



Adding Unusual Transports to The Serval Project

Alexandros Tsiridis
&
Joseph Hill

Supervised by:
Dr. Paul Gardner-Stephen

The Serval Project



Serval is a telecommunications system comprised of at least two mobile phones that are able to work outside of regular mobile phone tower range due thanks to the Serval App and Serval Mesh.

Uses various communication technologies.

- WiFi (Client / AP / Ad-hoc)
- Bluetooth (Device discovery / Peering)
- UHF
- Always looking for more...

Wi-Fi Direct



Wi-Fi Direct is a certification mark for devices that implements the Wi-Fi Peer-to-Peer (P2P) specification. Wi-Fi Peer-to-Peer is a technology that allows two or more devices to communicate directly without the need of an access point. The following list is components of the P2P architecture:

1. P2P Device
2. P2P Group Owner role
3. P2P Client role

Our Research Goals



1. How can Wi-Fi Direct be used to support the Serval Mesh network on the Android platform?
2. Can the discovery protocols supported by Wi-Fi Direct be used to transfer data in a mesh topology
3. Which types of Serval traffic can be supported and what performance can be obtained using this technique?

Related Work

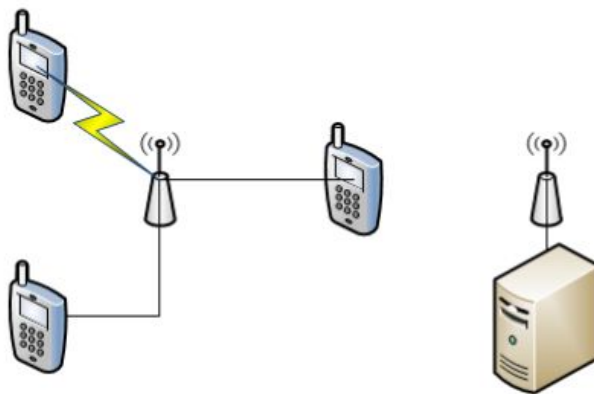


Bluetooth:

- Uses the name of the device to transfer broadcast data.
- It also forms connections to send unicast data

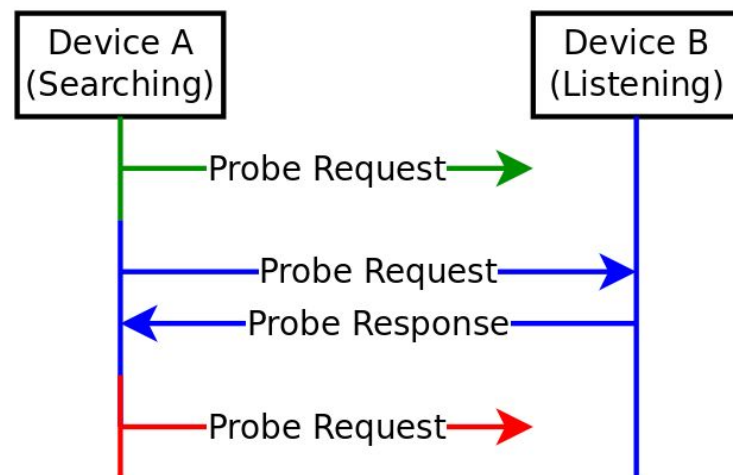
Automatic Android-based Wireless Mesh Networks[1]:

