Off Grid communications with Android

- Meshing the mobile world

m0nk and stoker have fun @ DefCon 20

Who are you guys?

- m0nk – Josh Thomas
  - jthomas@accuvant.com
  - m0nk.omg.pwnies@gmail.com
- Stoker – Jeff Robble
  - jrobble@mitre.org
  - mistr.stoker@gmail.com
- We work(ed) @ The MITRE Corporation (of CVE fame)

tl; dr:

- https://github.com/monk-dot

A placeholder so m0nk can babble

Where data goes to die

- Fukushima
- Katrina
- Haiti
- < Insert your “favorite” recent natural disaster here >
- Other?

Why do I care about Mesh networks?

- Physical infrastructure is prone to failure, networks shouldn't be
- Bypass the Cellular networks
- Bypass Wi-Fi networks
- Share information when infrastructure is broken or untrustworthy
- Extend and bounce other networks via bridging / tethering
- Headless

Ok, kind of cool. What about "Off Grid"?

- Single point of failure = single point of sniffing / filtering
- I don't trust someone else being able to turn off my network, do you?
- When you want to share info, but don't want anyone watching ☺

There should really be a funny pic below

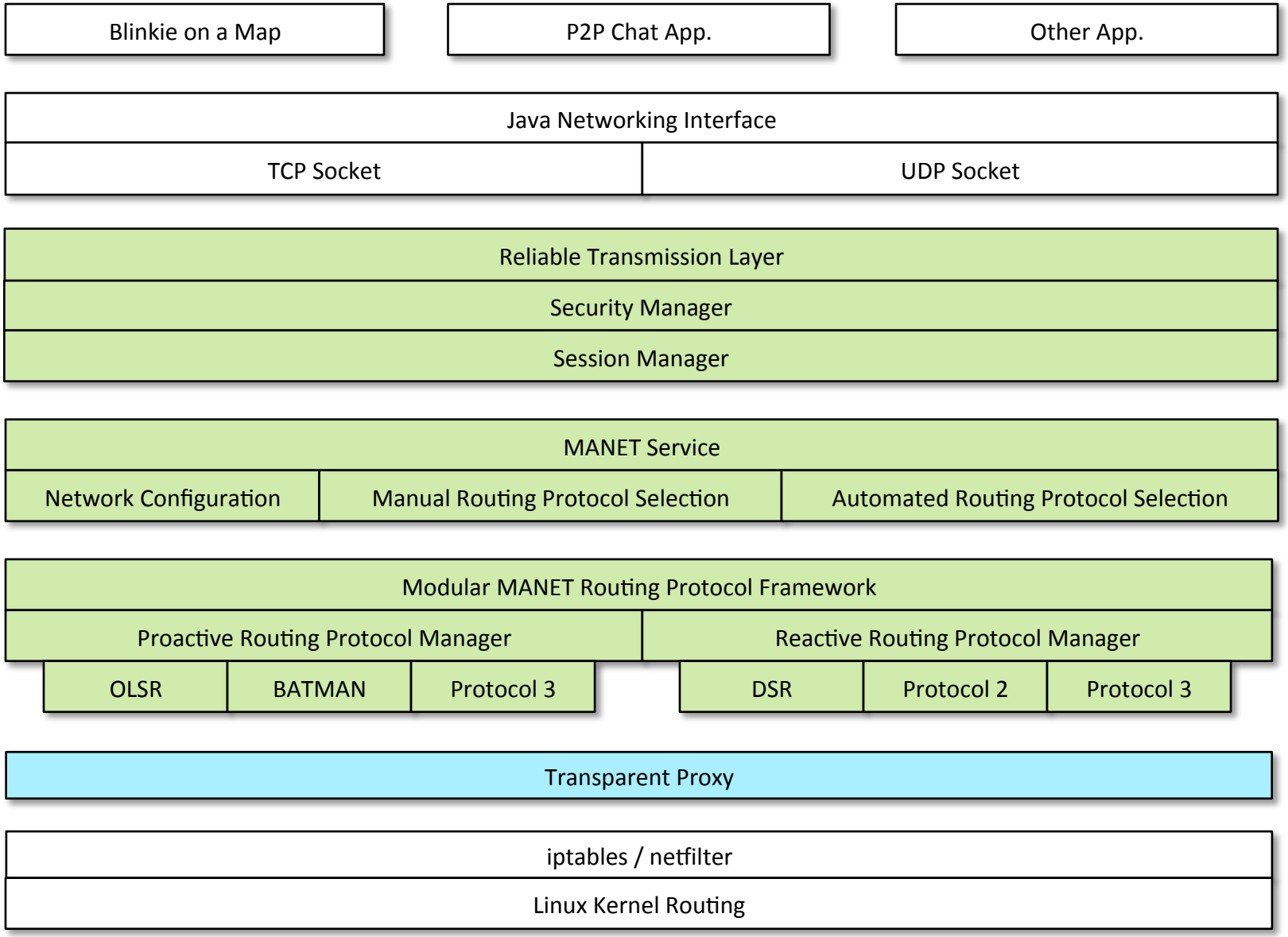
Your pocket contains more than a consumption device for Grumpy Fowl

- Wi-Fi chip with a fairly fat pipe
- Cell modem and baseband processor
- A ton of sensors
- (Somewhat) quality NAND and RAM
- A very under clocked and underutilized processor
- Power
- A boring screen that blinks!

