

# Project Byzantium

Networking for the Zombie Apocalypse

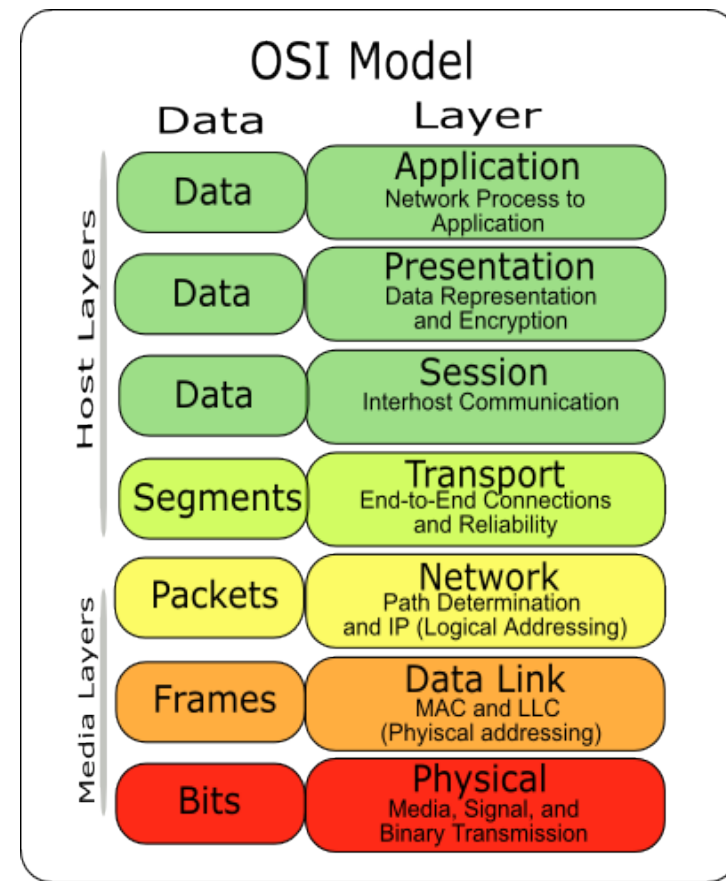
# The Internet is BROKEN.



It fails on many levels, but let's just focus on the lowest level ones for now.

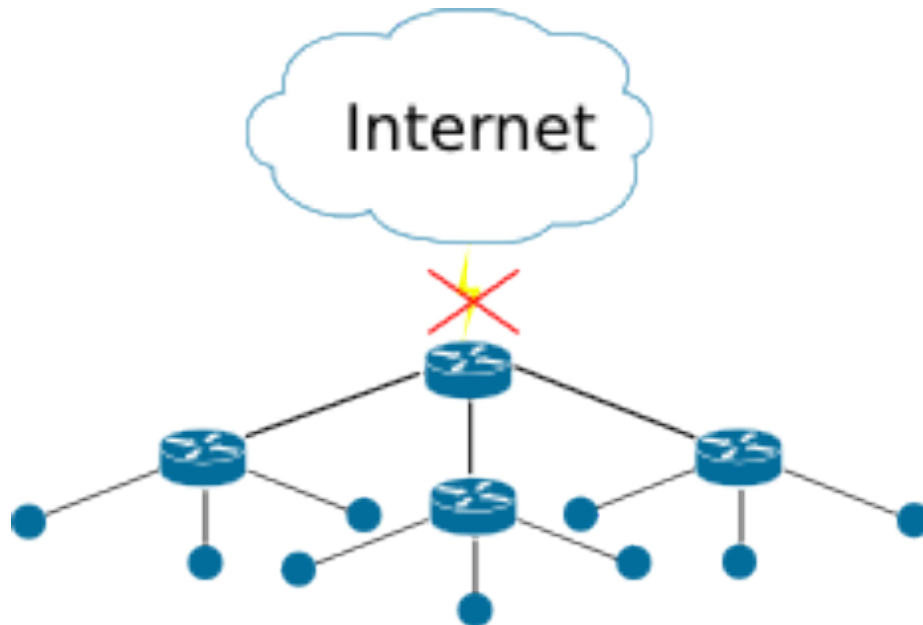
# Assumptions

- You know what the Internet is
- You're familiar with the OSI model
- You know what routing does (layer 3)
- You know how to use 802.11 (layers 1 & 2)
- You like being connected