- AP
- Client
- Relay



Device Discovery

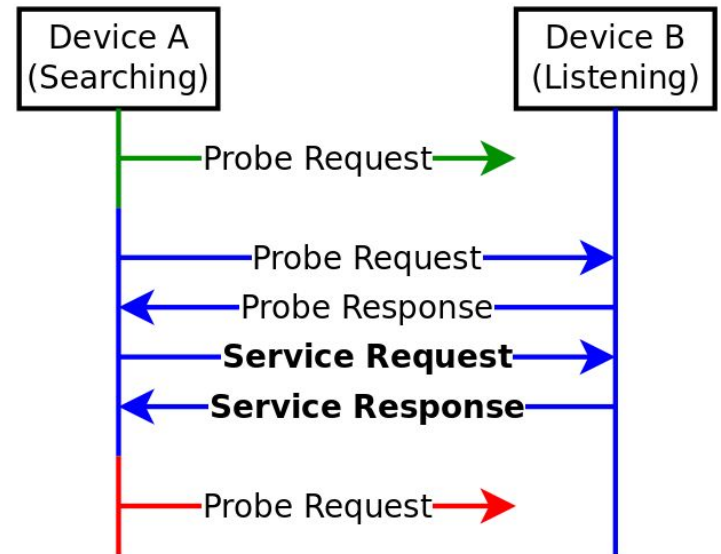
- Social Channels
 - 1, 6, 11 (2.4 Ghz)
- Find Phase
 - Requires complementary device states
 - Listen State
 - Device listens on randomly chosen channel
 - Remains the same until P2P discovery completes
 - For random amount of time
 - Listens at least 500 ms every 5 seconds
 - Search State
 - Device sends probe requests on each of the social channels
 - Will not respond to probe requests



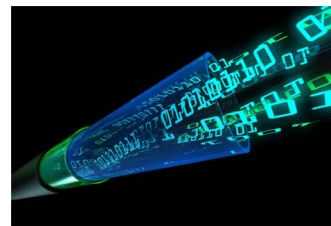
Service Discovery

This functionality of the Wi-Fi P2P technology allows devices to check what services an already discovered (not connected) device offers. This is done by exchanging frames between the discovered devices prior to group formation using the Generic Advertisement Service (GAS) protocol/frame exchange.

- Dependent on device discovery
- Done while device is in Search State
- Result may be up to 64 K of data
- Supports Fragmentation
- Initiated upon request, not continuously running



Service Discovery As A Transport



- Service discovery can be used as a poll based transport.
- One device asks the other devices if they have data for it.
- A device adds local services in order to send data to other users.
- The specification uses unicast to send data.
- No need for user interaction.
- Can be automated.
- It can communicate with everyone in range.

Our Method



- Source will make data available by encoding it in a service string
- Destination will retrieve data by querying for a service with its own ID
- TCP behavior (Window, ACK , SEQUENCE)
- Convert Serval messages to byte stream and back
- Use of Universal Plug and Play (UPnP) Service Discovery protocol
- UUID encodes:
 - Acknowledgment numbers
 - Fragment numbers, for multiple services in a single response
 - Sequence numbers
 - Destination ID

uuid:0000049f-0001-0576-4d3a-5108fcbb9a2f::XLTS9kJD...

Acknowledgment Fragment Sequence Destination ID Base64 Encoded Data

The diagram shows a UUID string with brackets and labels identifying its components. The string is 'uuid:0000049f-0001-0576-4d3a-5108fcbb9a2f::XLTS9kJD...'. Brackets and labels below the string identify: '0000049f' as Acknowledgment, '0001' as Fragment, '0576' as Sequence, '4d3a-5108fcbb9a2f' as Destination ID, and 'XLTS9kJD...' as Base64 Encoded Data.