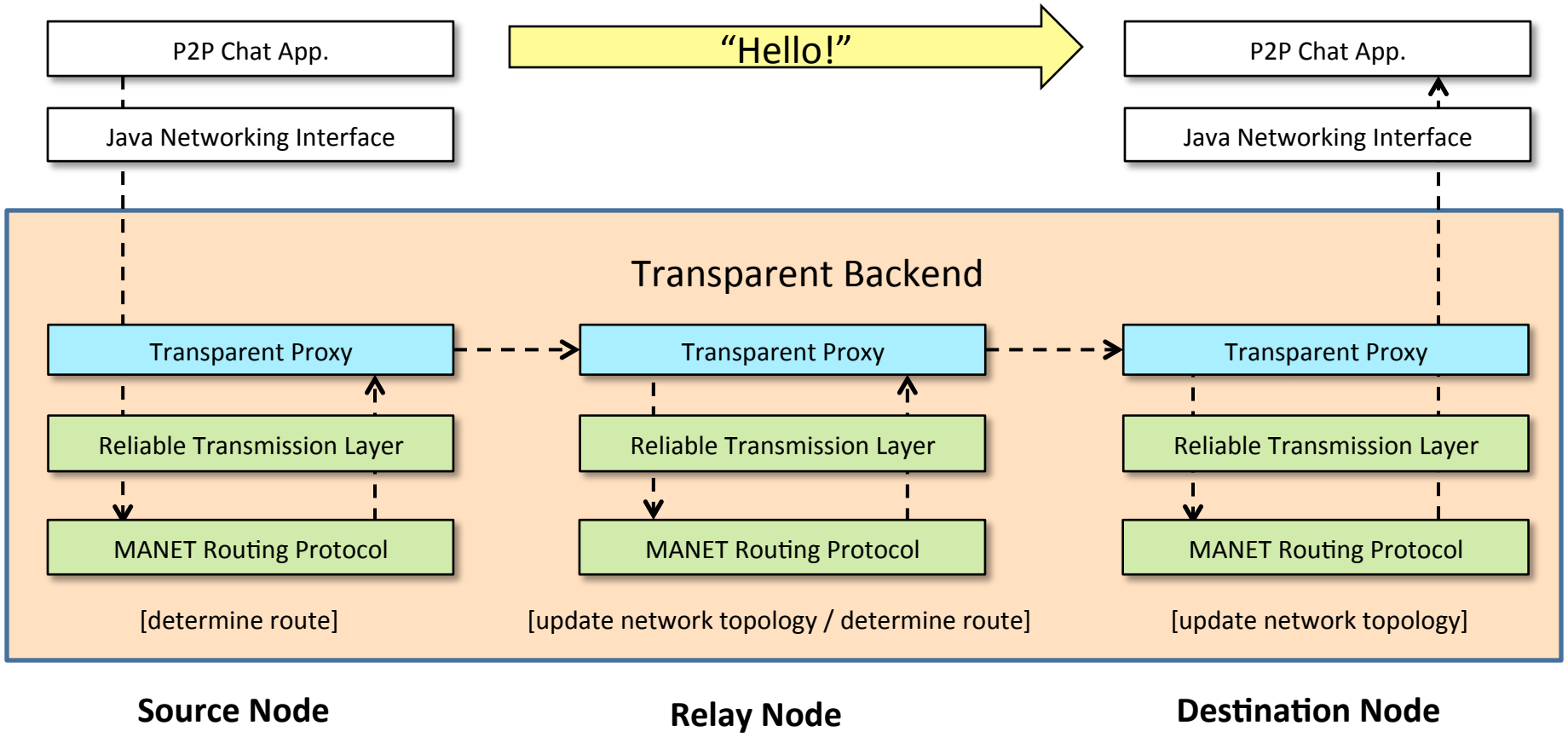
The SPAN framework

- We did the boring stuff so you don't have to!
- General Overview of the framework, what / why / how
  - Harnessing SPAN for your own project?
  - Repurpose root to muck with your Wi-Fi chipset

SPAN + Android Technical Architecture



Data Flow


P2P Chat App.
"Hello!"
P2P Chat App.
Java Networking Interface
Java Networking Interface
Transparent Backend
Transparent Proxy
Transparent Proxy
Transparent Proxy
Reliable Transmission Layer
Reliable Transmission Layer
Reliable Transmission Layer
MANET Routing Protocol
MANET Routing Protocol
MANET Routing Protocol
[determine route]
[update network topology / determine route]
[update network topology]
Source Node
Relay Node
Destination Node

Why we love Broadcom

- Flipping chipsets into Ad-Hoc Mode

Device	**Wireless Chip**
Samsung Nexus S 4G	Broadcom BCM4329
Samsung Galaxy Tab 10.1	Broadcom BCM4330
Samsung Galaxy S II Epic Touch 4G	Broadcom BCM4330
Samsung Galaxy Nexus	Broadcom BCM4329
ASUS Eee Pad Transformer Prime	AzureWave AW-NH615 (rebranded Broadcom BCM4329)
Motorola Razr Maxx	Texas Instruments WL1285C
iPhone 4S	Broadcom BCM4330
Nokia Lumia 900	Broadcom BCM4329

Kernel v. Metal

Wireless Extensions Support	No Wireless Extensions Support
Samsung Nexus S 4G	Samsung Galaxy Nexus
Samsung Galaxy Tab 10.1	ASUS Eee Pad Transformer Prime
Samsung Galaxy S II Epic Touch 4G	Motorola Razr Maxx

- Dear Vendors: Please either stop mucking with your kernel source or provide it to the community

