

Problem 1. Embedding trees

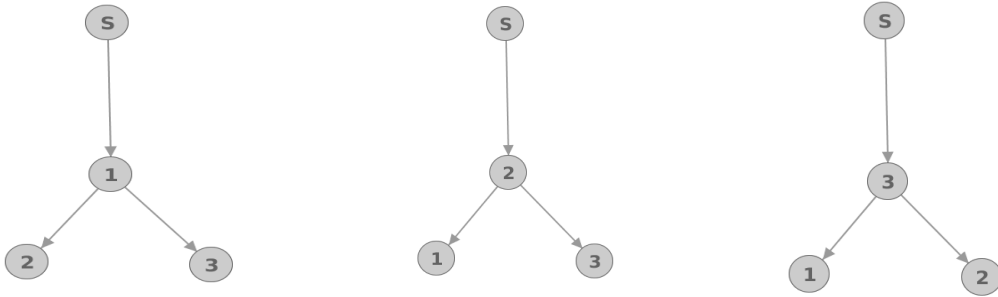
Consider a network of one server and $N = 3$ peers, given that (1) $u_s = 2$, $u_1 = 3$, $u_2 = 2$ and $u_3 = 1$ (all in Mbps), (2) all nodes have unlimited download capacity, and (3) each peer in a multi-cast tree can upload to any number of peers.

(a) Find the maximum multi-cast rate r_{max} . Draw a multi-tree that achieves this maximum.

(b) Find the rate r_i of the i th multi-cast tree, for all $i = 1, \dots, T$, where T is the number of multi-cast trees from part (a).

(c) Now consider a network with one server and $N = 3$ peers, with $u_s = 3$, $u_1 = 3$, $u_2 = 2$, and $u_3 = 1$ (in Mbps). But now we impose the constraint that each peer in a multi-cast tree can only upload to at most one peer, so as to limit the overhead in maintaining the states of the peers. Draw the resulting multi-tree that achieves the maximum multi-cast rate, and compute the per-tree rates r_i .

Solution: (a) First we construct a maximum multi-cast tree according to the theoretical design of P2P network



according to mathematical analysis, without the consideration of user download speed, the maximum multi-cast rate $r_{max} = \frac{1}{T} = \min\{u_s, \frac{u_s + \sum_i u_i}{N}\} = \min\{2, 8/3\} = 2$.

(b) From previous calculation we conclude that $u_s \leq (u_s + \sum_{i=1}^N u_i)/N$, thus in this case the rate of tree i carry a rate $r_i = (\frac{u_i}{\sum_{j=1}^N u_j})u_s$

$$\begin{aligned} u_1 &= \frac{3}{6} \times 2 = 1 \\ u_2 &= \frac{2}{6} \times 2 = \frac{2}{3} \\ u_3 &= \frac{1}{6} \times 2 = \frac{1}{3} \end{aligned}$$

(c). In this case the we are solving a single session problem that

$$\begin{aligned} \max \quad & r = \sum_{t \in T} y_t \\ \text{s.t.} \quad & \sum_{t \in T} t y_t \leq C(v), \forall v \in V \end{aligned} \tag{1}$$

where y_t denotes the rate in subtree t , and C_v denotes the maximum uplink capacity of node v . Notice that the problem is nonconvex, thus we need to reformulate the original problem into its dual problem

$$\begin{aligned} \min \quad & \sum_{v \in V} C_v p_v \\ \text{s.t.} \quad & t p_v \geq 1, \forall t \in T \end{aligned} \quad (2)$$

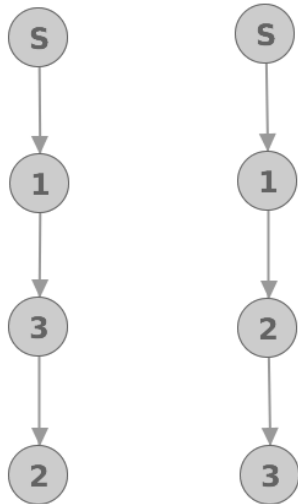
where p_v denotes the dual variable of node v .

A proposed algorithm to solve this problem from is as following

```

Primal-Dual Algorithm: Single-Session
 $p(v) \leftarrow \frac{\delta}{C(v)}, \text{flow}(v) \leftarrow 0, \forall v \in V, Y \leftarrow 0,$ 
 $D \leftarrow 0$ 
while  $D < 1$ 
    Pick tree  $t \in T$  with the smallest  $Q(t, p)$ 
     $y \leftarrow \min_{v \in I(t)} C(v)/m_{v,t}$ 
     $\bar{t} \leftarrow \arg \min_{v \in I(t)} C(v)/m_{v,t}$ 
     $\text{flow}(v) \leftarrow \text{flow}(v) + y m_{v,\bar{t}}, \forall v \in I(\bar{t})$ 
     $Y \leftarrow Y + y$ 
     $p(v) \leftarrow p(v)(1 + e^{\frac{m_{v,\bar{t}} y}{C(v)}})$ 
     $D \leftarrow \sum_{v \in V} C(v) p(v)$ 
end while
Compute scaling factor  $\alpha \leftarrow \max_{v \in V} \frac{\text{flow}(v)}{C(v)}$ ;
Output capacity  $r^* \leftarrow Y/\alpha$ ;
    
```

By solving the dual problem according to the algorithm, we can obtain the maximum rate tree as following



where the rate of tree 1 is 1, and the rate of tree 2 is 2. □

Problem 2. Aloha

There is a simpler random access protocol than CSMA that is just as famous. It is called **Aloha**, as it was invented in Hawaii in the early 1970s, and further led to the development of packet radio technologies. The operation of (the slotted time version of) Aloha is easy to describe. During each timeslot, each of a given set of N users chooses to transmit a packet

with probability p . We assume that if two or more users transmit at the same timeslot, all packets are lost. This is the only feedback available at each transmitter. Each lost packet is retransmitted with probability p too. We assume this process continues until a packet is eventually transmitted successfully.

(a) Assume the channel supports 1 unit of capacity (say, 1 Mbps) when there is a successful transmission. What is the throughput S as a function of N and p ?

(b) What is the optimal p , as a function of N , to maximize the throughput?

(c) As the network becomes large and $N \rightarrow \infty$, what is the maximized throughput? You will see it is not a big number, which is intuitive since slotted Aloha described above has neither the listen and wait nor the exponential backoff features. Aloha takes the least amount of communication and coordination and it does not even use carrier sensing. It also has a low throughput. CSMA leverages implicit message passing throughput can be high for a very small number of users but drops as the crowd gets larger. A centralized scheduler would have incurred even more coordination overhead, and would in turn provide the best performance. But in many networks, it is infeasible to afford a centralized scheduler.

Solution: (a) The probability of a successful transmission is calculated by

$$S = Np(1 - p)^{N-1}$$

(b) Assume N is a constant and we obtain the first derivative of throughput S that

$$\frac{dS}{dp} = N((1 - p)^{N-1} - (N - 1)p(1 - p)^{N-2})$$

Setting the derivative equals to zero, we can obtain that

$$p = \frac{1}{N}$$

and in this case

$$S = (1 - \frac{1}{N})^{N-1}$$

(c) From previous calculation, we can obtain that the maximized throughput given N is $S = (1 - \frac{1}{N})^{N-1}$. By calculating the value of maximized throughput limiting to infinity,

$$\lim_{N \rightarrow \infty} S = \frac{1}{e}$$

we can draw that when the network gets larger, the maximized throughput will converge to $\frac{1}{e}$ □