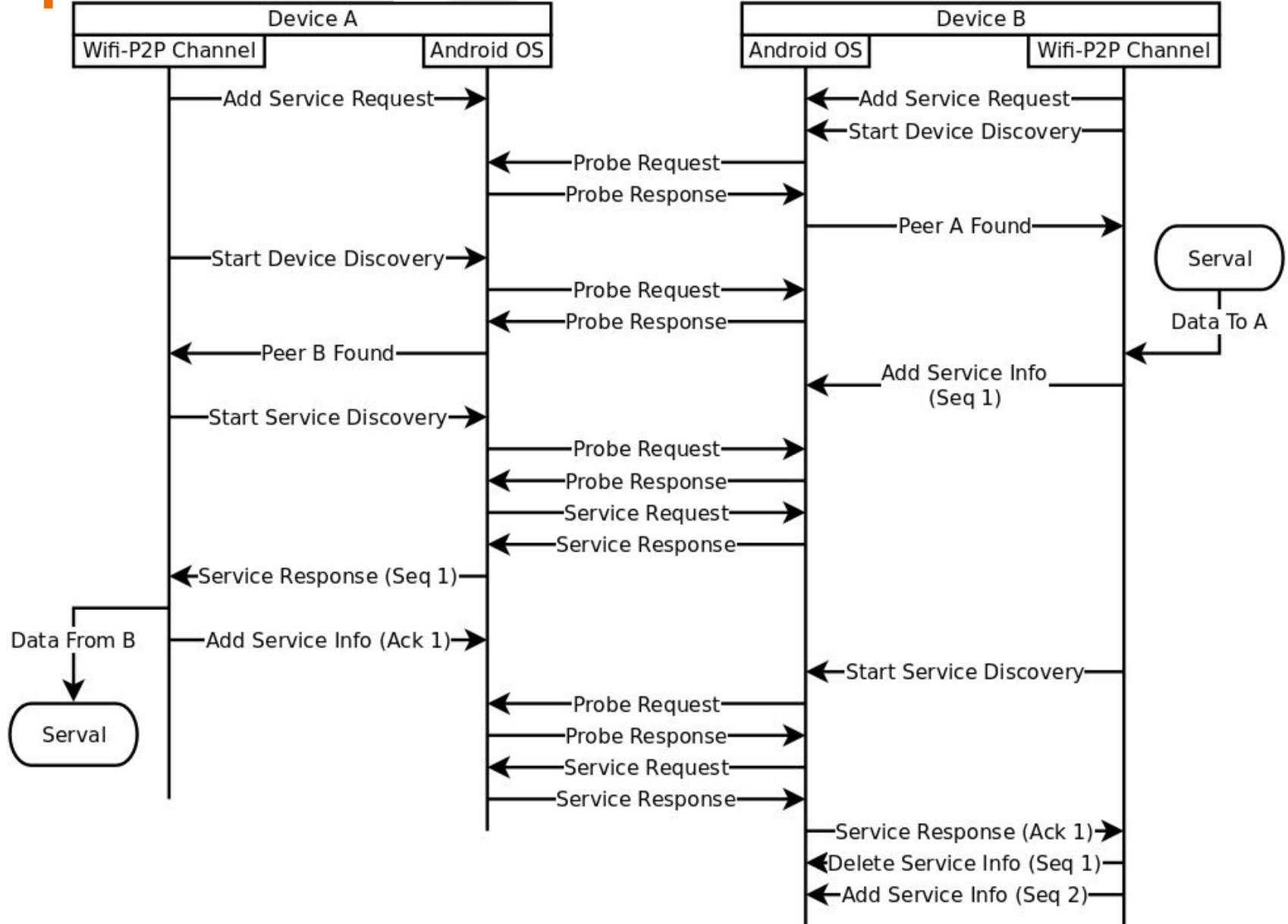
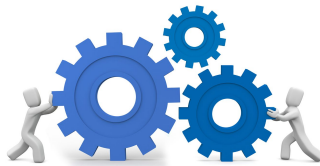
API Limitations & Bugs



- Wifi framework fulfills requests without notifying the application
- API does not provide indication of service discovery state
- Response size limits significantly lower than 64 K allowed by specification
 - Large responses not delivered to application
 - Fragmentation not implemented on most devices
- Different versions of Android OS behave differently
 - Limits in service strings, response size, request size
- Malformed packets generated after 128 different service requests

Length (2 Octets)	Protocol (1 Octet)	Transaction ID (1 Octet)	Query (Length - 2 Octets)
0x02 0x00 (2)	0x00 (0)	0x7e (126)	-
0x02 0x00 (2)	0x00 (0)	0x7f (127)	-
0x02 0x00 (2)	0x00 (0)	0xff 0xff 0xff 0x80 (-128?)	-

