



# Peer to Peer Networks on Android

Paul O'Neil and Steven Presser

# Agenda

- Motivation
- Constraints
- Tools and Barriers
- Architecture
- Demonstration

# Agenda

- **Motivation**
- **Constraints**
- **Tools and Barriers**
- **Architecture**
- **Demonstration**

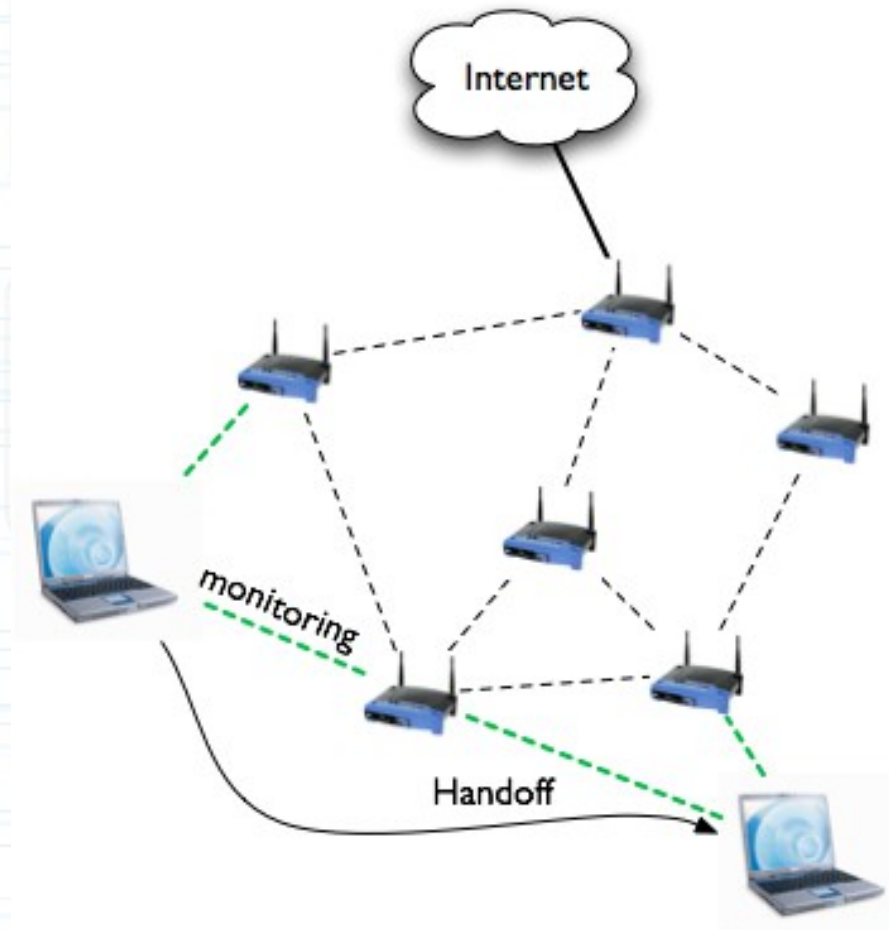
# Motivation

- Communication Critical in Disasters
- Infrastructure Networks likely damaged
- However, end users likely still have battery powered wireless devices
- Quickly deploy internet enabled devices in new areas
  - Only the edge needs an uplink

# Existing Solutions

- Peer to Peer network
- However, theoretical P2P already exists
  - Let's operate with unmodified end-user devices
- SMesh
  - Uses a system of mobile devices and fixed routers to create a mesh network
  - Requires rooting the phones
  - Overlays (via Spines) running on Android

# Existing Solutions



<http://www.smesh.org/>

# Agenda

- Motivation
- **Constraints**
- Tools and Barriers
- Architecture
- Demonstration

# Constraints

Cannot:

- Root
- Require other OS
- Require different radio
- Require any modification beyond installation of an application



# Agenda

- Motivation
- Constraints
- **Tools and Barriers**
- Architecture
- Demonstration

# Technology

- Android
  - Can have long-lasting outbound connections
  - Use the VPN interface to capture packets
  - Java
- WiFi Direct
  - Use 802.11-like protocols without an access point

# Android Apps

- Activity
  - One screen of an application
  - Control interface
- Service
  - Long lasting background worker
  - Networking
- Communicate using message passing
- The app *has* to be multithreaded
  - No network on the GUI thread

# Android

- iOS does not have the flexibility we need
  - Can't have the persistent outgoing connections
  - Can't capture packets via VPN
    - Apple implements the client as part of the operating system

# WiFi Direct

- Point-to-Point connectivity
- Targeted at:
  - Printers
  - Projectors
  - Transferring photos
  - Multiplayer Gaming?

