che abbiamo visto finora, c'è sempre la possibilità che ci siano collisioni
- Le collisioni ovviamente implicano tempo sprecato e quindi inefficienza
- Vediamo se c'è un modo migliore per ***evitare del tutto le collisioni (!)***

Protocolli collision-free

- Anche detti **CSMA/CA** (***collision avoidance***)
- In questa classe di protocolli, si fa un uso intelligente del contention period per evitare del tutto le collisioni
- Nel contention period si decide tutto, e poi la trasmissione del prossimo ciclo di frames continua come stabilito dal contention period, quindi senza collisioni

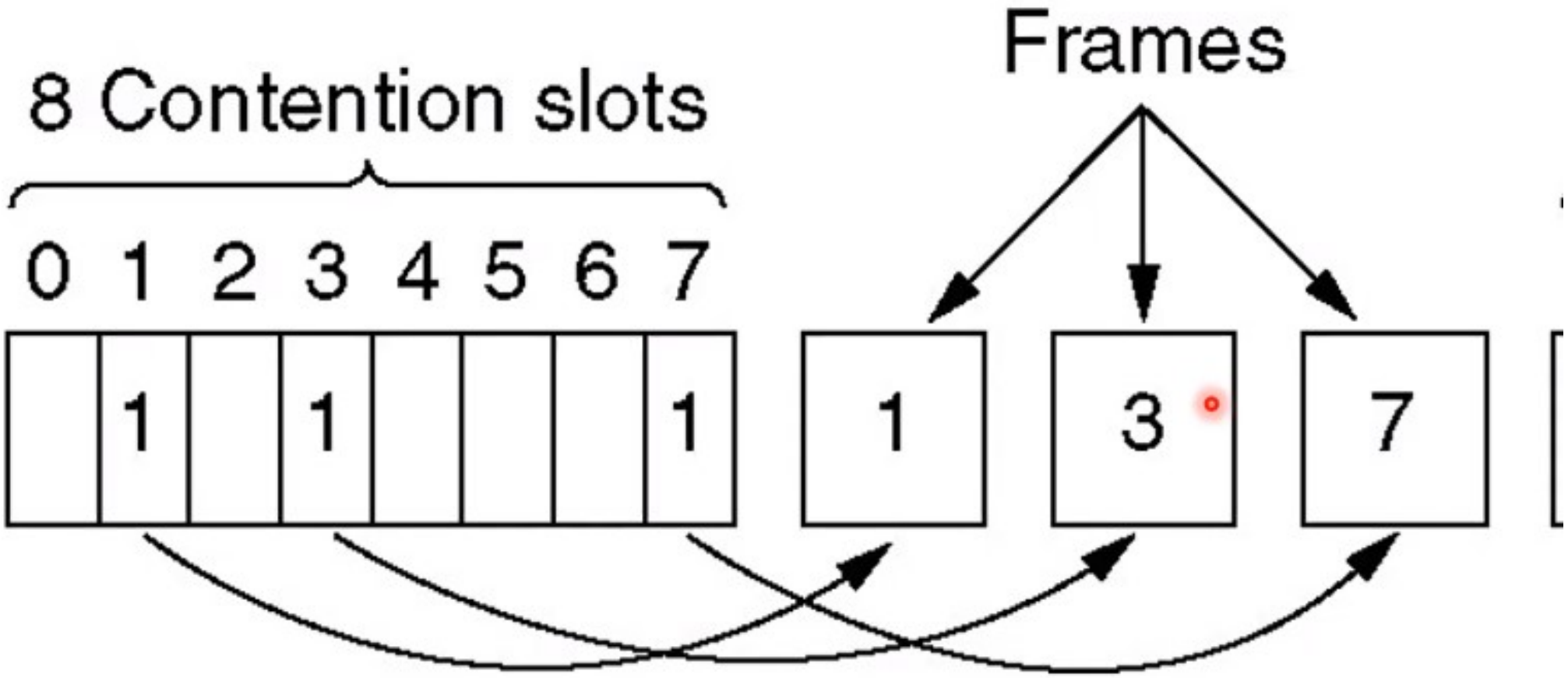
Basic bit-map

- Un modo è abbastanza ovvio: prendere il contention period, e dividerlo in intervalli uguali per ogni stazione (ricordiamo: niente collisioni!)
- Ogni stazione potrà quindi segnalare nel suo "mini-slot" se vuole trasmettere un frame (fa una prenotazione, quindi è anche un cosiddetto ***reservation protocol***)

Basic bit-map

- Quanto grande sara il mini-slot?
- Basta un bit per segnalare la nostra intenzione di trasmettere
- →il contention period conterrà un bit per ogni stazione

Basic bit-map



Efficienza?

