1 Introduction

The lecture can be divided into two parts. The first part of the lecture is based on: *A perspective on Multi-Access Channels* by Robert G. Gallager. The paper is


The second part of the lecture is based on: *An Asymptotically Nonadaptive Algorithm for Conflict Resolution in Multiple-Access Channels* by Janos Komlos and Albert G. Greenberg. The paper is


First Part of the lecture is the continuation of the previous lecture (Thursday 18th Jan). In the previous lecture, we discussed the Multi-access channel using Information Theory Approach. We also had a small introduction of the Collision Resolution Approach. The first part of the lecture is based on Collision Resolution Approach.

2 Collision Resolution

2.1 Assumptions

For Collision Resolution approaches, we make some assumption which we have discussed a bit in the last lecture. Here we formally state the assumptions.

2.1.1 Slotted System

Our assumption is that each message transmission is equal to the length of slot and because of this there is a requirement for synchronized clocks. This further infers that short and long messages are not considered and hence scheduling of these messages are not considered in the model.
2.1.2 Collision / Perfect Reception

The approach ignores noise and communication problem. It assumes if more than one transmitter tries to send packet \( k \geq 2 \) collision occurs, otherwise there is no collision and hence perfect reception.

2.1.3 Infinite set of transmitters

The approach does not model infinite set of transmitters and hence it is inaccurate as queuing is not considered.

2.1.4 Poisson Arrival

The assumption is that a Poisson arrival model is used, with a constant rate \( \lambda \).

2.1.5 \((0, 1, c)\) Immediate Feedback

We assume that by end of each slot every transmitters receives a feedback whether 0 or 1 packet was transmitted, or a collision has occurred in that slot. Since each transmitter receives a feedback at the end of slot, the algorithm is adaptive. Algorithms designed with this assumption can be modified to handle delayed feedback.

2.2 Collision Resolution Strategies

2.2.1 Slotted Aloha

In slotted aloha, we make assumption that we have discussed earlier. Here, whenever a packet arrives it is transmitted in the next slot. Whenever a collision occurs, each packet involved in collision is backlogged and remains backlogged until it is fully transmitted. The number of backlogged packets at each interval time \( t \) is the state of the model.

\[ \text{New state} = \text{previous state} + \text{number of packets backlogged} - \text{number of packets sent} \]
If at state $k$ at time interval $t$ and at state $k + i$ at time $t + 1$, then the number $i$ is the new packet arrival in that time interval minus the number of successful transmissions in that time interval. It follows that $i = -1$, if no new packet arrived and one backlogged packed was successfully sent, $i = 0$ when either no packet is received and none sent or one packet received and one sent. The model assumes that packets that just arrived in the last slot (that have not collided yet) are always transmitted. Only those that have collided already (backlogged) follow the probability $p$ transmission rate.

Given Poisson Arrival rate $\lambda$,

$$P_r(j \text{ arrivals}) = (e^{-\lambda \lambda})/j!$$

So, for 0 and 1 arrivals, we have

- $P_r(0 \text{ arrival}) = e^{-\lambda}$
- $P_r(1 \text{ arrival}) = \lambda e^{-\lambda}$

We will define the Markov Chain Model, where $P_{k,k+i}$ is the probability of going from state $k$ to $k + i$. This is one step case as the model is memory-less, so as you get to $k+i$ state, you forget about previous states.

For $i = -1$, no packet arrived but one get through
Figure 3: Probability of going from state $k$ to $k+i$

- $P_{k,k+i} = kp(1-p)^{k-1}e^{-\lambda}$

where,

$k =$ number of packets at time $t$

$p =$ probability of slotted aloha.

For $i = 0$, two cases exist

- Nothing arrives and either 0 or 2 or more backlogged packet is transmitted. So, no successful transmission.

- 1 packet arrives and transmits without collision since no backlogged packets transmit.

$P_{k,k+i} = [1-kp(1-p)^{k-1}]e^{-\lambda} + (1-p)^k\lambda e^{-\lambda}$

For $i = 1$, one packet arrives and at least one backlogged packet transmits.