Ad-hoc Mode

- Leveraged Wi-Fi Tether for Root Users app.
  - Edify script for setting up ad-hoc mode using cross-compiled iwconfig
- Some phone wi-fi drivers don't support ad-hoc mode
  - Wi-Fi Tether app. switched to using softAP
  - softAP: software enabled portable wireless access point
- Needed to compile Wireless Extensions support into kernel
  - Compiled vendor open source software
  - Dumped zImage and drivers to AnyKernel tree
  - Flashed using ClockworkMod Recovery

https://github.com/koush/AnyKernel

Why we love Broadcom

- Flipping chipsets into Ad-Hoc Mode

Device	**Wireless Chip**
Samsung Nexus S 4G	Broadcom BCM4329
Samsung Galaxy Tab 10.1	Broadcom BCM4330
Samsung Galaxy S II Epic Touch 4G	Broadcom BCM4330
Samsung Galaxy Nexus	Broadcom BCM4329
ASUS Eee Pad Transformer Prime	AzureWave AW-NH615 (rebranded Broadcom BCM4329)
Motorola Razr Maxx	Texas Instruments WL1285C
iPhone 4S	Broadcom BCM4330
Nokia Lumia 900	Broadcom BCM4329

Kernel v. Metal

Wireless Extensions Support	No Wireless Extensions Support
Samsung Nexus S 4G	Samsung Galaxy Nexus
Samsung Galaxy Tab 10.1	ASUS Eee Pad Transformer Prime
Samsung Galaxy S II Epic Touch 4G	Motorola Razr Maxx

- Dear Vendors: Please either stop mucking with your kernel source or provide it to the community.

Where are my packets?

- Android <= 4.0 (ICS) devices filter out UDP broadcasts when the screen is off
  - WifiManager.WifiLock doesn't help
- First approach: Force screen to always stay dimmed even when user presses power button
  - Create wakelock
    - powerManager.newWakeLock(PowerManager.SCREEN_DIM_WAKE_LOCK | PowerManager.ACQUIRE_CAUSES_WAKEUP, "ADHOC_WAKE_LOCK")
  - Register an IntentFilter for Intent.ACTION_SCREEN_OFF
  - Acquire wakelock when intent received

Where are my packets?

- Second approach: Set dhd_pkt_filter_enabled=0 when loading wi-fi kernel module
  - Required recompiling Galaxy Nexus wi-fi driver

Plug and Play / Dynamic routing algorithms and you!

- Adjusting packet routing at runtime, a 5 minute primer on untrustworthy routing tables
- The tradeoffs of Bandwidth vs. Network Scale and Multi-Hop headaches
- File share, Chat, Disconnected Twitter and VOIP over a Mesh. Oh, the fun we can have.

This slide should not be needed

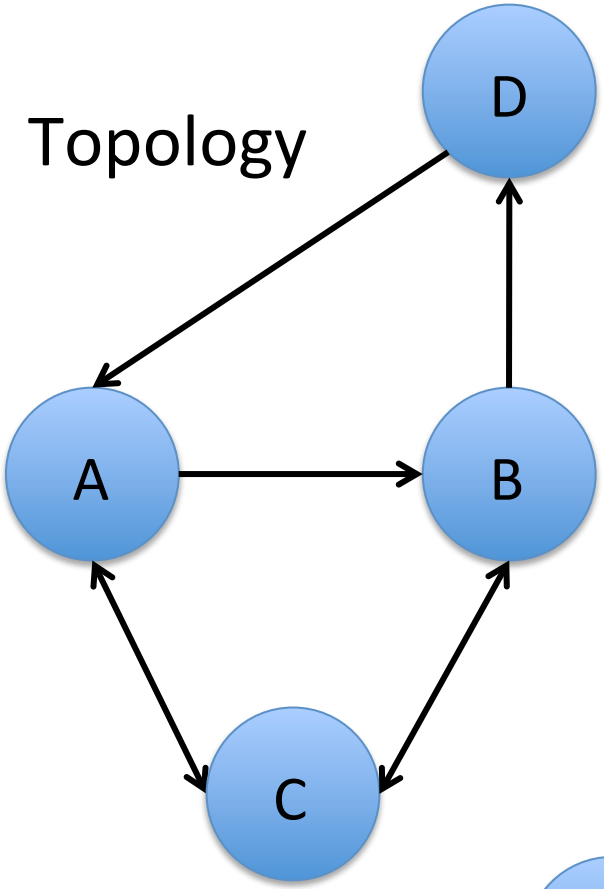
- What do I use a network for?
  - Chat
  - Data and file sharing
  - VoIP
  - Situational Awareness and Crisis management
  - Disconnected Twitter