- A pieno carico, se la taglia del frame è d, è ovvio che l'efficienza sarà **d/(d+1)**
  (c'è un solo bit "sprecato")

Problemi

- Il problema del bit-map è che abbiamo un bit per ogni stazione, quindi con ***tante stazioni*** il contention period può diventare molto lungo
- → se trasmettono poche stazioni, il tutto è inefficiente

Ricapitoliamo

- Finora, abbiamo visto i protocolli tipo ***CSMA***:
- Funzionano meglio a carico basso della rete, peggio a carico alto (troppi conflitti → troppa latenza)
- Poi abbiamo visto quelli ***collision-free***:

Vediamo un po'...

- Il vero problema di ogni protocollo multiaccesso è ovviamente quello della competizione
- Nel caso CSMA, non abbiamo considerato il numero di stazioni
- Nel caso collision-free, abbiamo invece considerato il numero di stazioni

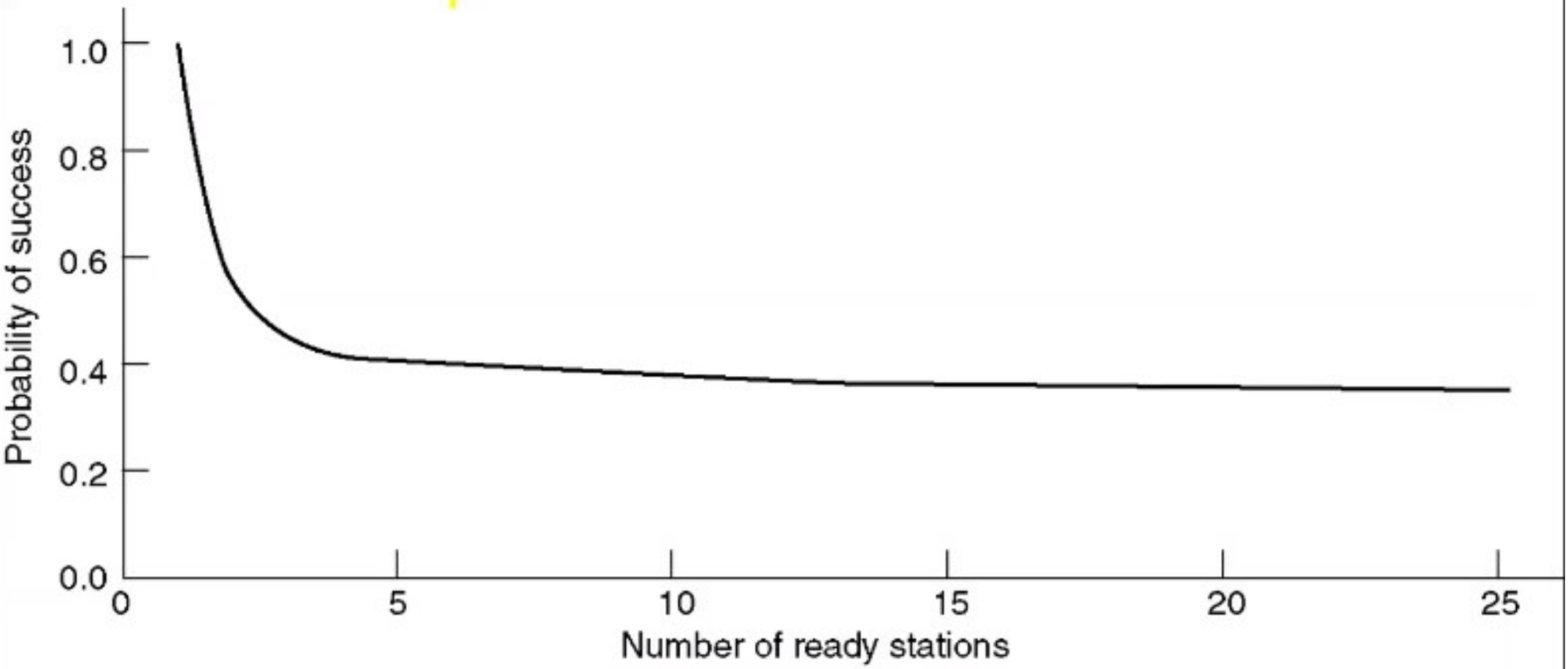
Vediamo…

- Potremmo allora studiare cosa succede a CSMA quando fissiamo il numero di stazioni
- Supponiamo il caso migliore, slotted time, e che ogni stazione trasmetta con probabilità p

