

## EE 5516: Introduction to Networking (Fall, 2018)

---

### Class locations and times

Teacher	Zhen Zhao, Ph.D.
Lecture times	Wednesday 5:30-8:00pm
Lecture room	Wachman 313

### Project – Design a scalable live video architecture

In this project, you will design a network-based system supporting live video streaming from home cameras to the mobile apps assume you design everything from scratch. To help you with starting designing, here is something you might want to keep in mind:

1. In your design, your camera should be ready to connect to minimize the waiting time when users try to start streaming from the mobile app. In order to make this possible, your cameras should connect to a server and keep its connection there. For this project, you are expected to describe how the cameras are connected to the server and how the connections are maintained. For example, your camera should establish a TCP connection first starting with a 3-way handshaking. After the TCP connection is set, you would like to maintain this connection as a persistent TCP connection. One way of doing so is to keep a websocket connection over TCP. You also need think about how to make sure the persistent connection keeps active (network is not stable in the real world, you need find a way to figure if the persistent connection is really active as you plan to be). You also want to consider network security in your design. Not any camera could use your service, should only those authenticated ones.
2. Once your camera is connected to the server, whenever the user wants to start streaming, the app could establish its own connection to the server. After that, the app needs to have a way to find which camera it should connect to and the server shall help that particular camera to exchange messages with the app s.t. they could try to build a logic connection between them. You can refer to how an online chatting room works. Again, you need consider how to avoid others to stream videos from your cameras. The network security is critical here to prevent anyone else but the real camera owners to stream from their own cameras.
3. The logic connection between the came and the app could consist of
  - Either a peering connection between the camera and the app. The reason behind this solution is it saves a lot of money for the system operation. If the streaming is directly from the camera to the app, as a service provide, you don't have to pay any streaming fee.
  - Or an indirect connection via a relay server. In some cases, say, your camera is behind a firewall with both ip and port mappings. Usually when the port is mapped on the firewall, the direct peering connection might not be able to be established.

You will have to consider both solutions in your design, maybe the peering is primary and the relay is secondary as a backup. For the peering connection, your app needs to know your camera's public IP address and port. Your camera also needs to know your app's public IP address and port. For the connection over the relay, you will have to think about only eligible uses can use your relay server, the network security is required.

4. Think about if your streaming should be over TCP or UDP, which could be the best for your app performance.
5. Also keep in mind, you're designing a scalable system. If you have millions of cameras, you don't expect them to connect to a single server. You can consider either a DNS server with mapping a url to a public IP randomly selected from a set of public IPs or a cluster with specifying each server's private IP.
6. What if a customer has multiple cameras, how do you set peering connection and/or relay server for streaming?

There is no perfect design. There is no "correct" or "wrong" with your designed system. No matter who designs what, there are always pros and cons. Please describe your network-based system's pros and cons.