$P_{k,k+i} = [1-(1-p)^k]\lambda e^{-\lambda}$

For $i \geq 2$, we don’t care what the backlogged packets do, just the fact that more than one arrives means that they will transmit and collision will occur.

$P_{k,k+i} = (\lambda^i e^{-\lambda})/i!$

In understanding how Markov Chain Model behave, we define Drift $D_k$ which is the expected value of $i$ conditional on $k$. $D_k$ is the expected value of $i$ given the value of number of backlog packets with a given constant $\lambda$. If $D_k > 0$ for sufficiently large $k$, then the system become more backlogged. It infers that if $\lambda$ decreases, the drift tends to increase.

The tradeoff in probability $p$ is very undesirable. If $p$ is large, there are more numbers of collisions and if $p$ is sufficiently small, there will be more delay. So the best option is keep changing $p$ of backlogged packets.

2.2.2 Splitting Algorithm

The concept of splitting algorithm is to split the set of packets involved in a collision into a transmitting set and a non transmitting set, while making others packet wait. The 2 papers “An Asymptotically Nonadaptive Algorithm for Conflict Resolution in Multiple-Access Channels” by Komlos and Greenberg and “A lower bound on the time needed in the worst case to resolve conflicts deterministically in multiple access channels” by Albert G. Greenberg and Schmuel Winograd are based on splitting algorithm.

A class of splitting algorithms is tree algorithm, with following properties:
• Backlogged packets gets priority
• Postpone newly arrived packets and service the packets already been involved in collision

Given a particular set of packets to be sent, we divide into 2 sets: transmitting sets and non-transmitting sets.

Splitting Algorithms could differ in following ways:

• Rules used for splitting could differ
• Rules for allowing waiting packets to be transmitted could differ

Each of splitting algorithm has 2 modes:

• normal mode
• collision mode

If a collision occurs in normal mode, all the transmitters goes to the collision mode and all new arrivals wait for end of collision mode. Packets involved in collision selects one of the two subsets. In the slot, following the collision the first of these subsets is transmitted. If further collision occurs, the subsets is further divided into subsets and iteratively transmits the packets. The channels decides whether it wants transmission or not.
3 Nonadaptive Algorithm for Conflict Resolution

An Asymptotically Nonadaptive Algorithm for Conflict Resolution in Multiple-Access Channels by Janos Komlos and Albert G. Greenberg.

The paper discusses on a splitting algorithm. The algorithm is non-adaptive and tells that the transmitter who is going to transmit. The algorithm discussed has following properties

- It has the same time complexity for resolving collisions as the best tree algorithm can have.
- The algorithm is non-adaptive and hence does not require any feedback.

The algorithm should have a priori of number of stations that actually want to transmit packets, which is some what a cons of the algorithm.

Problem - There are n stations tapped into the channel, \( N = \{1, 2, \ldots, n\} \). The slot of transmission are numbered 1,2,3,\ldots with some \( k \) devices decide to transmit.

The set \( I \) a subset of \( \{1, 2, 3, \ldots, n\} \), is the set that actually wants to transmit. The algorithm tells these devices to transmit the packets. All stations receive feedback \((0, 1, c)\). At each slot \( i \), the algorithm produces \( Q_i \subseteq 1, 2, 3, \ldots, n \) where \( Q_i \) is the set transmitter that are allowed to send at slot \( i \).

The best tree algorithms has complexity of \( O(k + k\log n/k) \). This paper proposes an algorithm that has the same complexity without feedback. Since the algorithm is non-adaptive, it can be defined by a sequence whose elements are all decided initially. The sequence of Query Sets are \( Q_1, Q_2, \ldots, Q_i \).

Given \( Q_i \) and \( I \), we can define the sequence of \( I_j \)'s(Transmission Set), where all \( I_j \) is the set of conflicting stations that are still transmitting in slot \( j + 1 \). Each \( I_j \subseteq I_0 \subseteq I \) and is defined as follows


- $I_{j+1} = I_j - Q_j$, if $|Q_j \cap I_j| = 1$ (Transmission Takes Place)
- $= I_j$, otherwise

If $|Q_j \cap I_{j-1}| = \{x\}$, we say that slot $j$ isolates $x$. Ideally, we need every slot to isolate 1 transmitter.