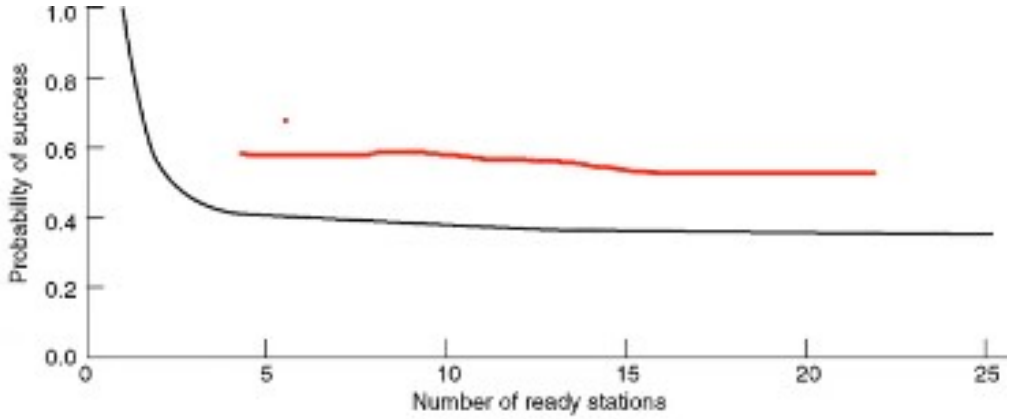
Vediamo…

- Facile: **N * p * (1-p)^(N-1)**
- Avendo un certo numero di stazioni, qual è dunque la scelta di p migliore?
- →differenziamo per p e risolviamo:
- **p = 1/N**
- Quindi date N stazioni, la miglior probabilità che le cose vadano bene è **((N-1)/N)^(N-1)**

Vediamolo graficamente:

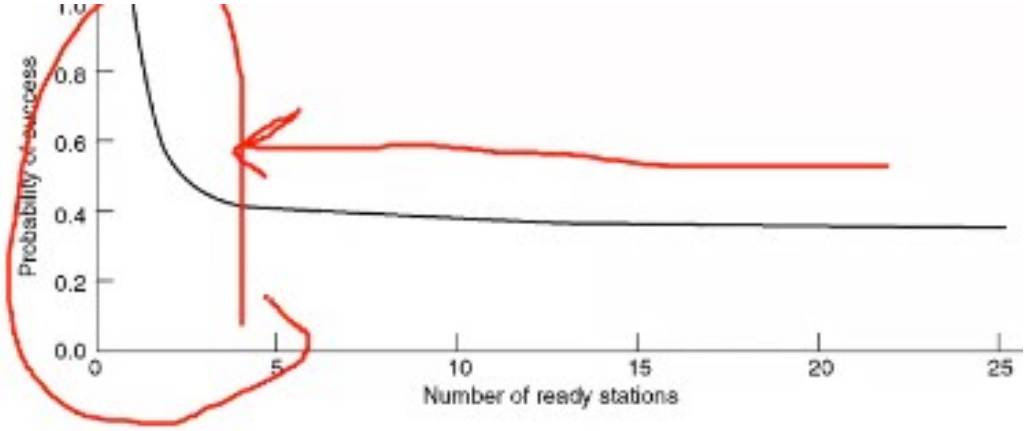


Allora...



- Vediamo una cosa interessante: quando ci sono poche stazioni, la probabilità di successo è buona.
- All'aumentare delle stazioni, la probabilità di successo decresce rapidamente (fino al valore asintotico, che sappiamo essere **1/e**, slotted Aloha!)

Allora...



