Lecture 10 Collision resolution

Collision resolution

- Many applications involve an inquiry over a shared channel, which can be invoked for: discovery of neighboring devices in ad hoc networks, counting the number of RFID tags that have a certain property, estimating the mean value contained in a group of sensors etc. Such an inquiry solicits replies from possibly large number of terminals *n*. This necessitates the usage of algorithms for resolving batch collisions (conflicts) with unknown conflict multiplicity *n*.

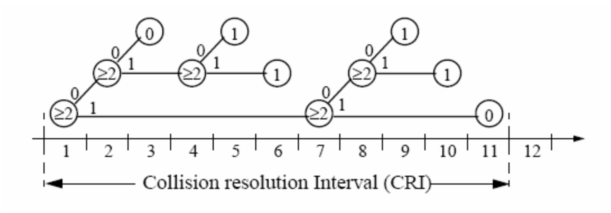
Collision resolution

- *Collision Resolution Protocols* (CRP) aim at resolving collisions as soon as they occur.
- In most versions of these protocols, new packets that arrive to the system are inhibited from being transmitted while the resolution of collisions is in progress.
- This ensures that if the rate of arrival of new packets to the system is smaller than the rate at which collisions can be resolved (the maximal rate of departing packets), then the system is stable.
- The basic idea behind these protocols is to exploit in a more sophisticated manner the feedback information that is available to the users in order to control the retransmission process, so that collisions are resolved more efficiently and without chaotic events.

Binary tree CRP

- Basic Binary tree CRP
  - When a collision occurs, in slot *k* say, all users that are not involved in the collision wait until the collision is resolved.
  - The users involved in the collision split randomly into two subsets, by (for instance) each flipping a coin.
  - The users in the first subset, those that flipped 0, retransmit in slot $k+1$ while those that flipped 1 wait until all those that flipped 0 transmit successfully their packets.
  - If slot $k+1$ is either idle or contains a successful transmission, the users of the second subset (those that flipped 1) retransmit in slot $k+2$.
  - If slot $k+1$ contains another collision, then the procedure is repeated, i.e., the users whose packets collided in slot $k+1$ (the "colliding users") flip a coin again and operate according to the outcome of the coin flipping, and so on.

Binary tree protocol



- Let $\tilde{B}_n$ denote the length of the CRI when n packets collided and let $B_n = E\{\tilde{B}_n\}$
- The fraction n/B_n is the effective service rate for the of packets.
- It can be shown that under Poisson arrival of rates as long as $\lambda < n/B_n$ the system is stable

29.2.2008 5

Binary tree protocol

- B_n is minimized by utilizing balanced "coin" that is the 0 and 1 should be selected with equal probability.
- It can be shown that $B_n \le \alpha n + 1, \quad \alpha \approx 2.886$
- Hence, the maximum throughput of the protocol is

$$\lambda \le \frac{n}{B_n} \approx 0.346 < e^{-1}$$

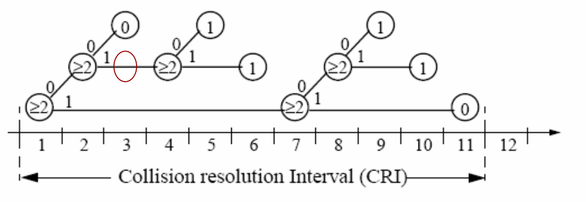
and the maximum delay is

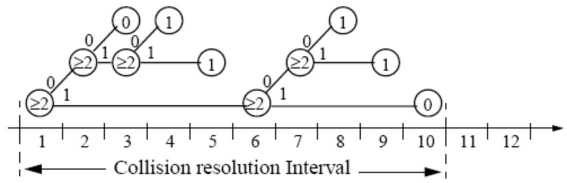
$$D \le \frac{\alpha^2 \lambda + 1 - \alpha\lambda}{(1-\alpha\lambda)^2} + 1$$

29.2.2008 6

Modified Binary tree protocol

- The binary tree algorithm can be improved by noting that empty slot is always followed by collision.
- This could be avoided by automatically splitting the users after empty slots.





Modified Binary tree protocol

- In case of modified Binary tree protocol, we have shorter B_n and thus larger arrival rate can be supported.

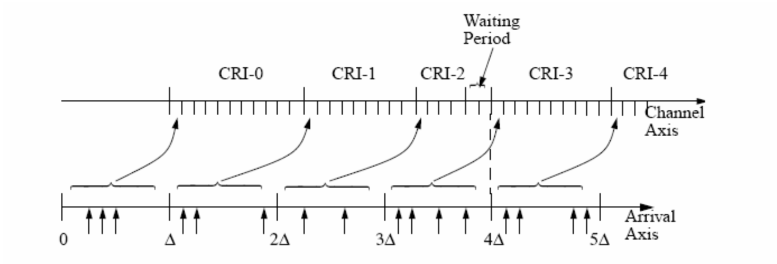
$$\alpha \approx 2.664$$

$$\lambda \leq \frac{n}{B_n} \approx 0.375 > e^{-1}$$

This actually exceed the best that could be achieved with ALOHA.

Epoch mechanism

- The collision resolution interval in binary tree protocol is long if the number of colliding packets is long.
- After one CRI, the packets that arrived during that period will collide. Since the period was long, also the number of colliding packets is expected to be large.
- To reduce the number of colliding packets, the arrivals during the CRI could be divided into epocs. The epocs would then be served in consecutive manner.



Interval-Splitting epoch mechanism

- First all users that arrived during (a,g) will transmit.
- If there is collision, instead of flipping a coin, the interval is split and users that arrived on (a,d) will be served. If there is further collision, we split the interval into (a,b) and (b,d) etc.