User Generated Replays

Pedro Silva, João Rodrigues, Rolando Martins, Luís Lopes, Fernando Silva

Context

Objective:

Share user-generated content (videos) over dedicated wireless networks.

No infrastructure is available

Local infrastructure is available
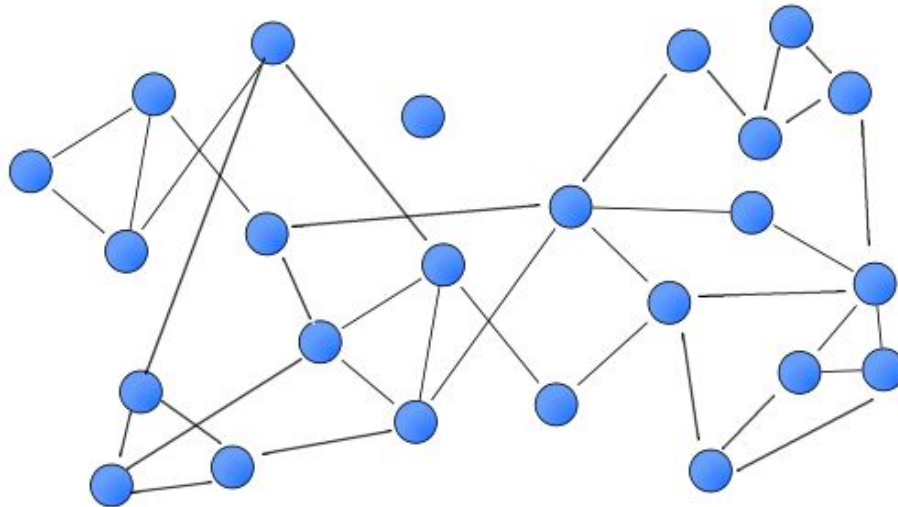(APs, edge-cloud)

Goals

In this presentation we focus on:

1. How to disseminate new content?

2. How to design and test such distributed app?

Is there a low-cost solution?

Standard	Bluetooth 802.15.1	WiFi-Direct	WiFi 802.11ac	WiFi-TDLS	LTE
Frequency Bands	2.4 GHz	2.4 / 5 GHz	5 GHz	2.4 / 5GHz	800 - 2600 MHz
Transfer Rate	25 Mbps	250 Mbps	250 Mbps	250 Mbps	100 / 50Mbps 4G
Range	~10 m	~200m	~200m	~200m	-
Network Size	max 7 connections per device	5-7 per group	max rec. 50	2	-
Connection Time	2-8 s	2-8 s	-	~0s	-

scaling issues

€€€

Can we leverage multiple technologies?

Control Plane

Overlay #1

Send and Receive Metadata

Gather Information

Forward Decision

Data Plane

Overlay #2

Send and Receive Data
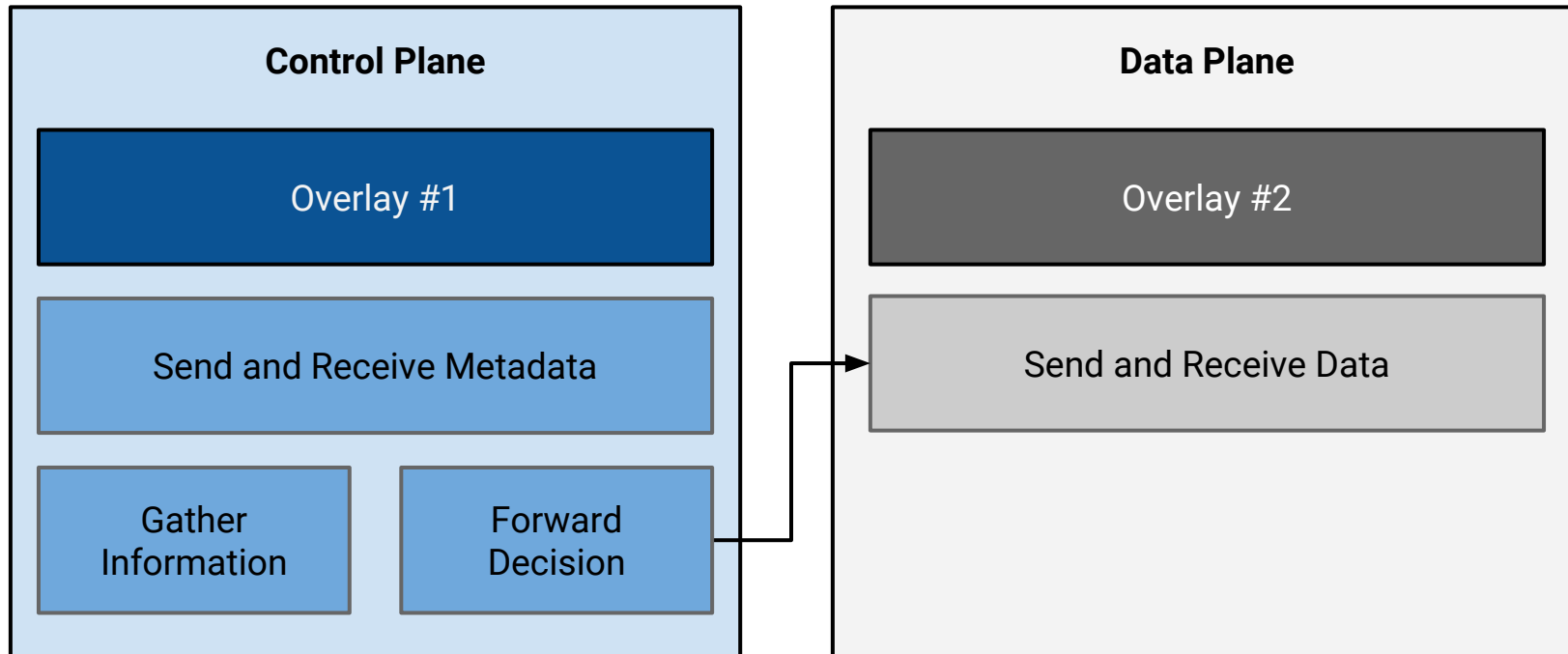
Advantage: Separation of concerns

Disadvantage: Complexity of implementation and testing

Control Plane

Bluetooth

- Typically lower battery consumption

- Exists research for overcoming group size restrictions

- We are exchanging small-size messages (~4KB max)

Issues to address

- How to build the network - algorithms ?

- Which communication protocol to use (gossip vs controlled flooding)?