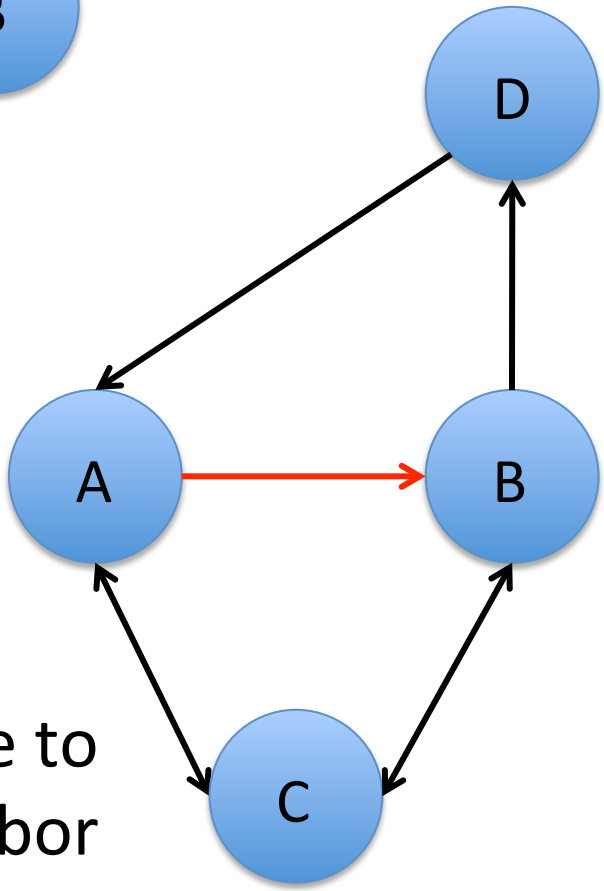
Ad-Hoc Network Routing 101

- Why BATMAN is better than OLSR?

