Surname : Pendyala
Firstname : Praveen Kumar
Matrikel # : 2919474

Some of the answers were arrived at while working in group with:
Ankush Chikhale : 2973449

**Problem 8.1**

a. Single tree vs Multi tree w.r.t. peer churn

**Single tree:**

Pros:
- Minimal overhead for transmission coordination (push-based)

Cons:
- Peer churn can disconnect complete sub-tree

**Multi tree:**

Pros:
- More resilent to peer churn compared to Single tree topolgy
- Minimal overhead for transmission coordination (push-based)

Cons:
- Overhead of maintenance
- Balancing of trees

b. Single tree vs Multi tree w.r.t. Fairness

**Single tree:**

Pros:
- Minimal overhead for transmission coordination (push-based)

Cons:
- Only few peers (inner nodes only) can contribute to upload bandwidth
- Peers at the edge (leaf nodes) only consume

**Multi tree:**

Pros:

- All peers can contribute
- Peer contribution can be adapted to Peer resources

Cons:

- Overhead of maintenance
- Balancing of trees

c. Tree vs Mesh w.r.t. delay upper bound

**Tree:**

Pros:

- Minimal overhead for transmission coordination (push-based)
- Lower upper bound on delay compared to Mesh topolgy

Cons:

- Overhead of maintenance of topology

**Mesh:**

Pros:

- Very robust against peer churn

Cons:

- Increased playback lag due to block maps exchange

**Problem 8.2**

a. Buffer is used to account for bandwidth and jitter fluctuations in the network

b. Playback stalls will increase

c. Alternatives to overlay multicast:
    - IP unicast
        i. IP unicast doesn't have maintenance overhead as compared to overlay multicast
        ii. So, for a small number client IP unicast may outperform overlay multicast
    - IP broadcast
        i. When the content is consumed by most of the nodes in the network, IP broadcast is better.
        ii. Since broadcast doesn't have maintenance overhead
    - IP multicast
        i. When all the entities in the network support IP multicast, this is the optimal solution
        ii. No maintenance overhead

d. Single-tree vs Multi-tree for very heterogenous clients.
    - **Multi-tree:** Multi-tree streaming has inherent support for very heterogenous clients. Clients with lower resources may take fewer inner node positions
    - **Single-tree:** If node resources can be identified at join time, Single tree can also serve as good alternative for very heterogenous client scenarios by placing them close to leaf positions

e. The source would have higher load than a normal server-client streaming scenario and thus, it defeats the whole purpose of building a complex system with additional maintenance overhead

**Problem 8.3**

Maximum SVC quality by P1 - (SD, Q1, 30Hz)

Layer elements requested:

- (CIF, Q0, 15) - P3 - 1 kbps
- (SD, Q0, 15) - P5 - 2 kbps
- (CIF, Q1, 15) - P3 - 2 kbps
- (SD, Q1, 15) - P5 - 4 kbps
- (CIF, Q0, 30) - P4 - 2 kbps
- (SD, Q0, 30) - P6 - 4 kbps
- (CIF, Q1, 30) - P4 - 4 kbps
- (SD, Q1, 30) - P6 - 8 kbps

Total bandwidth needed : 27 kbps

**P1 bandwidth dropped to 25 kbps**

Maximum SVC quality by P1 - (SD, Q1, 15Hz)

Layer elements requested:

- (CIF, Q0, 15) - P3 - 1 kbps
- (SD, Q0, 15) - P5 - 2 kbps
- (CIF, Q1, 15) - P3 - 2 kbps
- (SD, Q1, 15) - P5 - 4 kbps

Total bandwidth needed : 9 kbps