Data Plane

Strategy	No Infrastructure	Large Infrastructure (€€)	Hybrid (€)
Control Plane	*Bluetooth*	*Bluetooth*	*Bluetooth*
Data Plane	*WiFi-Direct*	*WiFi + TDLS*	*WiFi + TDLS + Direct*

Offers no guarantees:
- Quality of Service
- Scalability

Offers a central control point:
- Supervision of content
- Traffic monitoring
- Caching

7

"Worst-case" Scenario

No infrastructure - we need ad-hoc peer-to-peer networks

+ No cost (€€)
- Hard to guarantee scalability and content availability

Control Plane	Data Plane
Overlay #1	Overlay #2
Send and Receive Metadata	Send and Receive Data
Gather Information / Forward Decision	

Where to start?

start with

App working version

uses →

Prototype
network- algorithms

(formation & communication)

uses ↓

contains ↑

Plug-and-play modular architecture

Testing Setup

Note: The dev and test machines can be the same
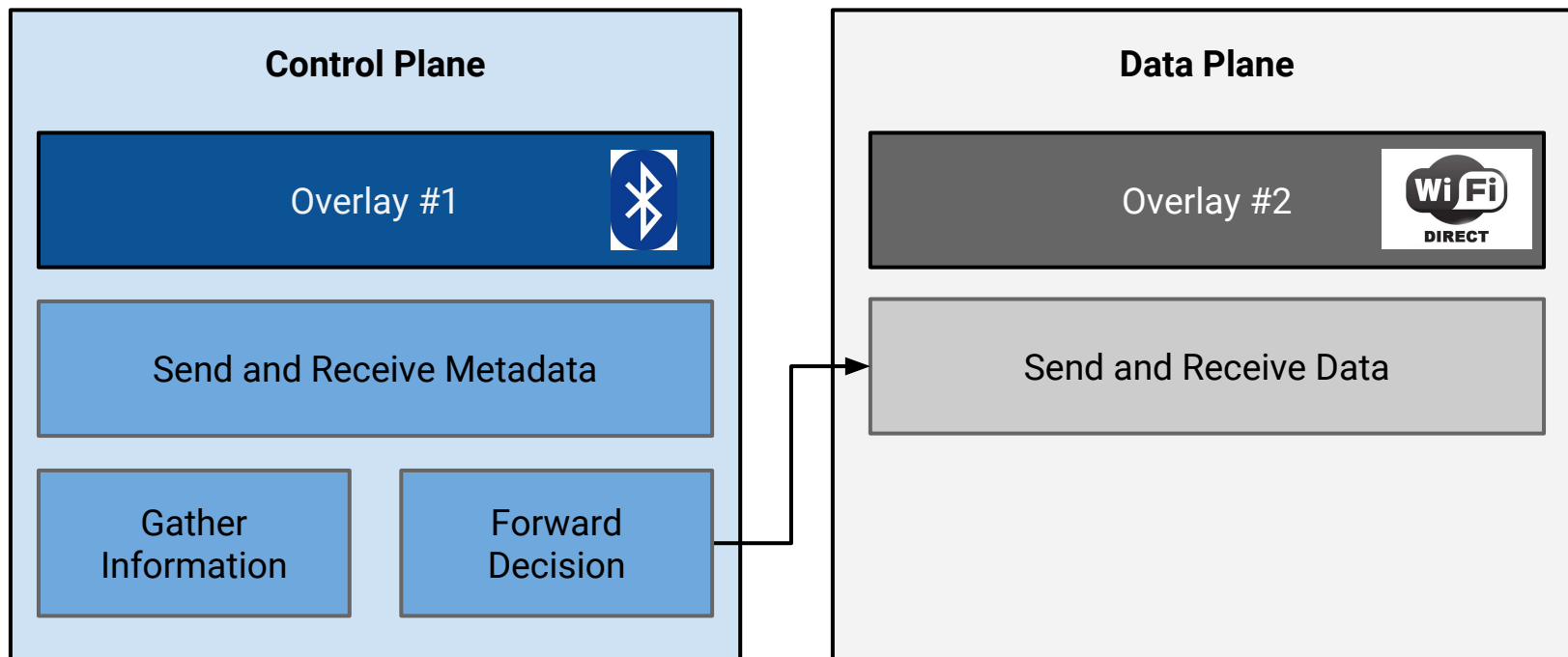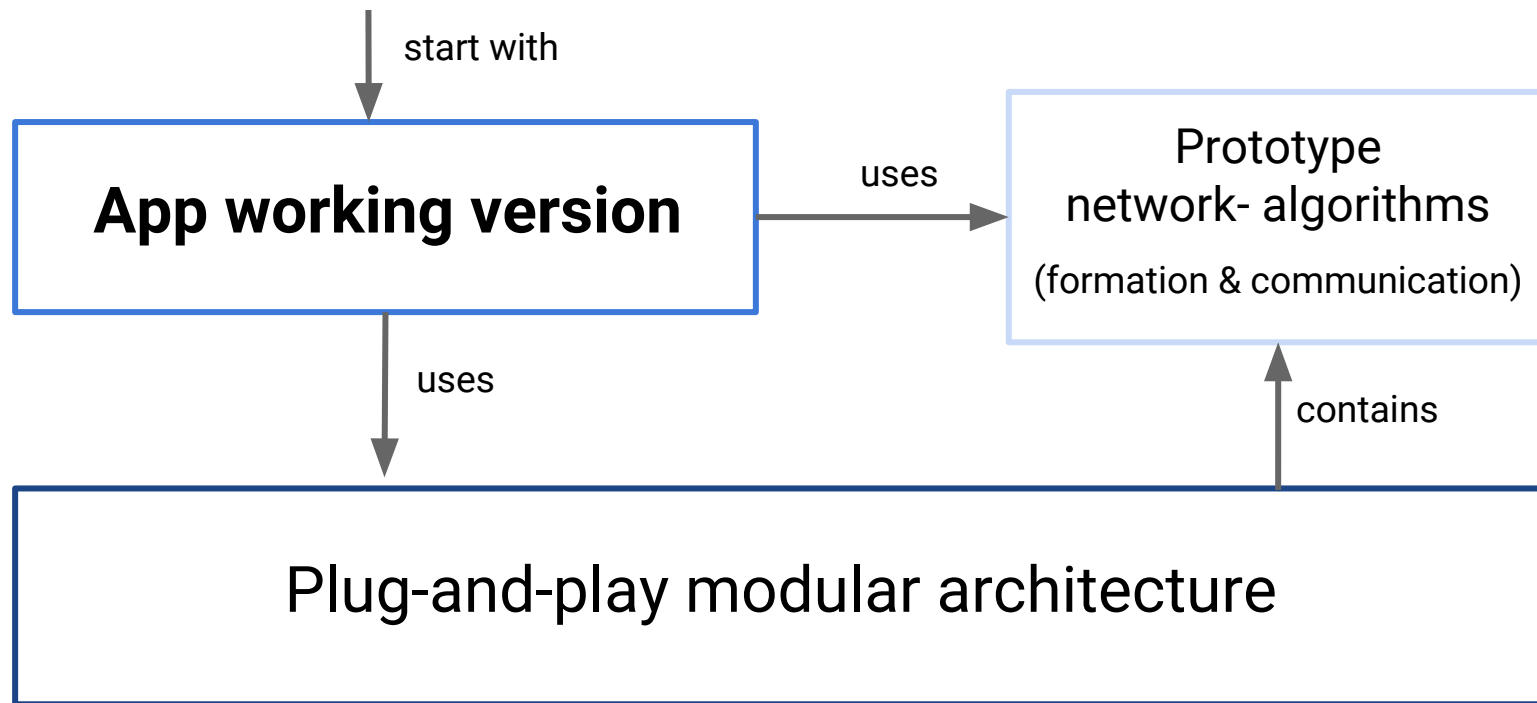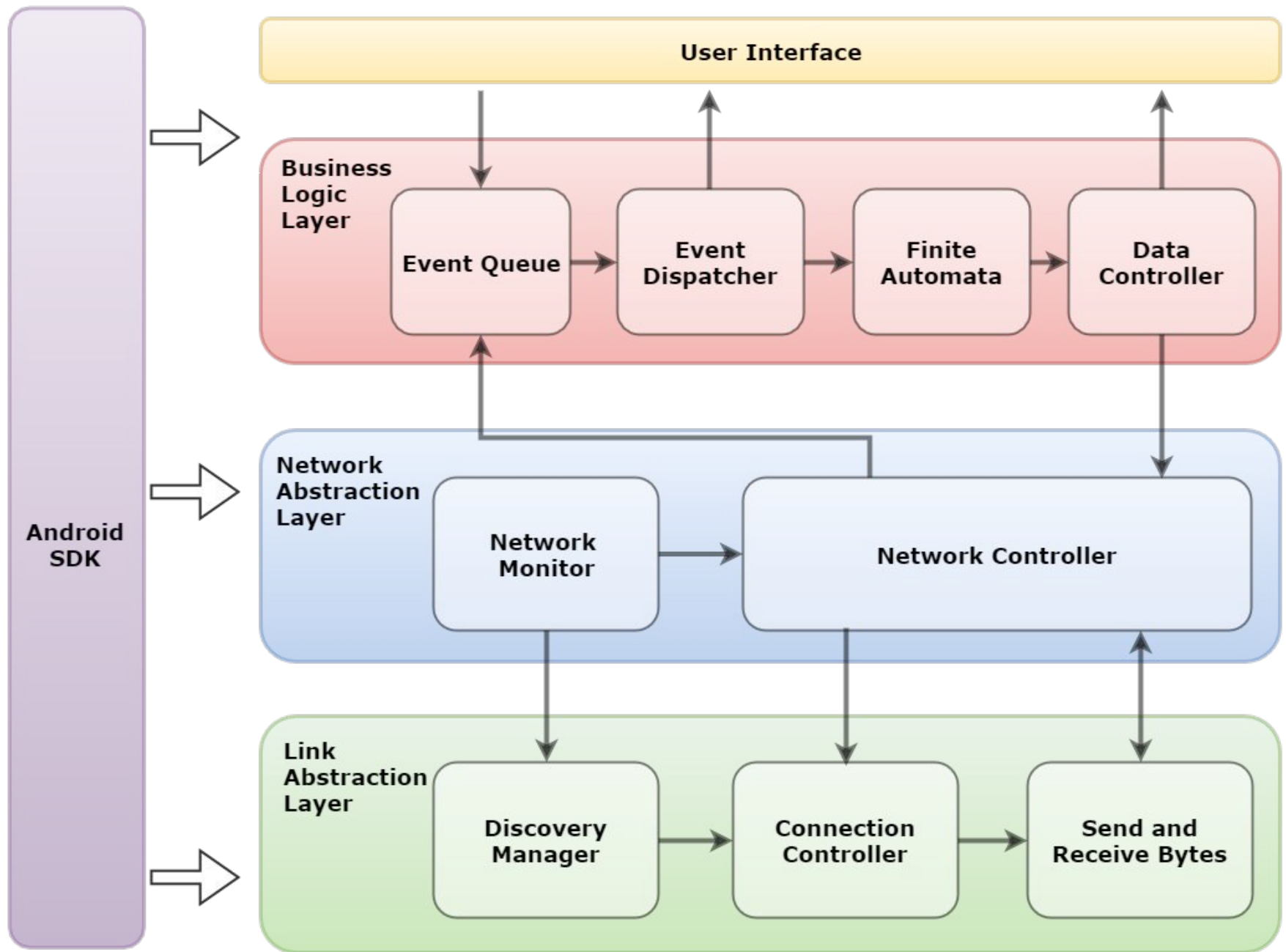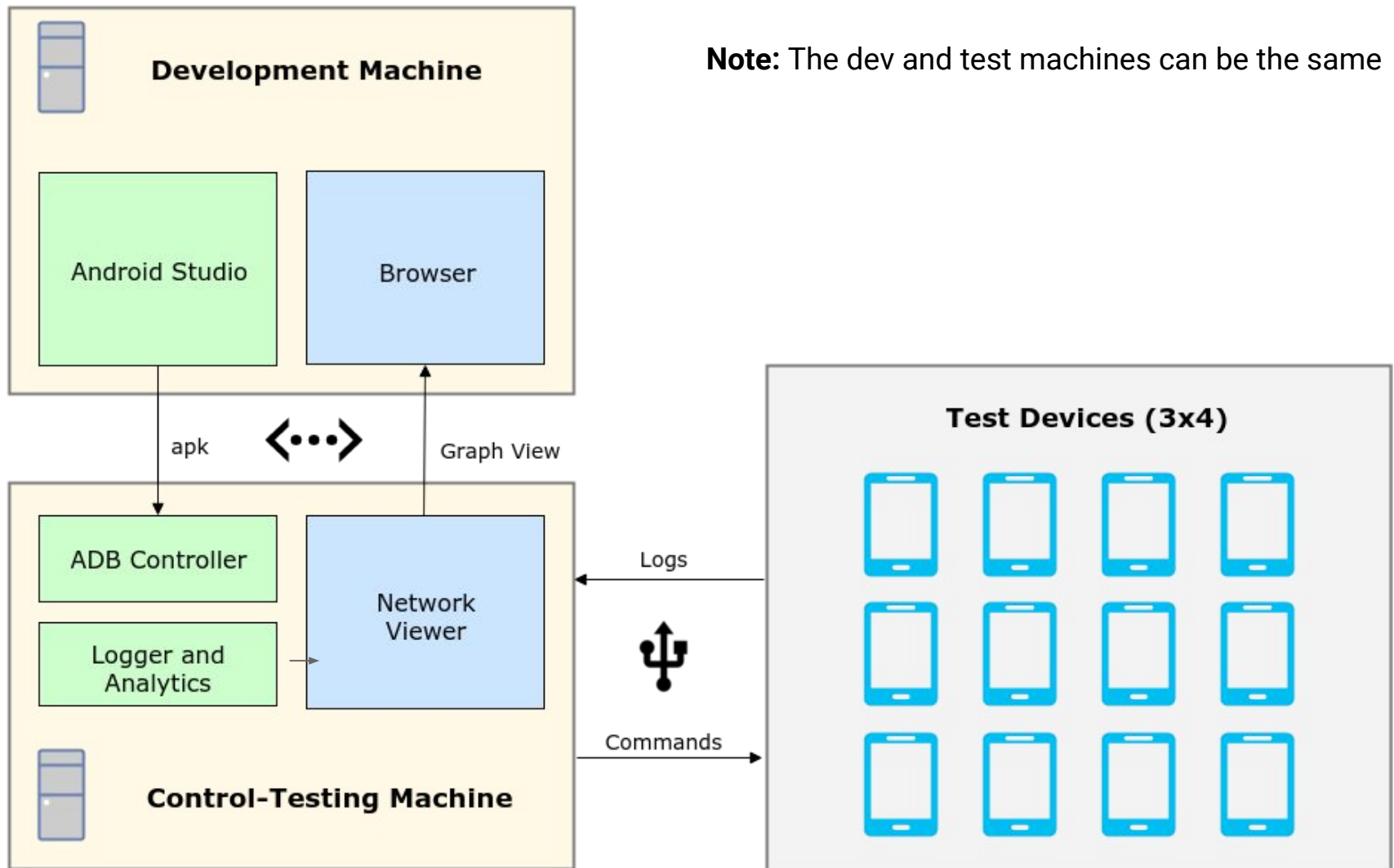
Proof-of-Concept Demo