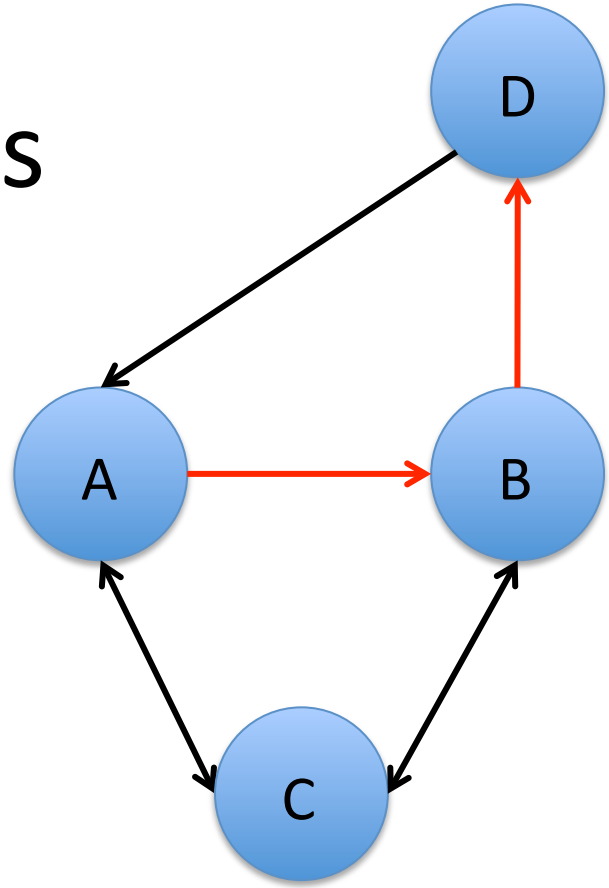
Definitions

Topology

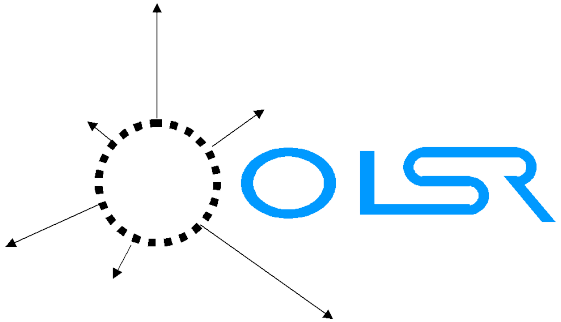




Route to
1-hop neighbor



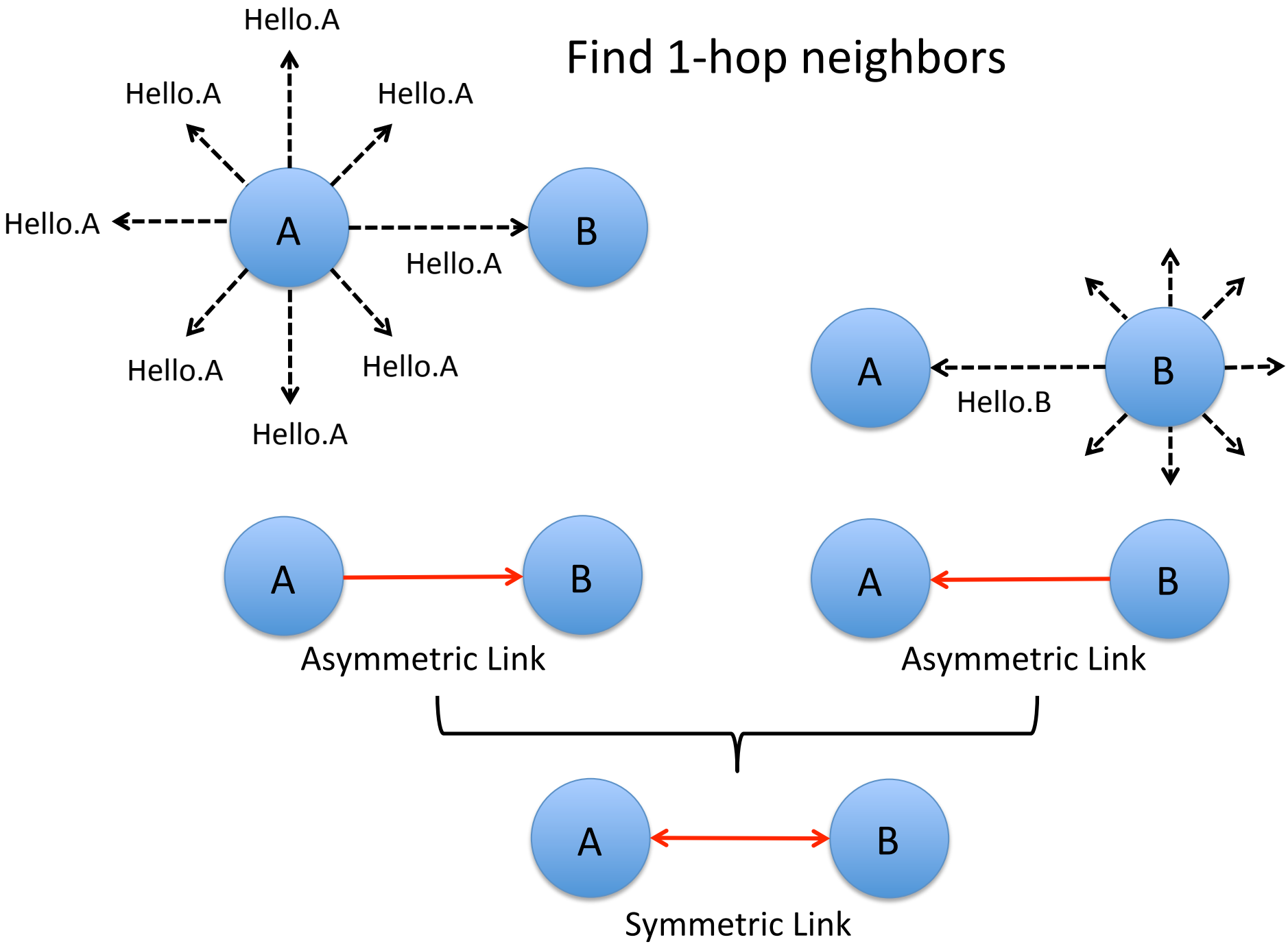
Route to 2-hop
neighbor



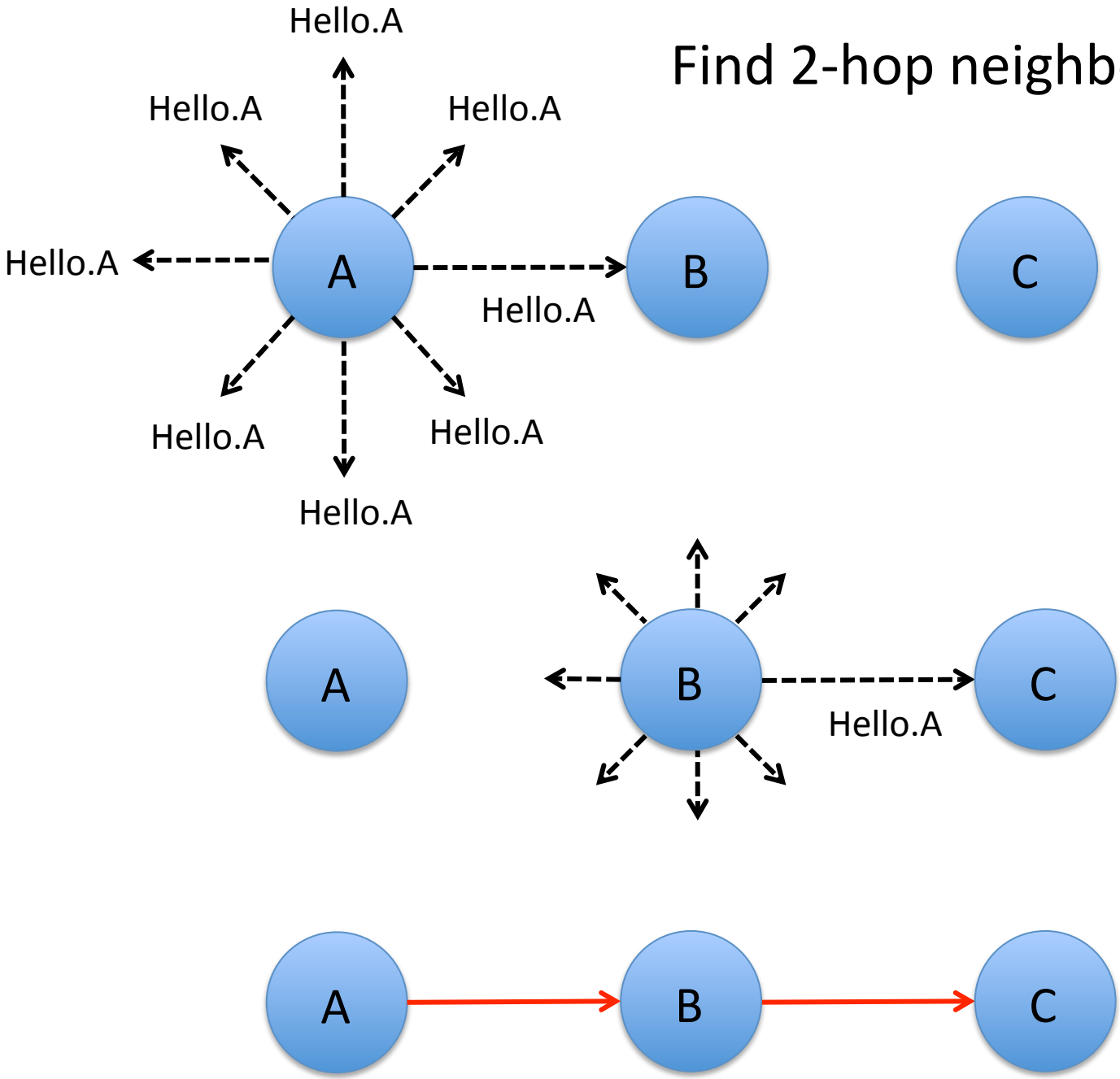
- Optimized Link State Routing Protocol (2003)
- Link-state protocol
  - Nodes know who they can talk to
  - Each node calcs entire route to every other node
- Proactive
  - Routes periodically planned in advance
  - Kernel-level routing table modified on-the-fly
- Dijkstra Open Shortest Path First algorithm
- Layer 3 in OSI stack