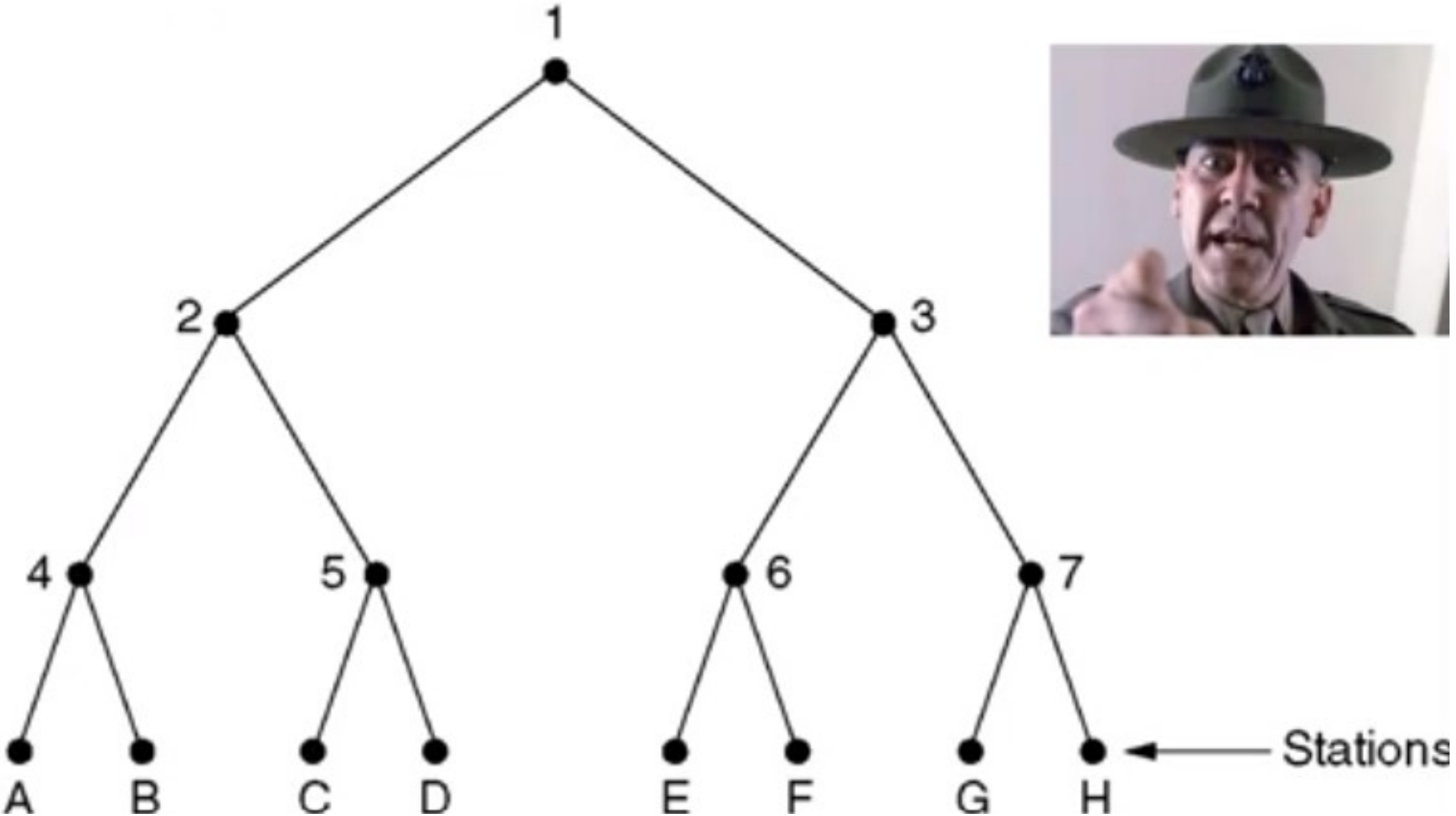
- Vediamo una cosa interessante: quando ci sono poche stazioni, la probabilità di successo è buona.
- All'aumentare delle stazioni, la probabilità di successo decresce rapidamente (fino al valore asintotico, che sappiamo essere **1/e**, slotted Aloha!)

Hmmm…

- Se la competizione non fosse determinata a priori, ma fosse invece ***dinamica***…?
- Cioè, se avessimo un protocollo che se ci sono tante/troppe stazioni che vogliono trasmettere, ***diminuisca dinamicamente la competizione***?