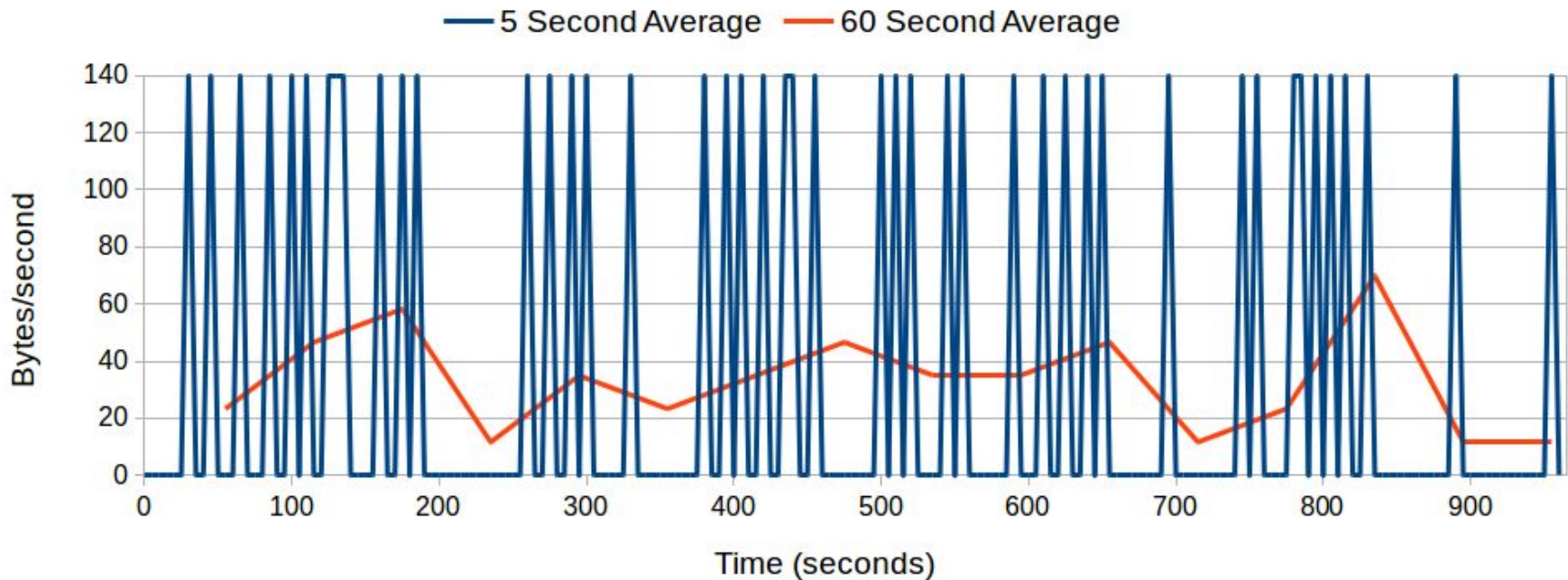
Implementation



Testing And Results

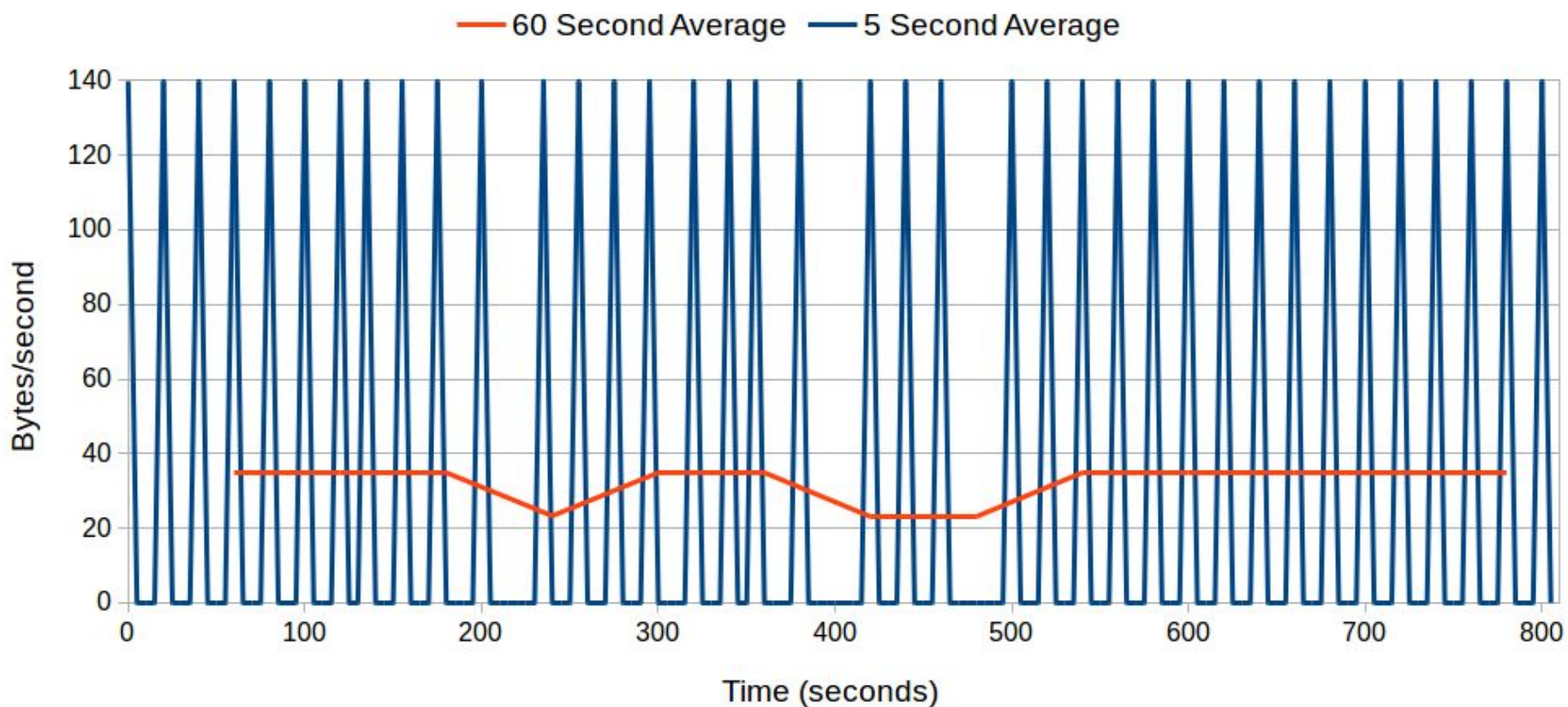


- Service Response Size
 - Different limits on different devices
 - Severely impacted by implementation issues
- Service Discovery Interval
 - Requires some randomization
 - 5 to 10 second interval, 11.65 - 69.9 B/s 60 second averages, 32.7 B/s overall