Material:

- 12 devices - Nexus 9
- 3 producers
- 9 consumers
- 3 videos (8.6 MB, 24.3 MB, 17.5 MB)

WiFi:

Router 802.11n:
- 130 Mbps
- 20MHz/channel
- 4 channels available
- 4MB/s effective bandwidth
(we use 1 channel)

WiFi-Direct:

Legacy Mode 802.11ac:
- 450 Mbps
- 40MHz/channel
- 8 channels available
- 7MB/s effective bandwidth
(we use 1 channel)

Step #1 - Setup

0 - Devices are disconnected

1 - Devices are connecting

2 - All devices are connected (bluetooth)

producers   consumers

4 - Producer shares video metadata

4 - Repeat steps 3 and 4 for other two producers

3 - Producer creates video
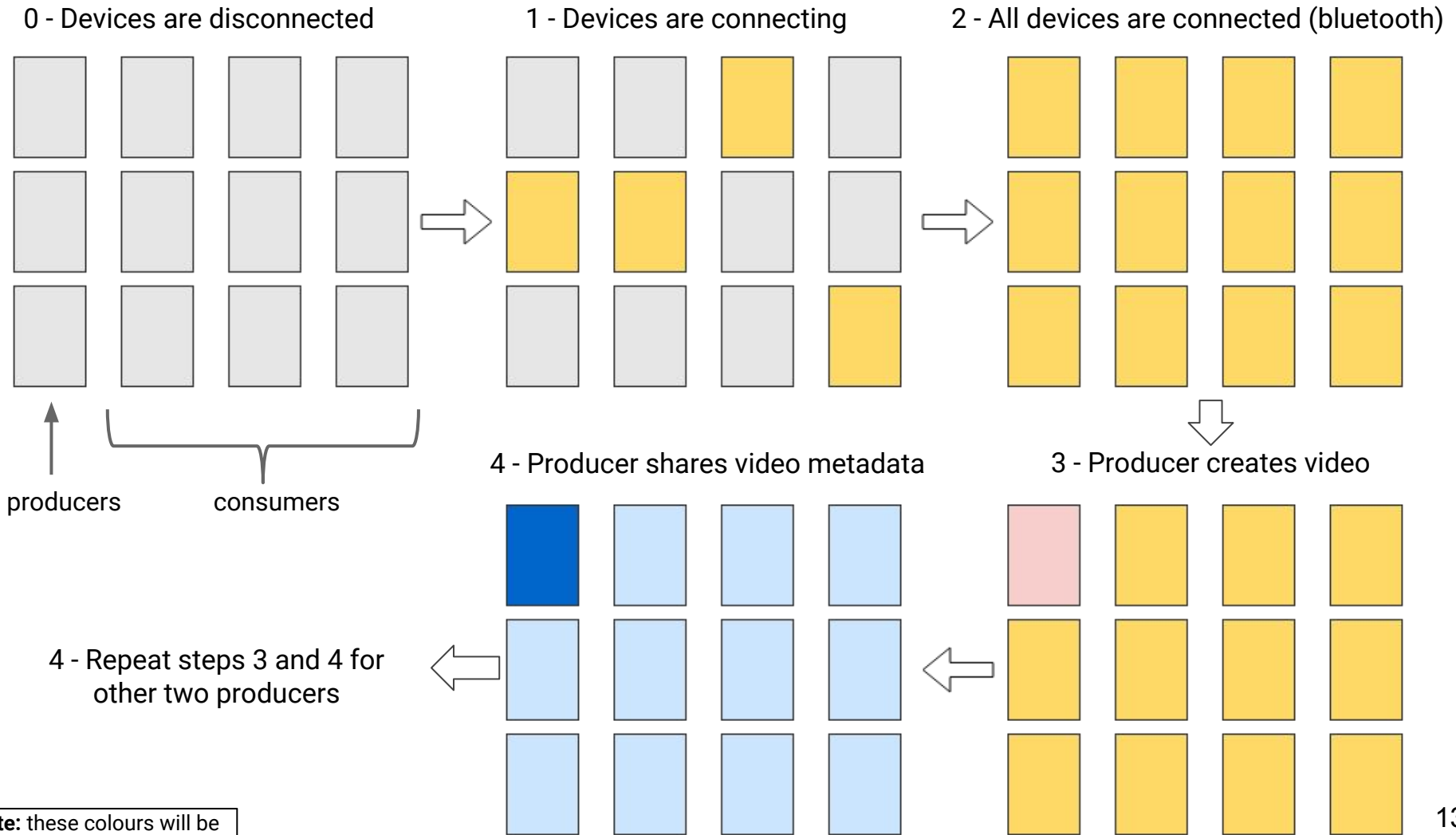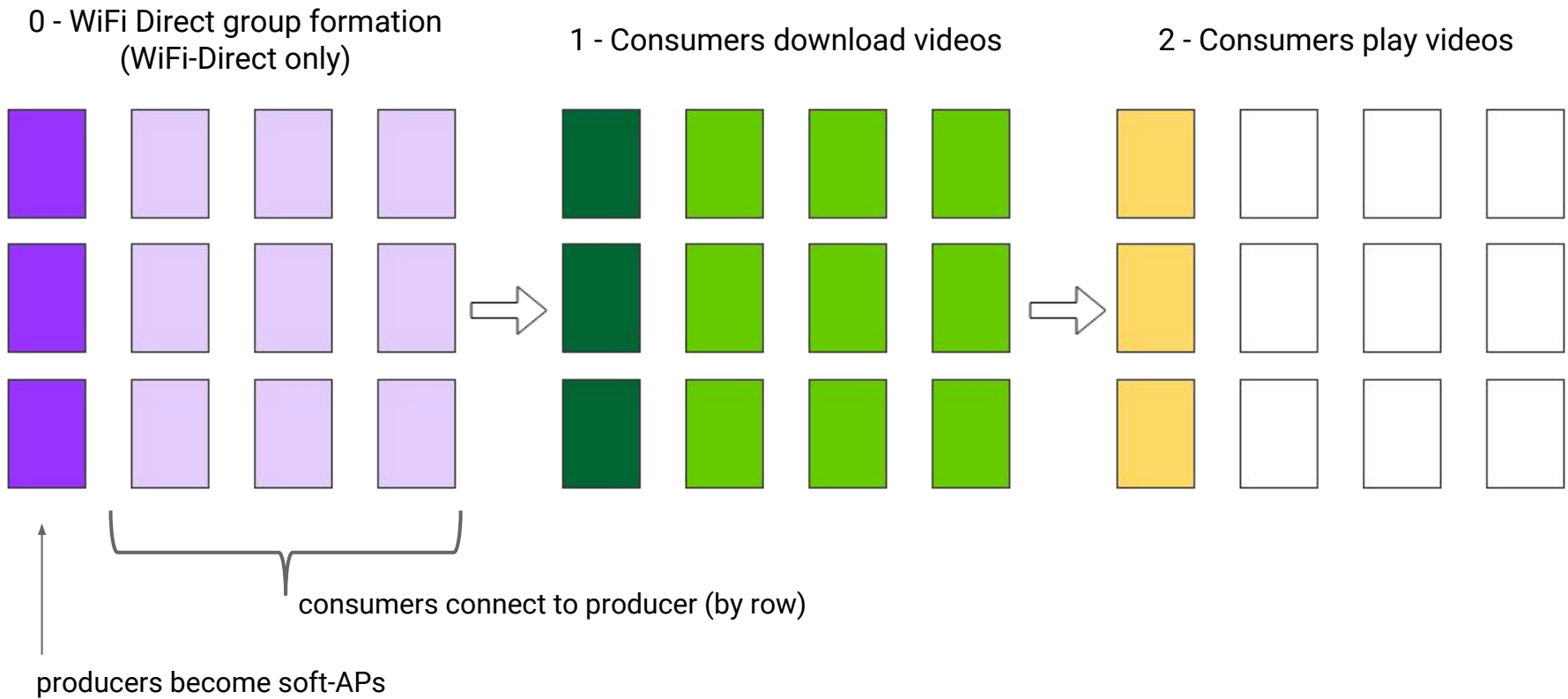
Note: these colours will be observed during the video

13

Step #2 - Data Transfer

0 - WiFi Direct group formation
(WiFi-Direct only)

1 - Consumers download videos

2 - Consumers play videos

consumers connect to producer (by row)

producers become soft-APs

Demonstration Video

Network Viewer
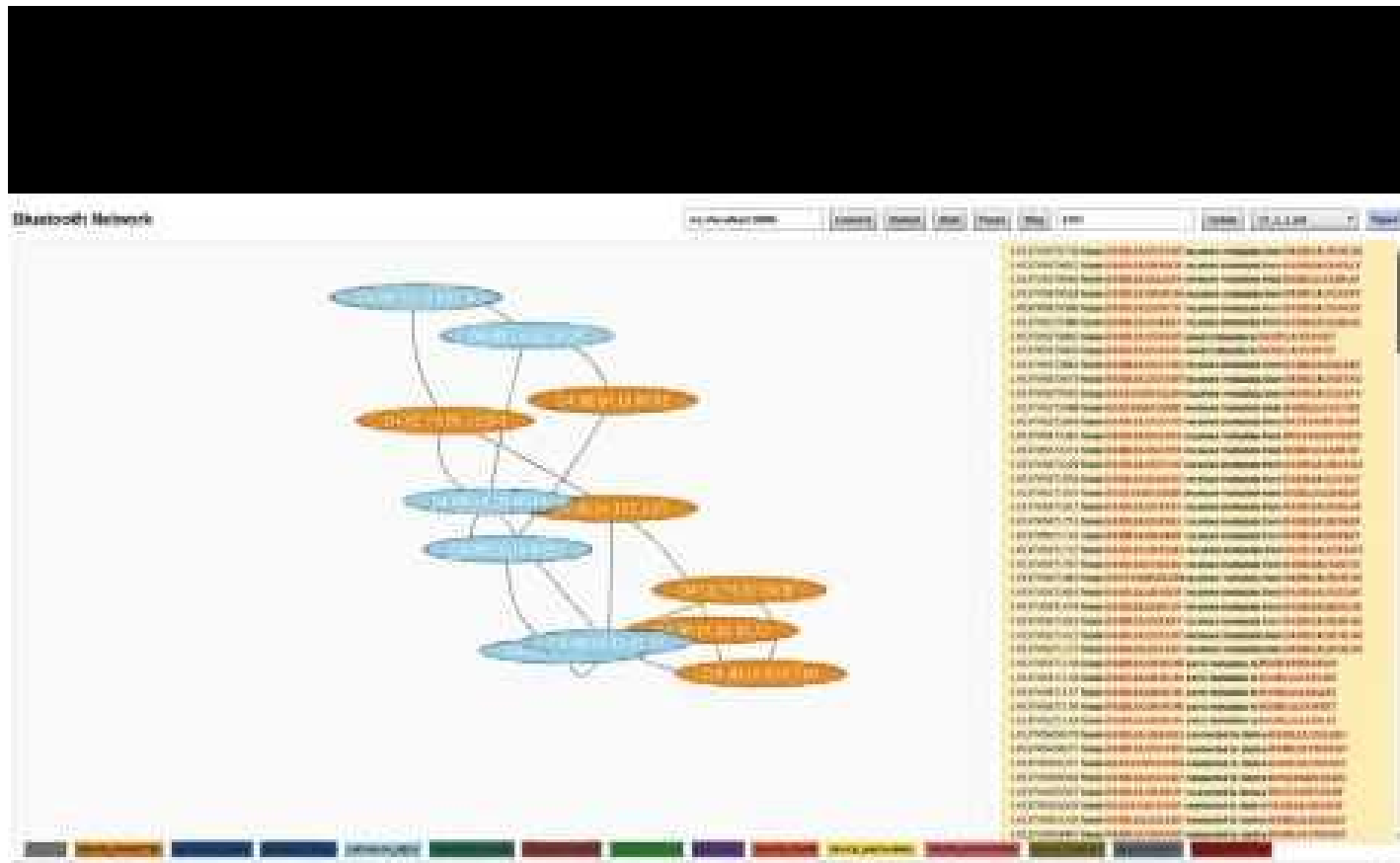
Measurements

Technology	Connection Overhead (1)	Mean Transfer Rate (2)	Mean Transfer Rate w/Ov (3)
Wifi	-	950 kB/s	-
Wifi-Direct	3.7s	3094 kB/s	1921 kB/s

(1) - Time spent connecting to AP

(2) - Averaged across all devices - *not accounting for (1)*

(3) - Averaged across all devices - *accounting for (1)*

Bluetooth

Network formation time = 40 +- 15 s (algorithm-dependent)

Future Work

1. API enhancement + test suites

2. Test algorithms and app behavior at scale
   - Compare strategies, algorithms and comm. protocols
   - Use simulator ns3 - Joaquim Silva

3. Address security issues
   - Identity theft and authentication mechanisms
   - Content validation - safe for viewing

The End - Thank you

2-Tier Strategies

Strategy	Control Plane (~4KB)	Data Plane (~4MB)	Advantages	Disadvantages
#1	*Bluetooth*	*WiFi*	Centralized control and supervision over content and network traffic	Requires infrastructure (€€) AP saturation in over-crowded environments
#2	*Bluetooth*	*WiFi-Direct*	Parallelization No infrastructure required	Behavior in scarce and greedy environments Limited range
#3	*Bluetooth*	*WiFi-LTE & WiFi-Direct*	Central control Parallelization	Requires infrastructure (€) Requires decision mechanism

How to disseminate content?

Can a single technology be the answer to this problem?

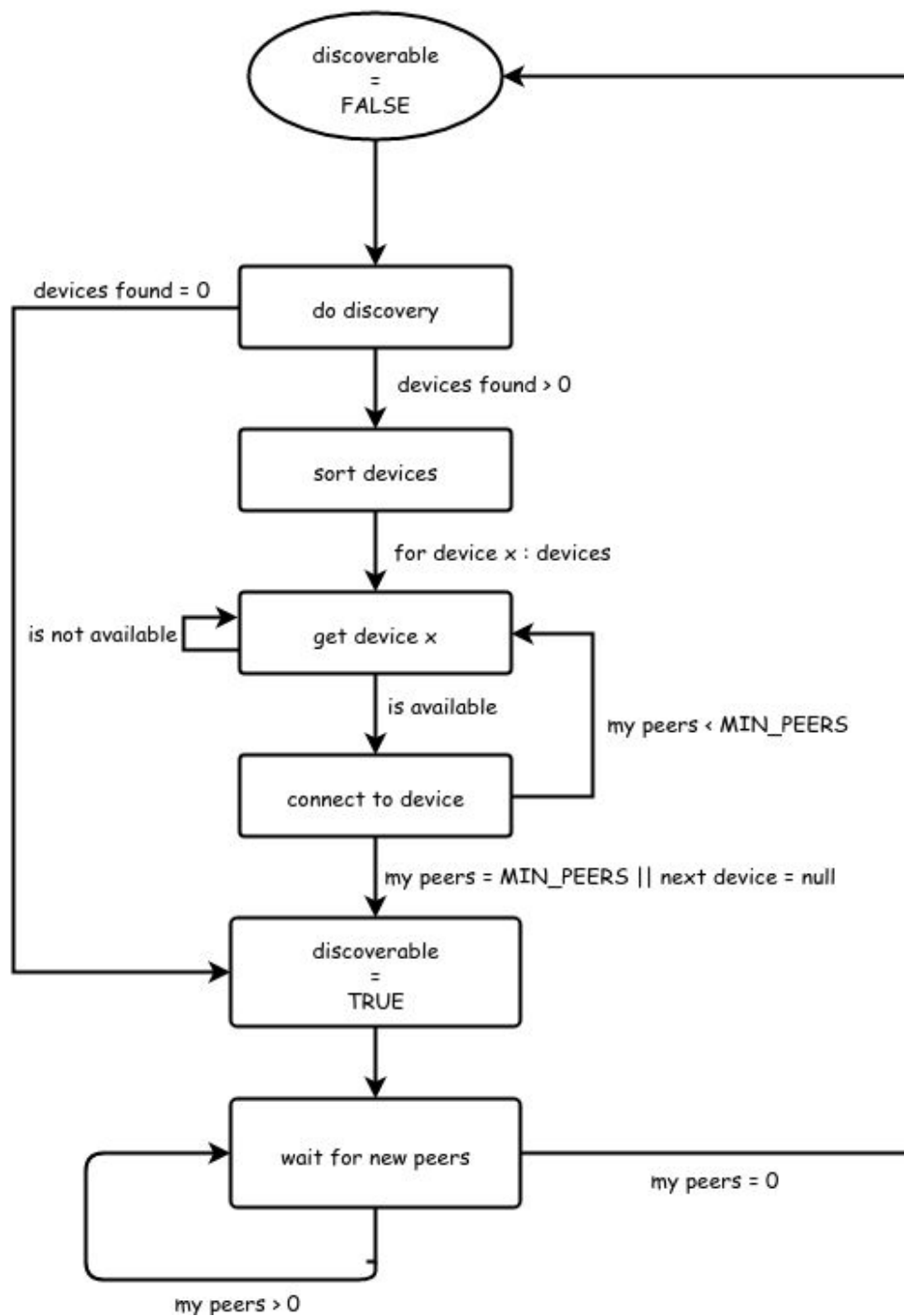| Technology	Advantages	Disadvantages
Bluetooth	No infrastructure required	Low range and transfer rates Restricted network size
WiFi-Direct	No infrastructure required Parallelization	Restricted network size Complexity
WiFi	Centralized control and supervision over content and network traffic	Requires infrastructure ($$) AP saturation

Proof-of-Concept Demo

Step #1 - Setup

1. Create control network - simple prototype algorithm
2. Producers create and share videos - flooding algorithm
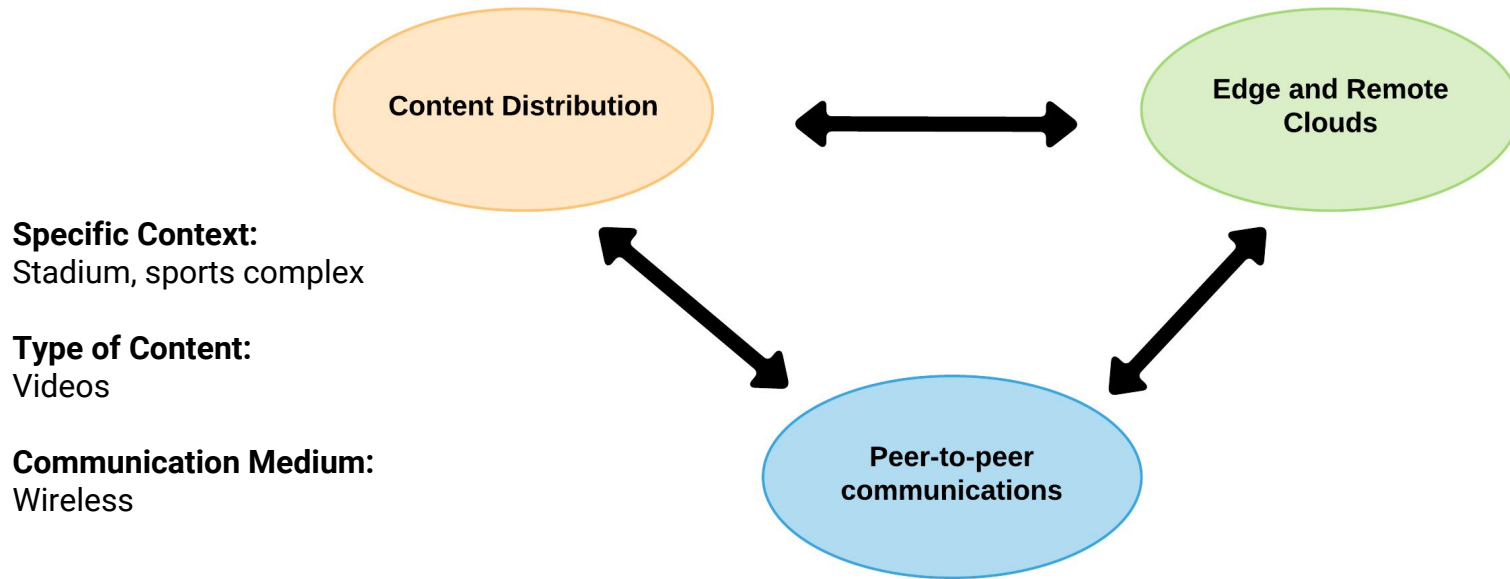
Step #2 - Data Transfer

