Project Matryoshka: NDN Multiplayer Online Game
Zhehao Wang, Zening Qu, Jeff Burke
123 UCLA REMAP

Introduction
Matryoshka is a peer-to-peer multiplayer online game (MOG) running on NDN. In this project, we identify the MOG synchronization problem. Then we propose an octree partition of the game world, and a two-step synchronization design.

Background
Peer-to-peer structures were explored for commercial online games. However, maintaining security and availability while scaling users has driven most multiplayer online games towards a client-server or client-superpeer architecture. Client-server multiplayer games face certain problems:
1. a small number of points of failure;
2. traffic centralizing at several servers.
To tackle these problems, we present Matryoshka, whose design is based on Sync: by exchanging data digest each party learns about the missing data, and then can retrieve data via built-in multicast delivery.

Gameplay
- Play as a matryoshka
- Explore unknown universe
- Discover other players
- Get player position updates
- Interact with environment

Problem Analysis
The challenge of the MOG we are addressing: peer-to-peer synchronization in a distributed virtual environment.

Synchronization
Players whose areas of interest intersect with each other should reach consistent conclusions about things in the intersected area.

Locality
A player only needs to know the updates of objects within its Area of Interest (AoI) in the virtual game world.

Figure 2 illustrates the problem. Each physical peer hosts a player, which has an AoI. Take peer 2 as example.
- It hosts player3.
- It should discover player1, player2, NPC2 and NPC3.
- Its knowledge of player1 and 2’s locations should be updated when they move.
- The AoI should move as player3 moves.

Virtual World Partitioning
Figure 3 illustrates the octree partition of the virtual environment. The whole world is represented by a top-level cube.
- Octree partitions the virtual environment into octants statically and recursively.
- With octree, we hope to provide a shared namespace for every peer maintaining the game.
- All the peers that care about the same region can share the data brought by synchronization interests towards the same nodes.

Design Overview
Two-step design:
- Discovery: which players are in a peer’s vicinity.
  - Peers with overlapping AoIs synchronize “discovery namespaces” for octants of mutual interest to find other objects in the game world:
    - They periodically express Interests in the discovery namespace containing the octant indices they are interested in to all peers, along with a digest of the object names they know in each octant.
    - Each peer responds with their knowledge of objects in the octants.
  - The namespace for discovery is given in Figure 4.
  - Game name component separates the game into several sub-worlds.
  - Octant indices indicate the octants absolute location in the game world.
  - Digest component contains the hash of the set of object name strings in that octant.
  - Every peer should have the same hash for octants belonging to their intersection when steady state is reached.

Figure 4: Discovery namespace

Update: what are those players doing.
- Peers express position update interests using the object names returned in step one.
- Peers respond with current location, with version appended. Digest in step one is modified based on this response.

Figure 5 illustrates the update namespace.
- Process name represents a game instance, which hosts the player avatar.
- Position and action components fetch the latest data.
Sample pattern of communication for a game instance (a) to discovery another instance (d) is given in Figure 6.

Future Work
Future work includes further evaluation of the design, and addressing problems caused by hierarchical octree partitioning.
- Difficult to represent AoIs that are close to the border between two sub-regions of the highest subdivision hierarchy.
- Difficult to represent spherical AoI.
- Need routing to keep up with the changes in AoI represented by octree.

References

Contact Information
- Email: wangzhehao410305@gmail.com