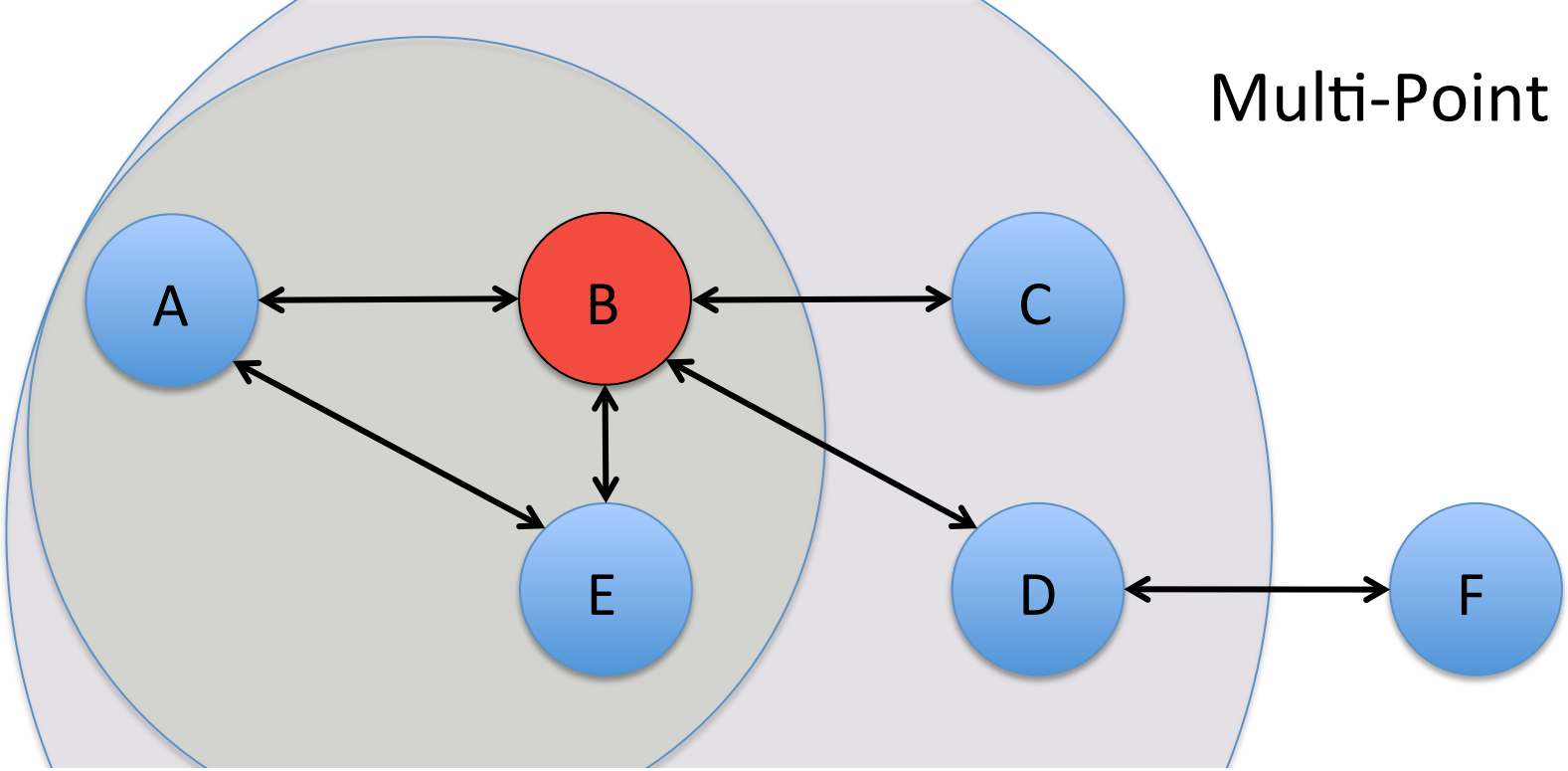
http://www.ietf.org/rfc/rfc3626.txt

Find 1-hop neighbors



Find 2-hop neighbors





- A selects B as MPR
  - All 2-hop nodes reachable through B
- All > 1-hop routes from A will go through B

OLSR

- Pros
  - Better than everyone sharing everything
    - Topology info dumps only between MPRs
  - Incremental improvements
- Cons
  - MPRs are throughput choke points
    - Isolated points of failure
  - Entire routes planned in advance, but next hop doesn't care about your route, it uses its own

BATMAN

- Better Approach to Mobile Ad-hoc Networking (2006)
- Next-gen OLSR
- Decentralize: No single point has all the data
  - No MPRs
  - Each node sends out originator msgs: “I exist”
  - Every other node keeps track of number of hops an originator msg took to reach them

http://www.open-mesh.org/projects/open-mesh/wiki/The-olsr-story

BATMAN

- Simplify: Only plan first step in route
  - Direct packets along route with lowest originator msg. hop count

Where Are We Today?

- OLSR still the most popular
- BATMAN gaining traction
- We can do better and so can you
  - If you are working in the space, please email us.

Smart Phones Have Sensors!

- Battery
  - Don't send packets to phones going dead
  - Send more packets to phones plugged in
- GPS
  - Form routes to phones closer to you
  - Form routes to phones that don't move often
- Accelerometer
  - Don't send packets to phones in motion
  - Predict phone movement and send packets to phones moving in the right direction

Reactive Protocols

- Stale routing table = What routing table?
- No we can play with motion and location in a useful way
- Don’t forget that if you pack node location into the headers it can been seen by others
- Downsides come with throughput issues

An aside on Delay tolerance

- Disconnected nodes act as disjoint message queues
- The protocol thinks of the device as a carrier pigeon (RFC 2549)
- Fall back to message passing

Scale, Delay and Hopping

- Though we see great improvements, simple proactive routing uses a ton of bandwidth to stabilize the network
  - Still, we can predict bandwidth and throughput metrics
  - VoiP good until we scale quite large
- Reactive routing has less chatter with the same bandwidth but is laggy
- Mix them FTW.

More Tunnels and some preliminary Security

- Jumping over the cell network or Wi-Fi (Mimicking VPN with standard Tunnels)
- Tunneling the mesh through the Internets!
  - VPN clusters and remote enclaves
- Securing the mesh from unwanted guests
- Jumping through unsecured mobile nodes

Jumping over the cell network or Wi-Fi

- Your phone has at least 2 network ports (Wi-Fi & Cell):
  - We can connect them
  - We can bridge them
- Tablet with no cell chip?
  - Plug in an ALFA wireless USB dongle
- Virtual mesh networks connected using simple VPN tunnels

IP Address Assignment

- Static IP assignment
- Generate a unique IP based on phone MAC address, IMEI, etc.
- DHCP requires a server or global knowledge of IPs in use

A Security Paradigm?

- Use Bluetooth or NFC to Bump transfer configuration info and keys
- Secure each link / node with its own keys
- Encrypt network data such that bounce or hop nodes cannot decrypt

Security

- Share symmetric key in config file distributed in-person via NFC
- Symmetric encryption using P2P Diffie-Helman key exchanges
- Asymmetric encryption using public / private key pair
- A third party certificate authority isn't practical

Security

- Serval public keys double as network addresses
- 256-bit Curve25519 public keys based on the CryptoBox NaCl crypto library
- Network intrinsically distributes keys!
- Uses CryptoBox authenticated encryption for unicast traffic
- Uses CryptoSign verified signing for publicly readable broadcast traffic
- CryptoSign uses a handwritten sign to confirm identity

http://servalpaul.blogspot.com/2012/04/making-security-simple.html

ICS & Wi-Fi Direct: android.net.wifi.p2p API

- "Provides classes to create peer-to-peer (P2P) connections over Wi-Fi Direct"
- Initial ICS drop is a very lame partial implementation of the spec
  - Kind of works like Bluetooth pairing
  - Wi-Fi doesn't support connecting to an AP and P2P at the same time
- Possible upgrade in JB?

Root required

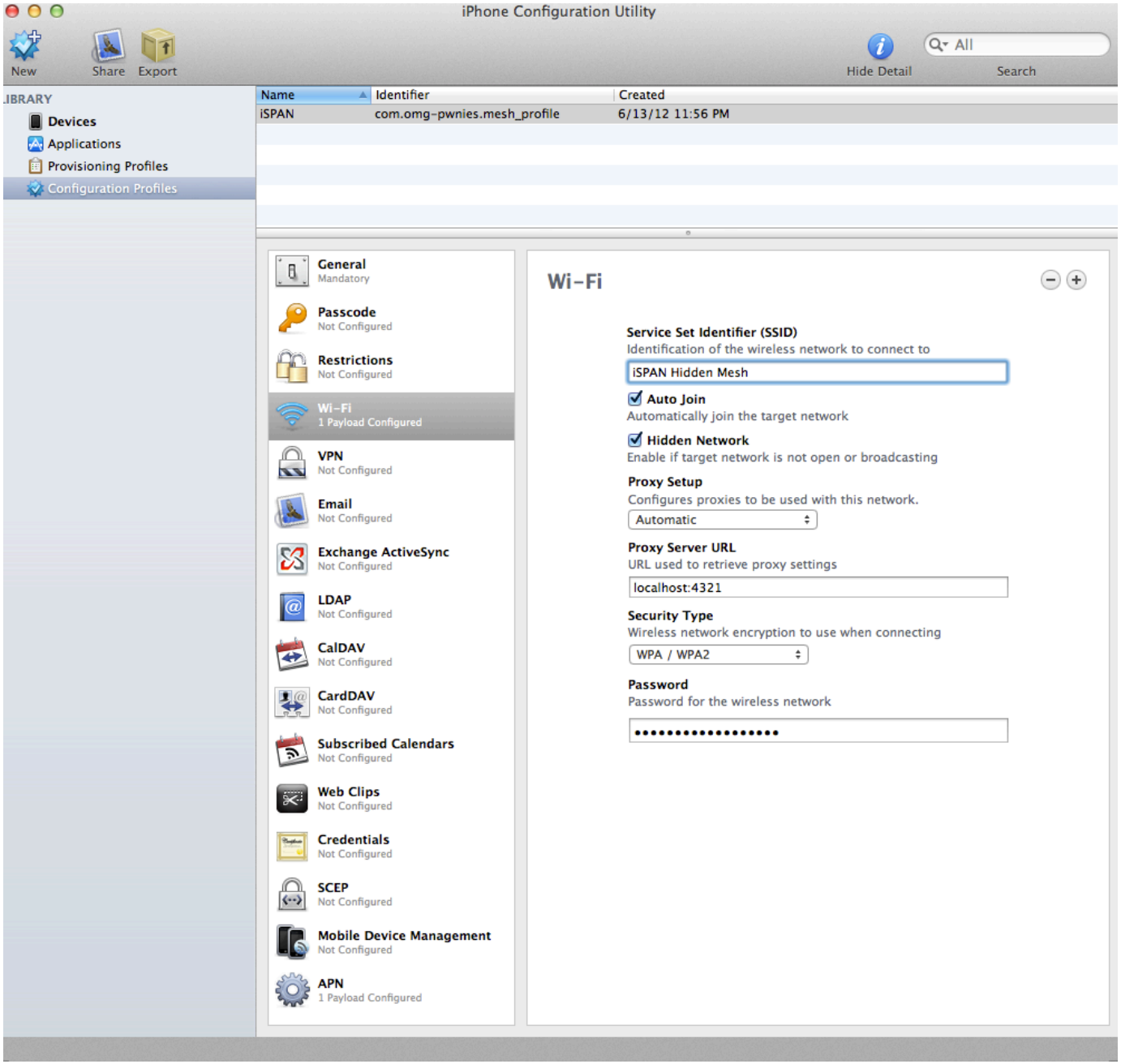
- Need root to modify iptables / routing tables
- Need root to mess with Wi-Fi driver and put phone in ad-hoc mode
- Grab Zerg, wrap in APK and pop the phone on install
- Over the Air install?