1. Consumers download videos from producers concurrently
2. Repeat this step for WiFi and WiFi-Direct

```
                          ┌─────────────┐
                          │ discoverable │◄──────────────────────┐
                          │      =       │                       │
                          │    FALSE     │                       │
                          └─────────────┘                       │
                                 │                               │
                                 ▼                               │
  devices found = 0      ┌─────────────┐                        │
 ◄─────────────────────  │ do discovery │                        │
 │                       └─────────────┘                        │
 │                              │                                │
 │                       devices found > 0                      │
 │                              ▼                                │
 │                       ┌─────────────┐                        │
 │                       │ sort devices │                        │
 │                       └─────────────┘                        │
 │                              │                                │
 │                      for device x : devices                  │
 │                              ▼                                │
 │       is not available┌─────────────┐                        │
 │      ┌───────────────►│ get device x │◄───────────┐          │
 │      └────────────────└─────────────┘            │          │
 │                              │                     │          │
 │                         is available       my peers < MIN_PEERS
 │                              ▼                     │          │
 │                       ┌─────────────┐             │          │
 │                       │connect to device│─────────┘          │
 │                       └─────────────┘                        │
 │                              │                                │
 │         my peers = MIN_PEERS || next device = null           │
 │                              ▼                                │
 │                       ┌─────────────┐                        │
 └──────────────────────►│ discoverable │                        │
                         │      =       │                        │
                         │    TRUE      │                        │
                         └─────────────┘                        │
                                │                                │
                                ▼                                │
                         ┌─────────────┐                        │
                    ┌───►│wait for new  │───────────────────────┘
                    │    │   peers      │      my peers = 0
                    │    └─────────────┘
                    └──────────┘
                      my peers > 0
```

23

Pseudo code (Simple P2P - First Discovered & Available, First Connected**):**

```
start: setDiscoverable(false)
        devices = discovery(timeout, 5 12)
        IF (devices != null)
                sortDevicesBy(devices, "rssi")
                FOR device : devices
                        IF (isPeerAvailable(device))
                                connectTo(device)
                                IF (countPeers() < MIN_PEERS)
                                        continue
                                ELSE
                                        break
        setDiscoverable(true)
listen:         wait_for_new_peers(timeout)
        IF (peersCount() == 0)
                goto start
        ELSE
                goto listen
```

Problem

Specific Context:
Stadium, sports complex

Type of Content:
Videos

Communication Medium:
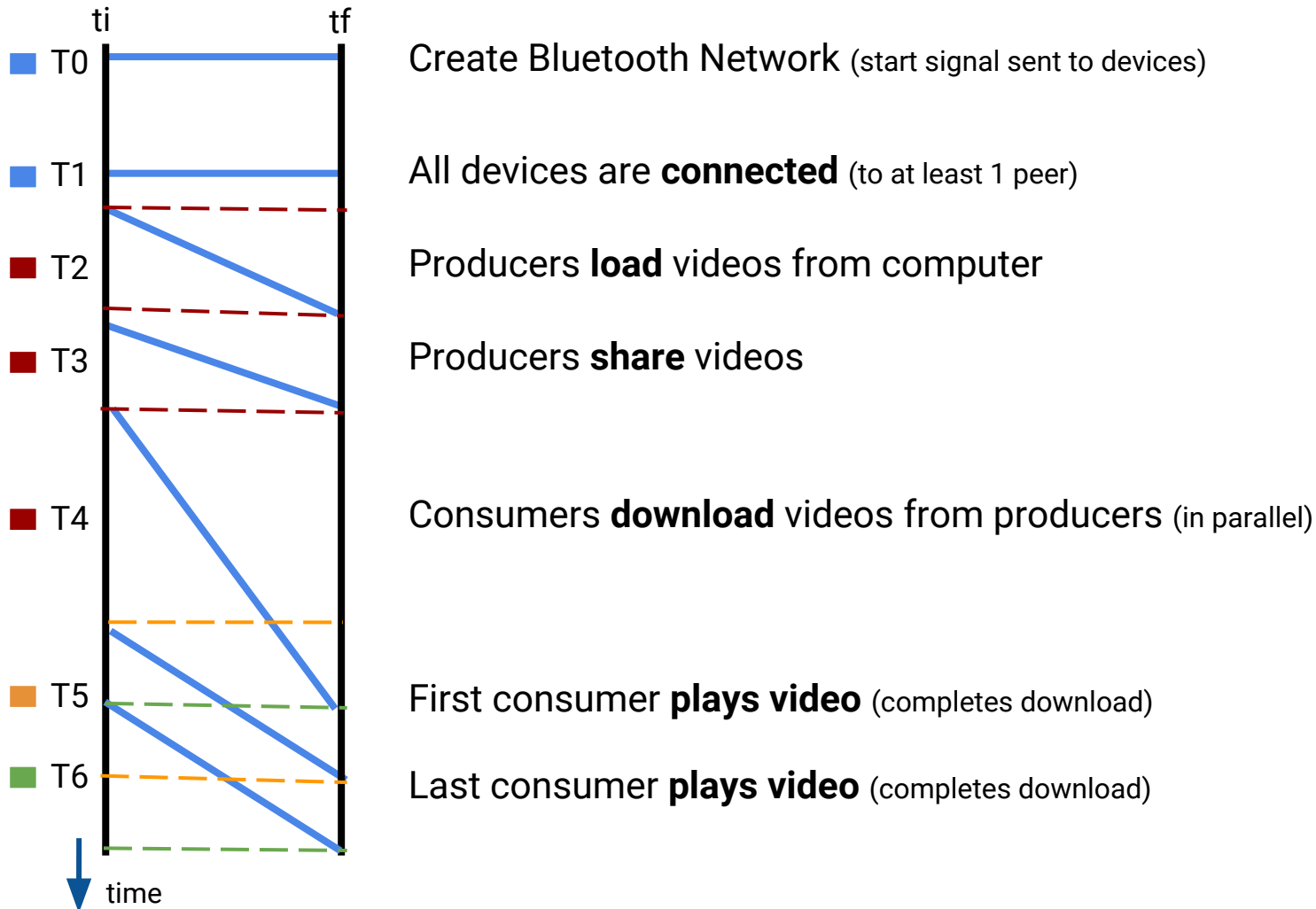Wireless

Challenges:

- How to leverage peer-to-peer connections for sharing content (human-usable)?