Limited-Contention protocols

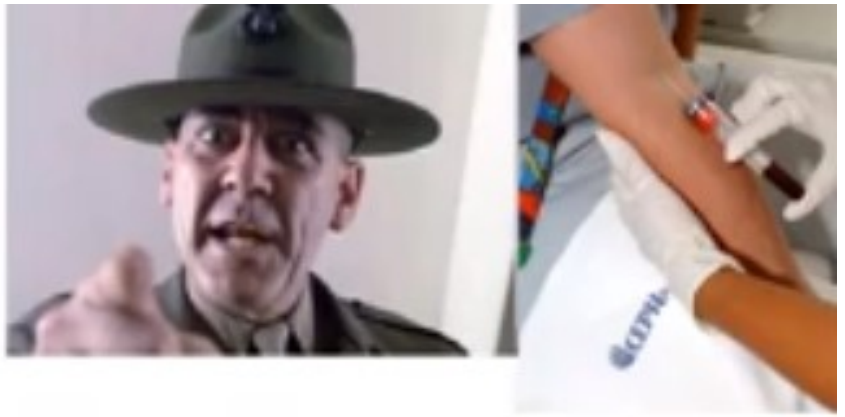
L'idea quindi:
Adaptive Tree Walk Protocol



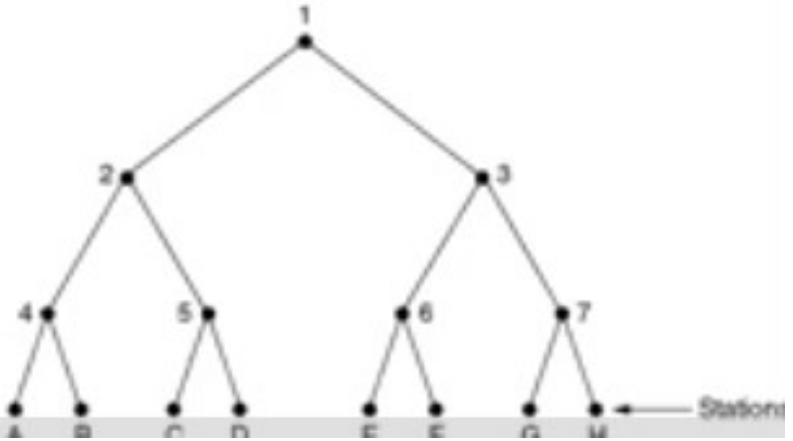
Ancora meglio…

- Possiamo fare ancora meglio: analizzando il traffico recente, possiamo ad esempio renderci conto di quante sono le stazioni che cercano di trasmettere in quel lasso di tempo…

Se…

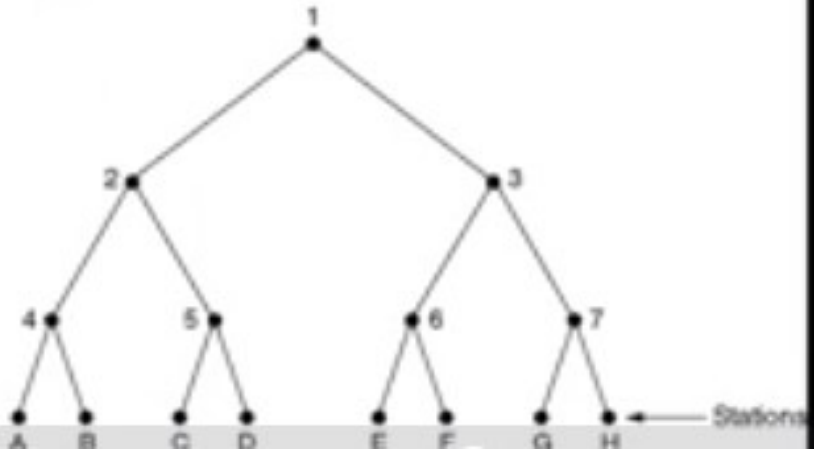


- Sappiamo quante sono circa le stazioni attive, inutile sprecare tempo a cercare proprio dalla cima dell'albero
- Potremmo cominciare direttamente da un sottopezzo dell'albero



Dove?

- Se siamo in un nodo a profondità **P**, i nodi sotto di lui diminuiscono all'incirca come **2^P**
- Se le stazioni attive (**A**) sono distribuite equamente, allora ci saranno circa **A/2^P** stazioni attive nel sottoalbero



A/2^P

- Vogliamo avere il sottoalbero più piccolo che abbia una stazione attiva
- → A/2^P = 1
- →**P = log2(A)**
- Quindi ad esempio, se abbiamo circa 8 stazioni attive, conviene cominciare direttamente a profondità 3.

Passiamo ora...

- Al caso wireless
- Nel wireless, ci sono ulteriori difficoltà, dovute ad un fatto fondamentale: