Distributed Systems II project

Personal Multimedia Server (PeMuS)

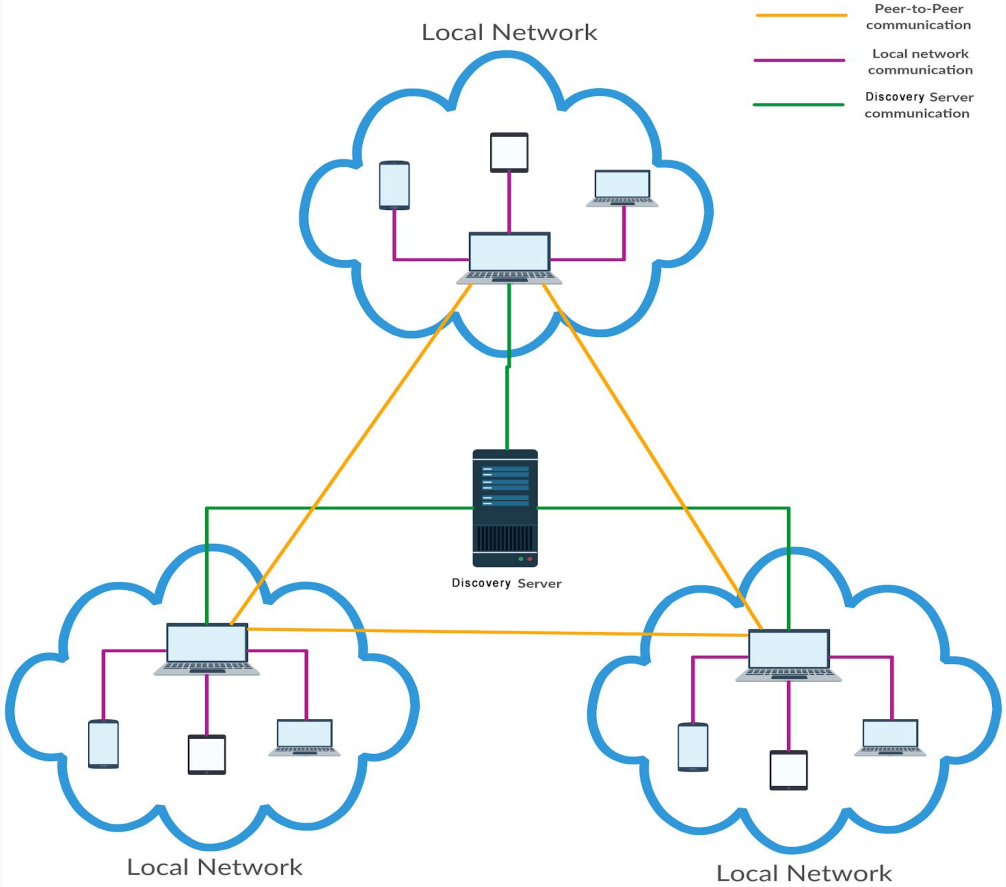
Personal Multimedia Server (PeMuS)

Distributed cross-platform application that gives the possibility to **share** and **stream** your videos and music in your favorite devices: phones, tablets, gaming consoles, and smart TVs whenever and wherever you are.

PeMuS - Main Technologies

- 
  Server environment that use Javascript for develop **BackEnd** applications.
- 
  Framework for create cross-platform desktop GUI applications using Chromium instances for the **FrontEnd.**
- 
  Web application framework to help organize the application into an MVC architecture and expose **API**.
- 
  PeerJS provides **peer-to-peer** API built on top of WebRTC, supporting both data channels and media streams.
- 
  Torrent framework for distribute and share files that uses WebRTC instead of TCP for true peer-to-peer transport.

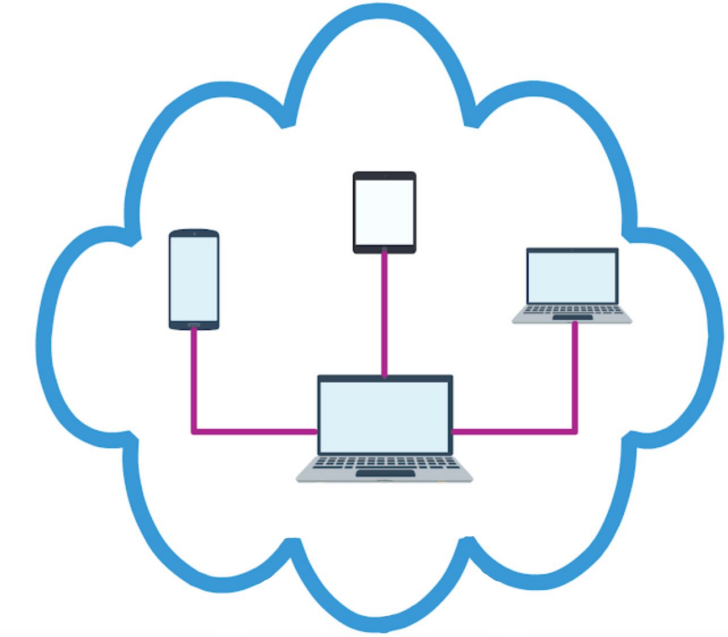
PeMuS - Architecture



PeMuS - Local Communication and Streaming

- Thanks to **Express framework**, for every device in the same network that have a browser is possible to require the streaming of a shared video or song.