# WiFi Direct

- There's no network name, so let's make one!
  - Group owner acts as an access point
- Pairing with another device means joining a group.
- Communication goes through the owner
- Everyone is told the address of the owner

# Barriers

- Android programming is conceptually challenging
- WiFi Direct functions poorly

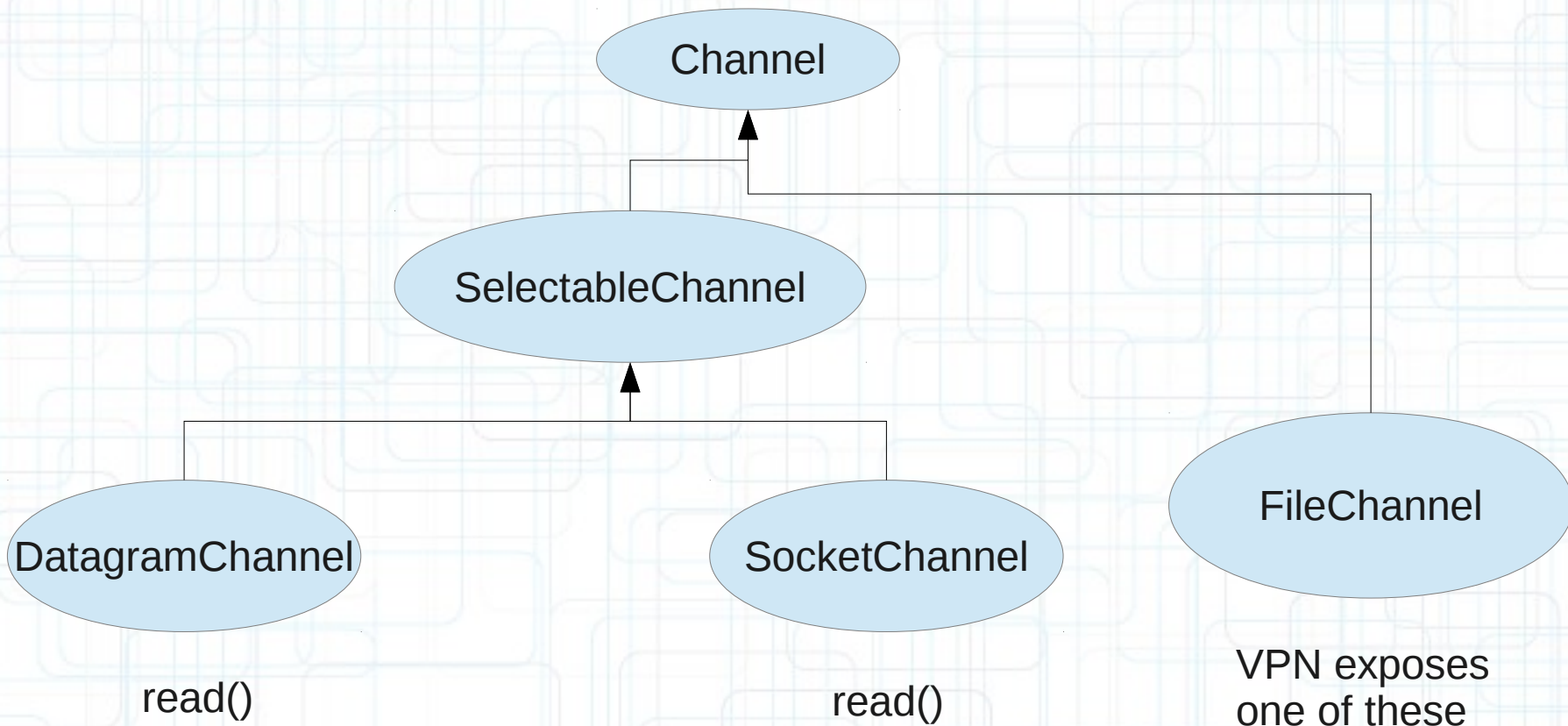
# Android Problems

- Everything is asynchronous and through callbacks
  - Requesting information about the current WiFi group
  - Spaghetti code
  - Maybe we're just not good enough at coding
- Some things can only happen on certain threads
  - Toast notifications, GUI access, Network



# Java is Not the Right Tool

- No unsigned types
  - Have to compute checksums



# WiFi Direct Authors would fail 600.437

The image displays two screenshots of an Android Wi-Fi Direct interface, illustrating a scenario where authors would fail to connect.

**Left Screenshot:**

- Device: Distributed\_8
- PEER DEVICES: Distributed\_5 (Available)
- REMEMBERED GROUPS: DIRECT-ku-Distributed\_8

**Right Screenshot:**

- Device: Distributed\_4
- PEER DEVICES: Distributed\_8 (Connected), Distributed\_5 (Available)