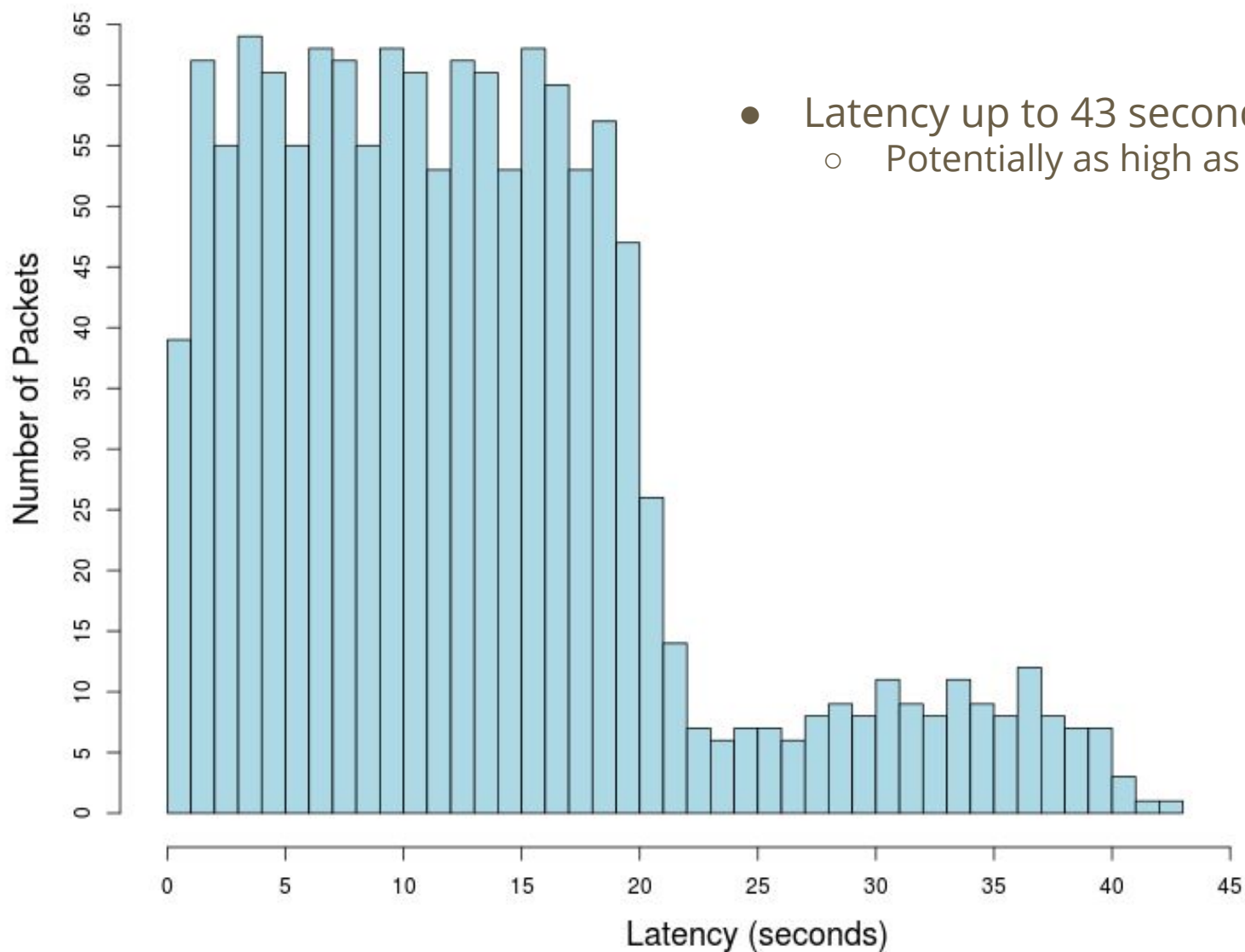


Testing And Results

- Throughput also of 32.7 Bytes/second with 18 to 20 second interval



Testing And Results

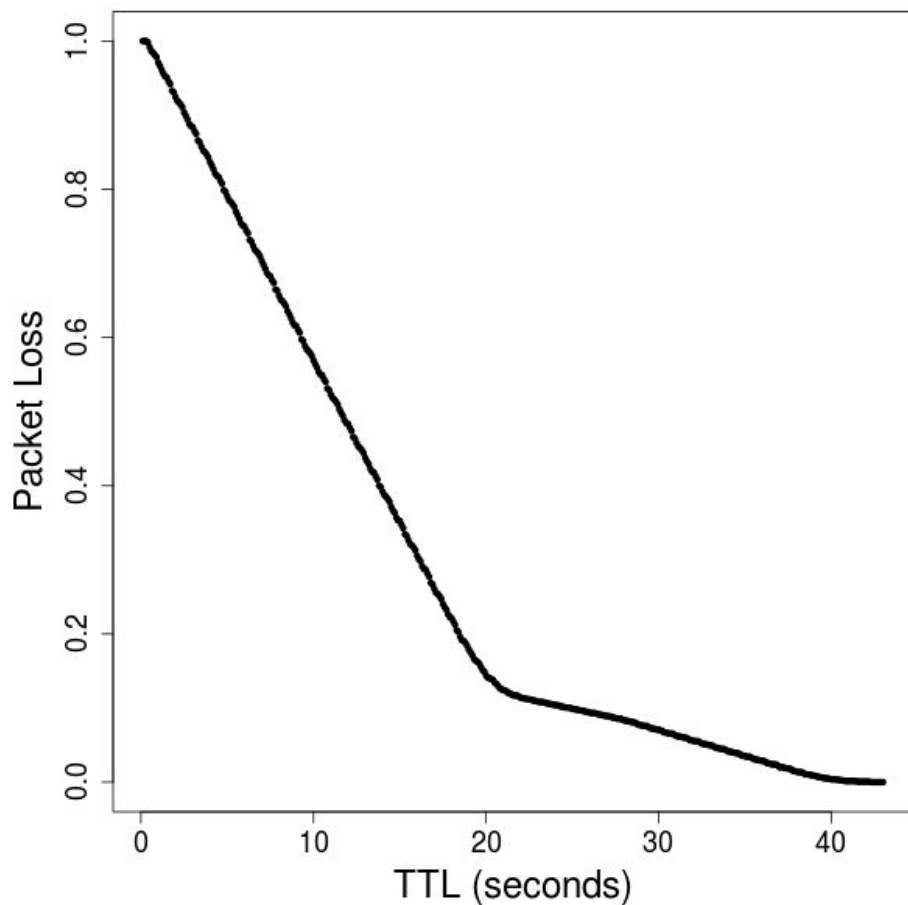


- Latency up to 43 seconds
- Potentially as high as peer timeout

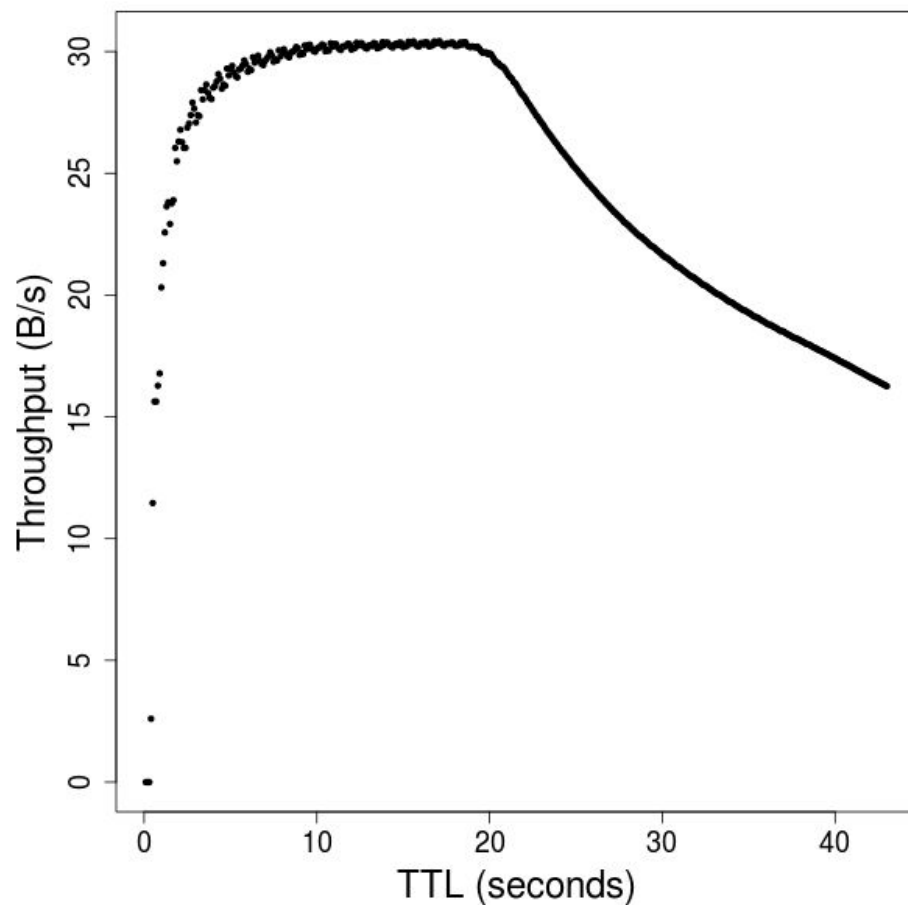
Testing And Results

- Non-reliable delivery performance slightly lower, 30.4 B/s at TTL of 17.1 s

Using previous latency data.



Throughput = (Frame Size × Loss) ÷ TTL



Conclusion



- Throughput too low for normal Serval traffic
- Can be used for simple messaging
- Significant potential if implementation issues are corrected
- Greater coordination could lead to higher throughput
 - Synchronize complementary states
 - Peers take turns issuing multiple service requests
 - Adaptive request intervals
- It can be used for configuration
 - Exchange data to connect through a different interface

Questions 

References



[1] Wong, P., Varikota, V., Nguyen, D. and Abukmail, A., 2014. Automatic android-based wireless mesh networks. *Informatica*, 38(4).

[2] Alliance, W.F., 2014. Wi-Fi Peer-to-Peer (P2P) Technical Specification, Version 1.5. Wi-Fi Alliance Technical Committee P2P Task Group.

[3] "The Serval Project". Servalproject.org. N.p., 2016. Web. 29 June 2016.