- Dynamic, highly volatile environment and networks.

- Define mechanisms and strategies for content sharing and caching (application/client-wise).

- How can edge clouds augment content dissemination and visualization?

- What are the roles and interactions between edge and remote clouds.

APIs

- Each layer provides a library with its own API.

- Up-down library dependency.

- To be shared as soon as considered stable.

- Each layer has its own test-suite (to do)

hyrax / hyrax-android-app
Nice!!

hyrax / HyraxNetworkLayerLibrary

hyrax / hyrax-business-layer

hyrax / Network-Simulations
Project containing the network simulation code.

hyrax / networkViewer

hyrax / hyrax-nomad
Framework nomad. Share any type of content over wir…

hyrax / hyrax-nomad-test
Android project for testing and extending the nomad fr…

hyrax / HyraxLinkLayerLibrary

Experiment #1 Timeline

ti   tf

■ T0   Create Bluetooth Network (start signal sent to devices)

■ T1   All devices are **connected** (to at least 1 peer)

■ T2   Producers **load** videos from computer

■ T3   Producers **share** videos

■ T4   Consumers **download** videos from producers (in parallel)

■ T5   First consumer **plays video** (completes download)

■ T6   Last consumer **plays video** (completes download)

time

Experimental Protocol

Setup

1. Create bluetooth network
2. Producers **load** videos from computer
3. Producers **share** videos

Scenario #1 - Parallel dissemination

- Consumers download videos from producers concurrently.
- Represents **Best-case** scenario - faster dissemination

Scenario #2 - Sequential dissemination

- Videos are disseminated "left-to-right" sequentially
- Represents **Worst-case** scenario - slower dissemination

Results - Setup

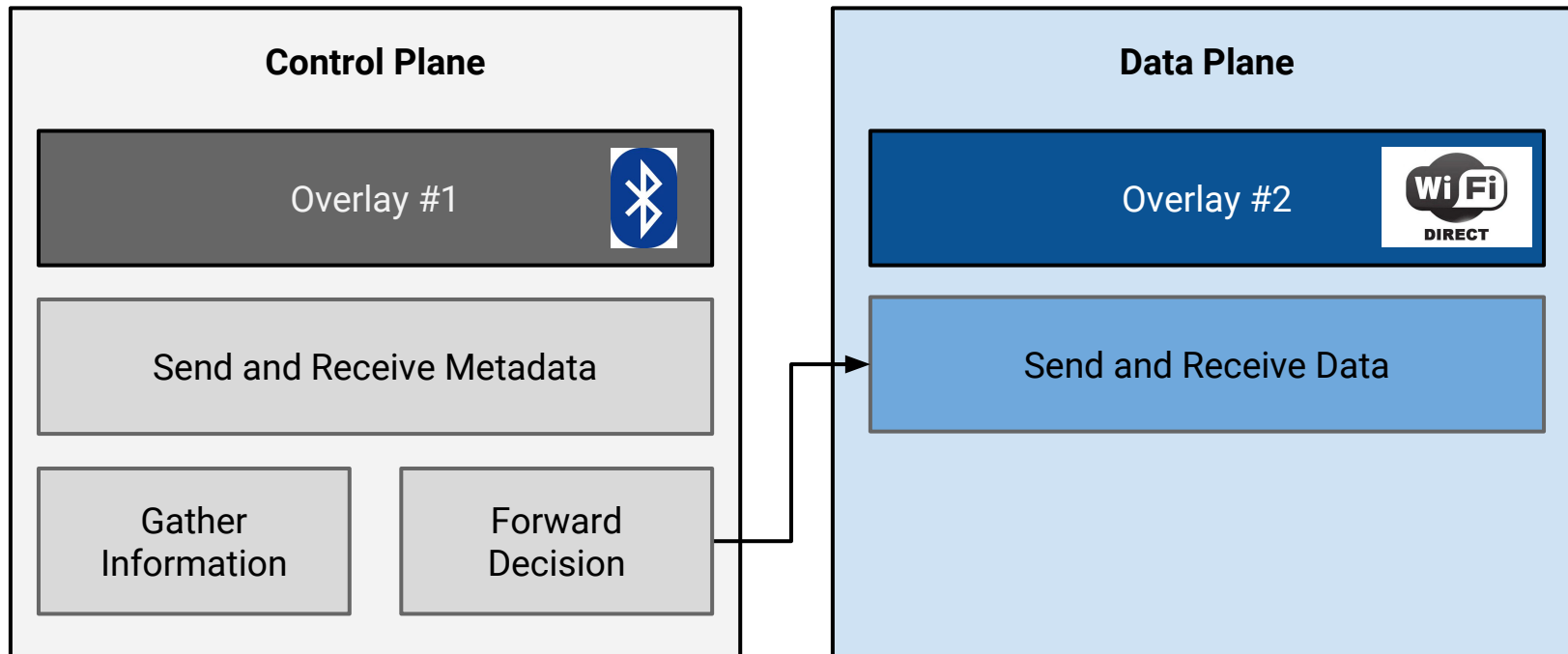| Technology	Observed Bandwidth	Network Formation Time (1)	Socket Creation Time (2)	Hop Overhead (3)
Bluetooth	~ 200 KB/s	~40 s	~ 3 s	200 ms/hop

Router Bandwidth - 4 MB/s

(1) -

(2) -

(3) -

(4) -

Current Strategy (#2)

Decouple network control and data planes

Dissemination Strategies

Strategy	Metadata & Control (~4KB)	Data (~4MB)	Advantages	Disadvantages
#1	Bluetooth	Bluetooth	No infrastructure required	Low range and transfer rates Network size restrictions
#2	WiFi-Direct	WiFi-Direct	No infrastructure required Parallelization	Network size restrictions Complexity
#3	WiFi	WiFi	Centralized control and supervision over content and network traffic	Requires infrastructure ($$) AP saturation

Dissemination Strategies

Strategy	Metadata & Control (~4KB)	Data (~4MB)	Advantages	Disadvantages
#4	Bluetooth	WiFi	Centralized control and supervision over content and network traffic	Requires infrastructure ($$) AP saturation in over-crowded environments
#5	Bluetooth	WiFi-Direct	Parallelization No infrastructure required	Behavior in scarce and greedy environments Limited range
#6	Bluetooth	WiFi-LTE & WiFi-Direct	Central control Parallelization	Requires infrastructure ($) Requires decision mechanism

Common Advantage: Separation of concerns

Common Disadvantage: Complexity of implementation and testing

Context

Objective:

Share user-generated content (videos) over dedicated wireless networks.

Study Cases:

#1 - No infrastructure is available

#2 - Local infrastructure is available: APs, edge-cloud

Technologies:

WiFi, WiFi-Direct, WiFi-TDLS, Bluetooth, LTE

We focus on:

1. How to disseminate content?
2. How to implement and test such distributed Android application?

User Generated Replays

Pedro Silva
João Rodrigues