- REMEMBERED GROUPS: DIRECT-71-Distributed\_5, DIRECT-ku-Distributed\_8

# WiFi Direct Authors would fail 600.437

```
I/wpa_supplicant( 3714): p2p0: P2P-INVITATION-RECEIVED sa=12:68:3f:87:3a:c2 persistent=1
E/WifiP2pService( 510): Unhandled message { what=147487 when=0 obj=network: null
E/WifiP2pService( 510): isGO: false
E/WifiP2pService( 510): GO: Device:
E/WifiP2pService( 510): deviceAddress: 12:68:3f:87:3a:c2
E/WifiP2pService( 510): primary type: null
E/WifiP2pService( 510): secondary type: null
E/WifiP2pService( 510): wps: 0
E/WifiP2pService( 510): grpcapab: 0
E/WifiP2pService( 510): devcapab: 0
E/WifiP2pService( 510): status: 4
E/WifiP2pService( 510): wfdInfo: null
E/WifiP2pService( 510): Client: Device:
E/WifiP2pService( 510): deviceAddress: 12:68:3f:87:3a:c2
E/WifiP2pService( 510): primary type: null
E/WifiP2pService( 510): secondary type: null
E/WifiP2pService( 510): wps: 0
E/WifiP2pService( 510): grpcapab: 0
E/WifiP2pService( 510): devcapab: 0
E/WifiP2pService( 510): status: 4
E/WifiP2pService( 510): wfdInfo: null
E/WifiP2pService( 510): interface: null
E/WifiP2pService( 510): networkId: 1 }
```