What about my…?

- A:
  - iPhone: In Theory
  - Black Berry: Maybe?
  - Windows Phone: Yes (why do you own one?)
  - Arduino / GumStix: Yes
  - Netbook / Linux / Mac / Windows Box: Yes
  - Toaster: Yes but Why?
- Framework is a mix of Java and C
  - If your box can run those…

iOS?

- Apple gave us a built in Wi-Fi proxy configurable with the iPhone Configuration Utility
- Ooohhh, is that an APN setting as well?
- Cool, now all we need is a simple server to proxy and route our data


iPhone Configuration Utility
New
Share
Export
Hide Detail
All
Search
LIBRARY
Devices
Applications
Provisioning Profiles
Configuration Profiles
Name
Identifier
Created
iSPAN
com.omg-pwnies.mesh_profile
6/13/12 11:56 PM
General
Mandatory
Passcode
Not Configured
Restrictions
Not Configured
Wi-Fi
1 Payload Configured
VPN
Not Configured
Email
Not Configured
Exchange ActiveSync
Not Configured
LDAP
Not Configured
CalDAV
Not Configured
CardDAV
Not Configured
Subscribed Calendars
Not Configured
Web Clips
Not Configured
Credentials
Not Configured
SCEP
Not Configured
Mobile Device Management
Not Configured
APN
1 Payload Configured
Wi-Fi
Service Set Identifier (SSID)
Identification of the wireless network to connect to
iSPAN Hidden Mesh
Auto Join
Automatically join the target network
Hidden Network
Enable if target network is not open or broadcasting
Proxy Setup
Configures proxies to be used with this network.
Automatic
Proxy Server URL
URL used to retrieve proxy settings
localhost:4321
Security Type
Wireless network encryption to use when connecting
WPA / WPA2
Password
Password for the wireless network
••••••••••••••••••

What else can we use the Mesh for?

- Mobile data redundancy using the Torrent protocol to raid data across all devices?
- Distribute threads and tasks across a cloud of unused processors?
- Spoofing?

Similar Projects

- Collaboration?

Freifunk

- German for "Free radio”
- Non-commercial open grassroots initiative to support free open radio networks in Germany
- Offers specialized OpenWrt-firmware
  - Routing based on OLSR or BATMAN
- Freifunk Berlin has 500+ nodes

http://berlin.freifunk.net/ http://wiki.freifunk.net/

- Android ad-hoc network framework

- Implemented features
  - VOIP calls between Serval Mesh-enabled phones
  - MeshMS, free mesh-based SMS

- Features under development
  - Serval Rhizome, distributed mesh-based data distribution platform
  - Serval Maps, mesh-based mapping application
  - Serval Morse, distributed micro-blogging service
  - A simple API for using Serval services

http://www.servalproject.org/

Future Work

- VOIP over the mesh
- IP address assignment
- Evaluate and improve Serval's approach to security
- iOS and Windows 8 port

Dumb enough to attempt a demo!

- Oh wait, we already did?

Shameless Plug

- GitHub repo:
  - https://github.com/monk-dot

Open Source Projects Used

- Wireless Tether for Root Users
  - "This program enables tethering (via wifi) for rooted handsets."
  - http://code.google.com/p/android-wifi-tether/
- olsrd
  - "An adhoc wireless mesh routing daemon"
  - http://www.olsr.org/
- monoutil
  - "A simple tool for network monitoring" using netfilter
  - http://code.google.com/p/monoutil/
- Processing for Android
  - "Processing is a language and environment for people who want to create images, animations, and interactions."
  - http://wiki.processing.org/w/Android
- Linux: iwconfig, iptables, dnsmasq, tcpdump, etc.