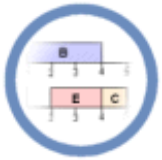




Lecture ”6”
Inter-Device Communication

<lecturer, date>

Outline

- iOS Connectiviy
  - **Networking Overview**
  - Multipeer Connectivity

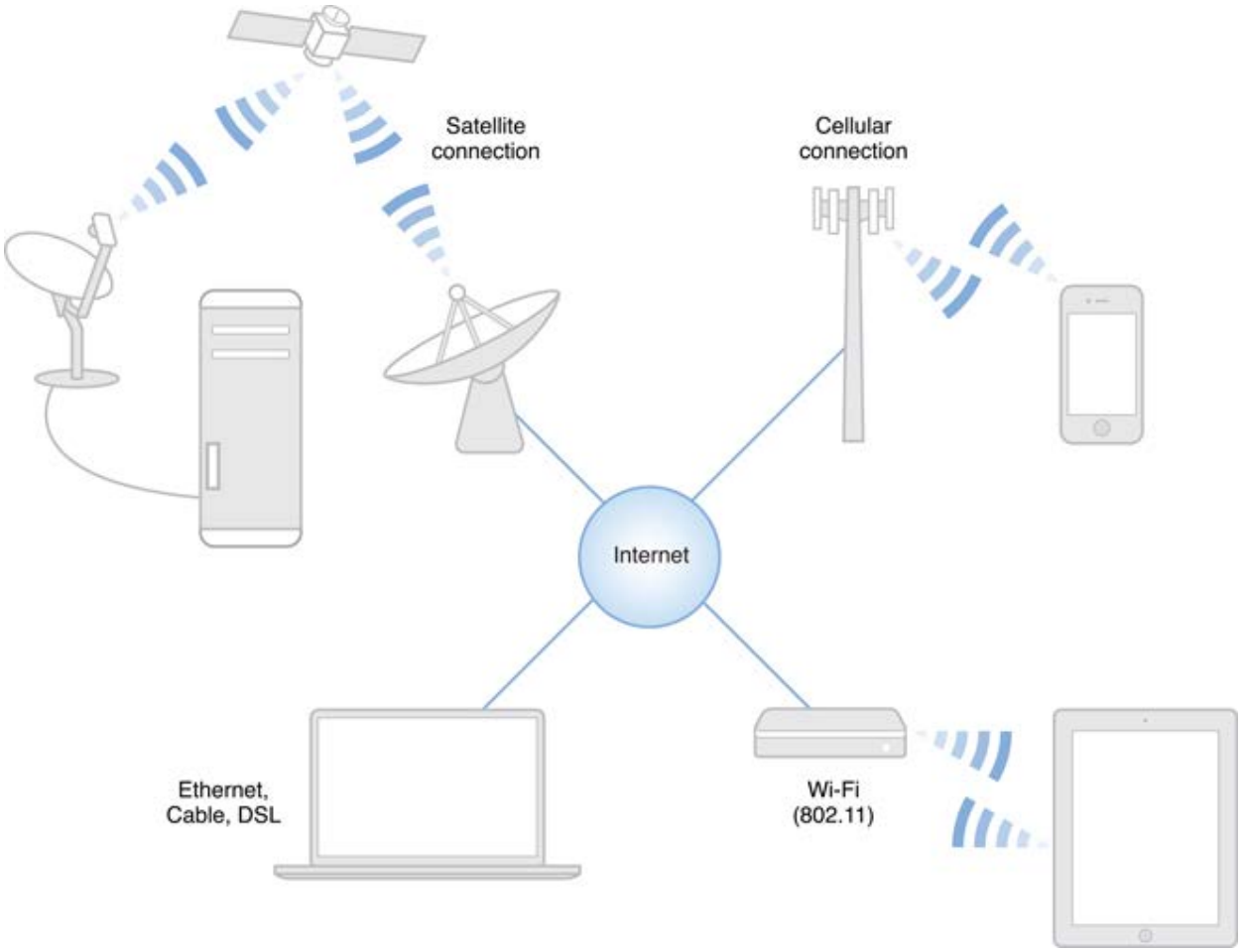
Networking Overview

- Complex world of networking
- Wild range of technologies e.g., cable model, DSL, Wi-Fi, satellite uplinks
- Each technology with distinct characteristics including differences in
  - Bandwidth
  - Latency
  - Packet loss
  - Reliability

Networking Overview

☹ Further complexity in the way from user to an Internet server!