# Things to Never See from a JVM...

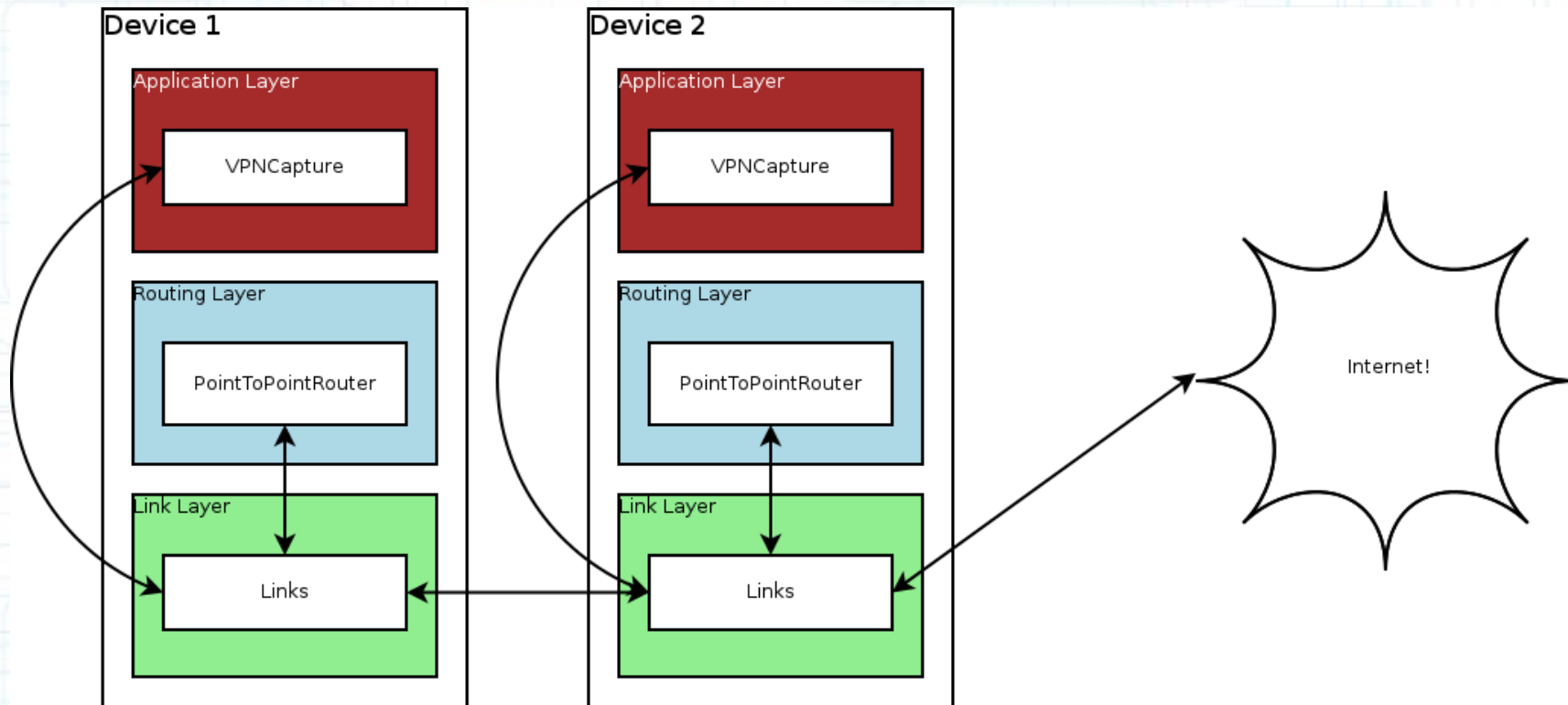
```
wifip2pnet System.err at edu.jhu.cnds.packets.TCPWrapper.handlePacket(TCPWrapper.java:66)
wifip2pnet System.err at edu.jhu.cnds.packets.TCPWrapper.handlePacket(TCPWrapper.java:44)
wifip2pnet System.err at edu.jhu.cnds.packets.IPv4Wrapper.handlePacket(IPv4Wrapper.java:98)
wifip2pnet System.err at edu.jhu.cnds.wifip2pnet.NATLink.send(NATLink.java:125)
wifip2pnet System.err at edu.jhu.cnds.wifip2pnet.NATLink.send(NATLink.java:97)
wifip2pnet System.err at edu.jhu.cnds.wifip2pnet.PointToPointRouter.send(PointToPointRouter.java:8
wifip2pnet System.err at edu.jhu.cnds.wifip2pnet.VPNLink.handleMessageFromVPN(VPNLink.java:78)
wifip2pnet System.err at edu.jhu.cnds.wifip2pnet.VPNCapture.run(VPNCapture.java:269)
wifip2pnet System.err at edu.jhu.cnds.wifip2pnet.VPNLink.run(VPNLink.java:55)
wifip2pnet System.err at java.lang.Thread.run(Thread.java:856)
wifip2pnet System.err Caused by: libcore.io.ErrnoException: sendto failed: EPIPE (Broken pipe)
wifip2pnet System.err at libcore.io.Posix.sendtoBytes(Native Method)
wifip2pnet System.err at libcore.io.Posix.sendto(Posix.java:141)
wifip2pnet System.err at libcore.io.BlockGuardOs.sendto(BlockGuardOs.java:169)
wifip2pnet System.err at libcore.io.IoBridge.sendto(IoBridge.java:487)
wifip2pnet System.err ... 12 more
wifip2pnet libc Fatal signal 11 (SIGSEGV) at 0x00006c68 (code=1), thread 1574 (Thread-153)
```