- It gives to different member of a network the possibility to enjoy at the same time music and videos without having duplicate files, everything can be stored in just one machine.



PeMuS - Remote Communication and Streaming

- To obtain a **user-friendly** application we want that the users don’t have to configure the NAT and port forwarding on their own router for use the public IP, for this reason using the Express framework would **not** be the best solution for the remote connections.

- Because of that the **Peerjs** framework is used to handle connection request, objects transmission and obtain a real-time notification system. Peerjs is built on top of **WebRTC** that provide web browsers to embed real-time text, audio and video communication.

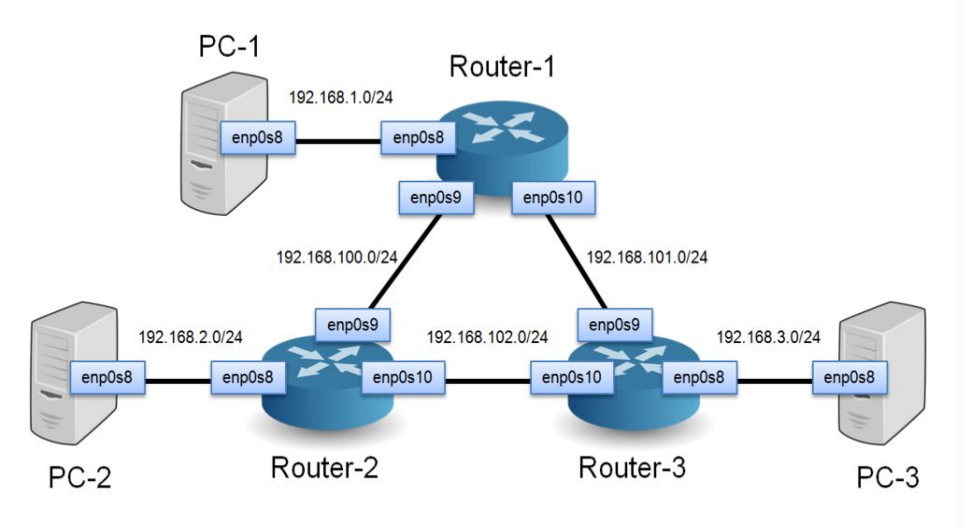
- Because of Peerjs limitation, that allow just real-time streaming, the new **WebTorrent** framework has been introduced. It use WebRTC instead of BitTorrent TCP for peer-to-peer data transmission. With this decentralized structure the application obtain an high level of distribution and scalability.

PeMuS - Network Virtualization for Testing

Using this guide: http://www.brianlinkletter.com/how-to-use-virtualbox-to-emulate-a-network/ I have created a virtual network using virtualbox.

The network is composed of routers and endpoint with the network structure similar to the one in the picture, for test the application and communication.

All the virtual machine are Ubuntu distros.



PeMuS - Network Impact

All the tests were made in the Mid Sweden University network.

- **Initial seeding:** ~4,8 KB/sec send for 12 songs and 2 videos
- **Music streaming from browser**: ~1,24MB/sec send for 1 song
  ~1,95MB/sec send for 3 songs at the same time
- **Video streaming from browser**: ~1,43MB/sec send for 1 video
  ~3,29MB/sec send for 3 videos at the same time
- **Music streaming from torrent**: ~1,31MB/sec send for 1 song
  ~2,10MB/sec send for 3 songs at the same time
- **Video streaming from torrent**: ~2.52 MB/sec send for 1 video
  ~3.48 MB/sec send for 3 videos at the same time

PeMuS - Possible Improvements

- **Remote Streaming for every device:** Use **ReactJS** for create native apps and allow remote streaming in Android, iOS and smart tv or use the **WebRTC** api of Peerjs and WebTorrent for stream directly from the browser.

- **Use mongoose ORM and mongodb:** At the moment all the permanent data are saved in XML files, this is convenient and light-weight at the moment but if the application will become more complex using a database is the best choice for grant the CRUD.

- **Process optimization:** A process optimization is required because sometimes some unused child processes remain open in background overloading the resources.

- **Photo stream:** Introduce in addition of video and music stream also photography stream.

Conclusion

Compared to others competitors like Plex, Kodi, Emby or Stremio, **PeMuS** is the only one that can grant a **remote decentralized communication** without the need of configure router for the users or the need of handle some forwarding server for the company and so save on maintenance costs.

At the moment the application require quite a lot of computational and network resources that has to be optimized to be competitive in the market.

Application Repo: https://github.com/nicolasebastianelli/PeMuS
Discovery Server Repo: https://github.com/nicolasebastianelli/PeMuS-Server