☺ The code must adopt to changing network condition including performance, availability and reliability



Networking Overview

- Transfer only as much as data is required
- Avoid timeouts whenever possible
- Allow user to cancel transactions taking too long through appropriate user interface
- Handle failures gracefully
- Degrade gracefully when network performance is low
- Choose appropriate APIs for the task
- Design to reduce security risks

Networking Tasks

- HTTP/HTTPS requests e.g., POST/GET
- Establishing connections to remote host with/without encryption/authentication
- Litening for incoming connections
- Send/receive data with connectionless protocols
- Publish/browse/resolve network services with Bonjour

☞ Security your responsibility

Outline

- iOS Connectiviy
  - Networking Overview
  - **Multipeer Connectivity**

Terminology

- Nearby : Within range of supported wireless technologies
- Peer : Nearby device
- Advertiser : Device discoverable by other nearby devices
- Browser : Device searching for nearby devices

Multipeer Connectivity

- Establishing communication between nearby devices, exchanging data/other resources without much effort
- Using Wi-Fi, Bluetooth

Multipeer Connectivity

Multipeer Connectivity Framework (MPC)

- On nearby iOS devices only
  - Same network infrastructure e.g., Wi-Fi, peer-to-peer Wi-Fi, Bluetooth
  - No long distance
- A device called peer get connected/communicate to many other at the same time
- Through a **Multipeer Connectivity Session**
  - ☞ Manage the whole communication and data exchnage between peers

Use Cases

- Interactive tutoring
- Collaborative document/photo editing
- File sharing
- Coordination between multiple devices
- Sensor data aggregation

Features

- Multiple wireless technologies
- Interface selection
- Convenience discovery and invitation UI
- Message-based/stream-based data
- Encryption/authentication

Multipeer Connectivity

- Essentials
  - Discovery phase
  - Session phase
- Advanced
  - Programmatic discovery
  - Security
    - Encryption
    - Authentication

Essentials: Phase 1

- Phase 1 : Discovering Peers
  - At least one device as a browser to search for peers
- Browser Tasks
  - Present nearby peers
  - Send invites
  - Handle invite responses
  - Connect peer to session

Essentials: Phase 1

- The second device discoverable : Advertise itself to nearby peers such that can be invited to a session

- Advertiser Tasks
  - Make device discoverable
  - Present invitations to user
  - Handle user response
  - Connect peer to session

Essentials: Phase 1

- Limited comunication/knowledeg with/of other peers
- Access to *discoveryInfo* data from nearby peers
- Any context data from other peers when they invite a session
- User chooses which peers to add to the session, then app invites those peers to join
- Apps running on nearby peers can accept/reject the invitation and ask user for permission

Essentials: Phase 2

- Phase 2 : Session Phase

  ✓ If the peer accepts the invitation

  - The browser establishes a connection with the advertiser

Session phase begins ✓

  - The app can communicate with one or more peers within the session
  - Framework notifies your app through delegate callbacks when peers join/leave session

Data Types Using MPC Framework

- Message-based data e.g., text, image, every thing could be converted to NSData object
- Streaming
- Resources e.g., files

Message Transmission Modes

Reliable

- Makes sure every thing will arrive to receiver
- Application critical data
- Retransmission
- In order delivery
- Takes more time

Unreliable

- Data sent in no-time
- Time sensative data
- No delivery guarantees
- No order guarantees
- Fast

Programmatic Discovery

- More flexibility
- Finding devices/sending invitations handled programmatically
- Bulit a custom UI for discovery
- Programmatic browsing/advertising

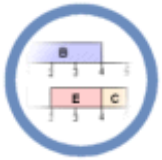
Browsing Functionality

1. A browsing UI built-in directly into the framework : A modal view displays all available/connected devices
2. A totally programmatic way : offers greater flexibility and customized browsing based on the need of the application

References

- iOS Connectivity:
  - https://developer.apple.com/library/ios/documentation/MultipeerConnectivity/Reference/MultipeerConnectivityFramework/
  - http://www.appcoda.com/intro-multipeer-connectivity-framework-ios-programming/
- Apple developer forums:
  https://idmsa.apple.com/IDMSWebAuth/login?appIdKey=4a75046cda87eab6386a9eae8caabb9824e328b9abc988119b39296495ec184c&path=/login.jspa





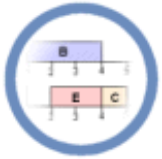
Lab ”6”
Inter-Device Communication

<lecturer, date>

Lab ”6”

- Your task is to create a Wi-Fi P2P app in Android following these steps:
  - Creating and registering a broadcast receiver for your app
  - Discovering peers
  - Connecting to a peer
  - Transfering data to a peer





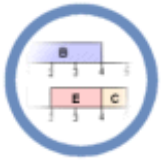
Seminar ”6”
Inter-Device Communication

<lecturer, date>

Seminar ”6”

- Compare network service discovery vs. Service discovery using Wi-Fi P2P connection and discuss how they work and in what sense they differ from each other.





Mini-Project ”6”
Inter-Device Communication

<lecturer, date>

Mini-Project ”6”

- Based on the lectures ”5” and ”6” discuss and compare wireless connectivity in Android vs. iOS.