Launching WifiP2PNet

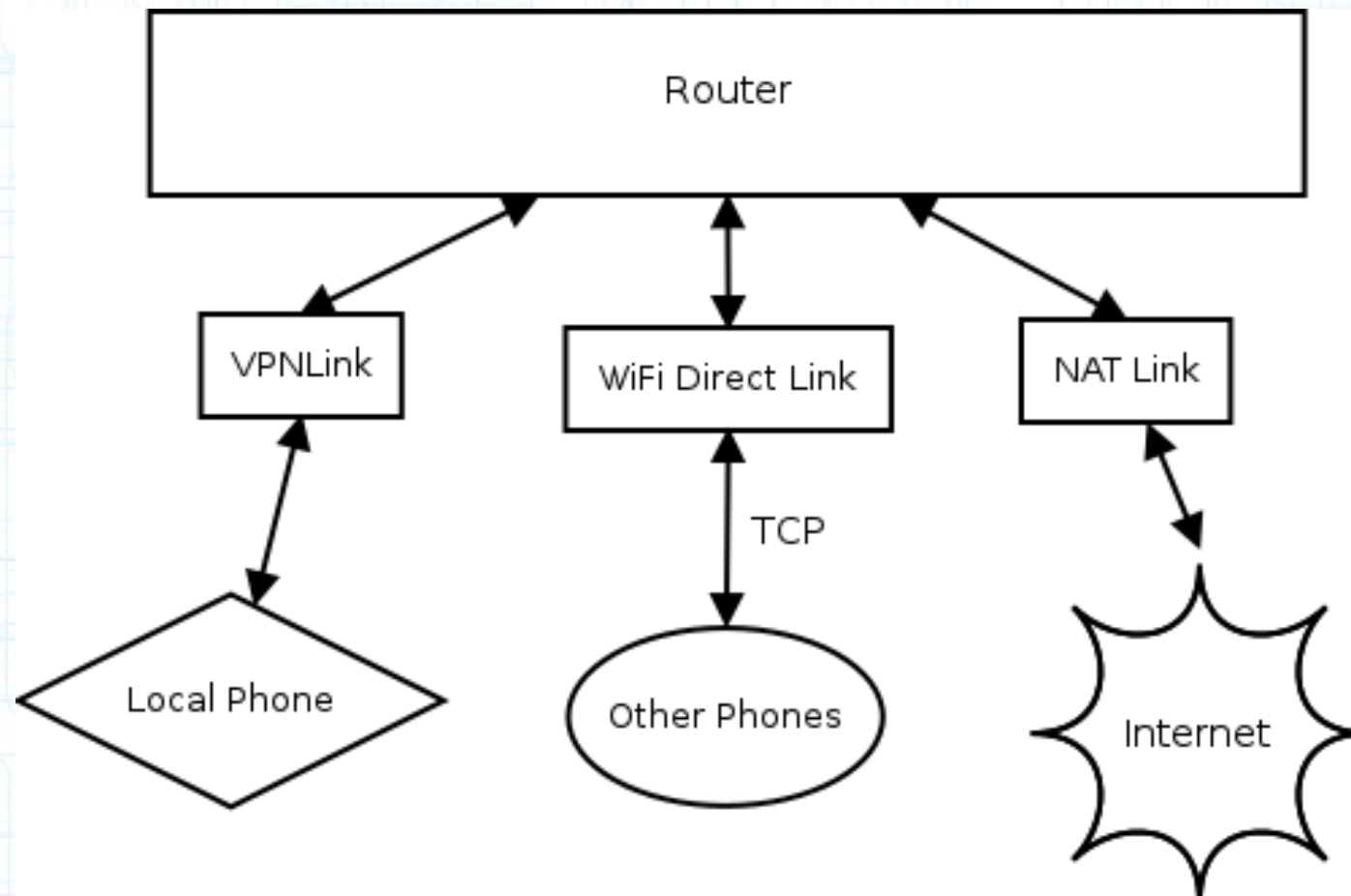
# Agenda

- Motivation
- Restrictions
- Tools and Barriers
- **Architecture**
- Demonstration

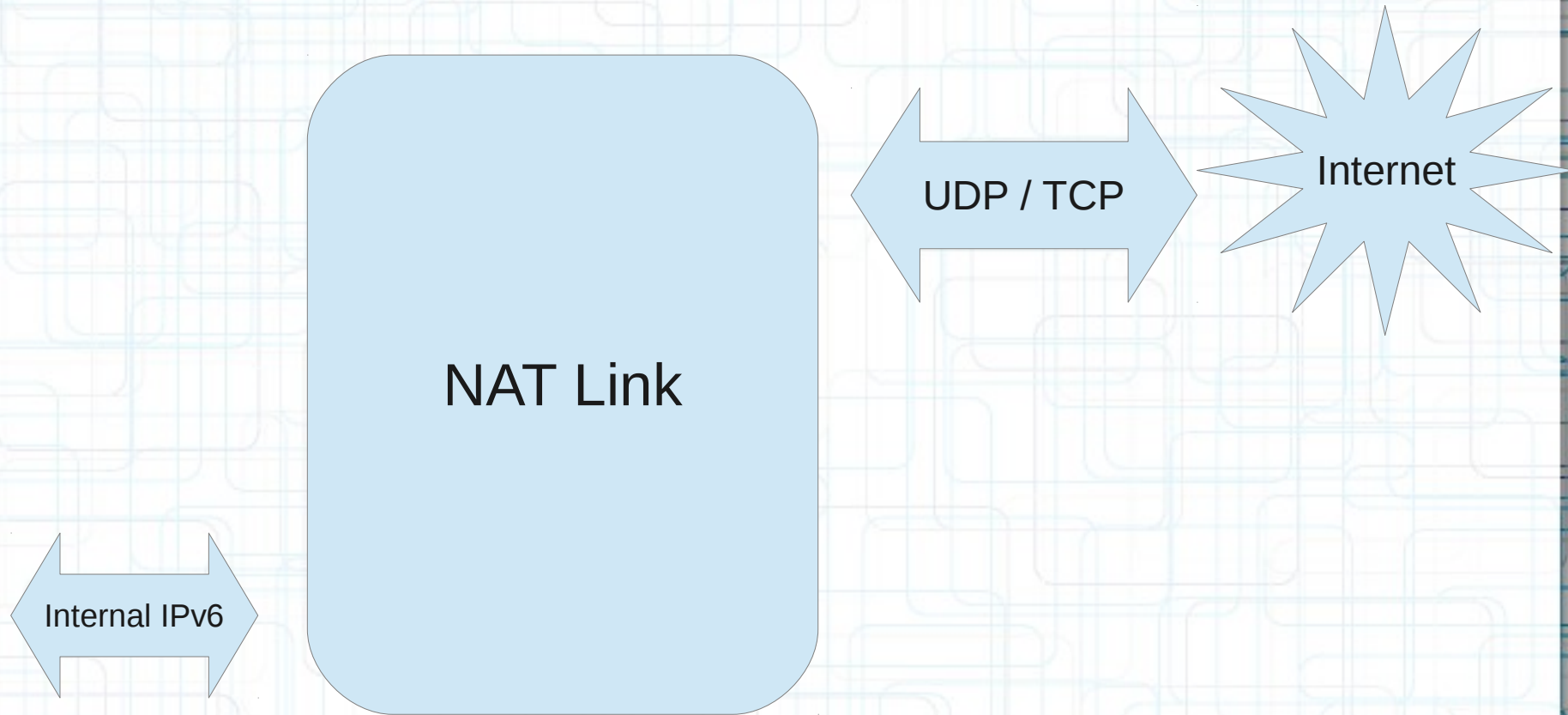
# Architecture



# Links



# NAT





# Agenda

- Motivation
- Restrictions
- Tools and Barriers
- Architecture
- **Demonstration**

# Demonstration

- Very Slow Browser
- Text Messaging

**Questions?**