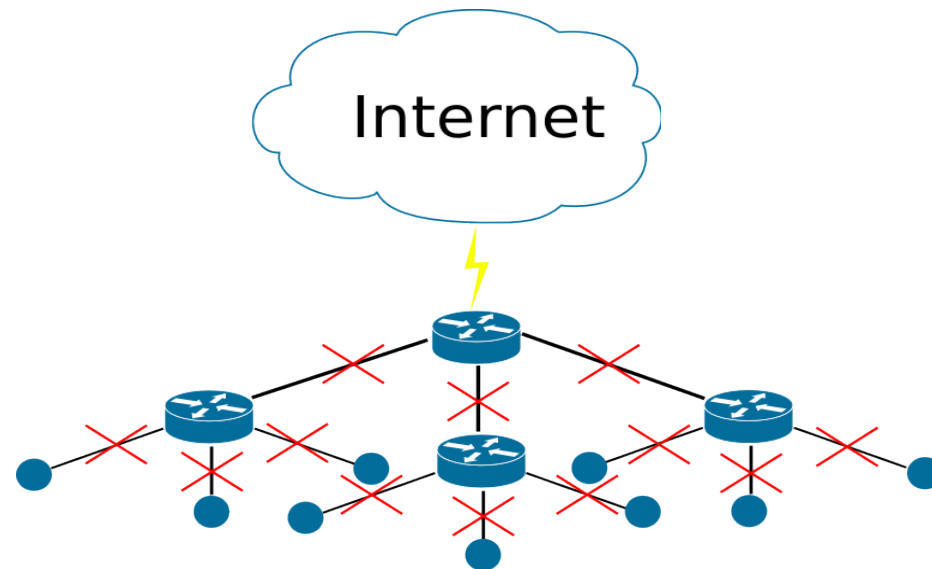
# Use Case: Egypt Problem

- Deliberate compromise of infrastructure
- ISPs offline
- Trunk lines go dark
- Services are filtered
- Deep packet inspection used to selectively block traffic
- Need to collaborate

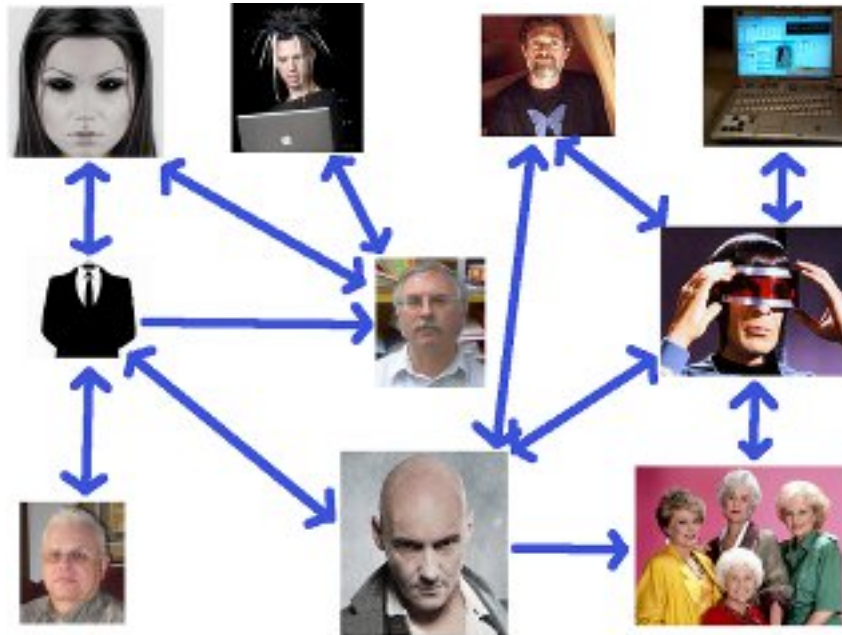


# Use Case: Katrina Problem

- Massive infrastructure failure
- Natural disaster
- Connectivity is patchy at best, likely unavailable
- What still works barely works
- Need to communicate



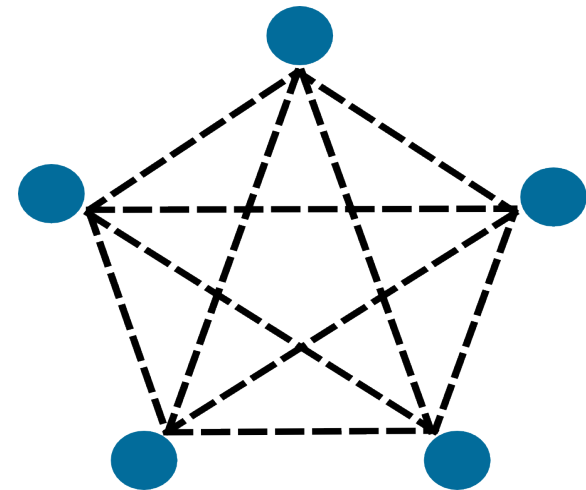
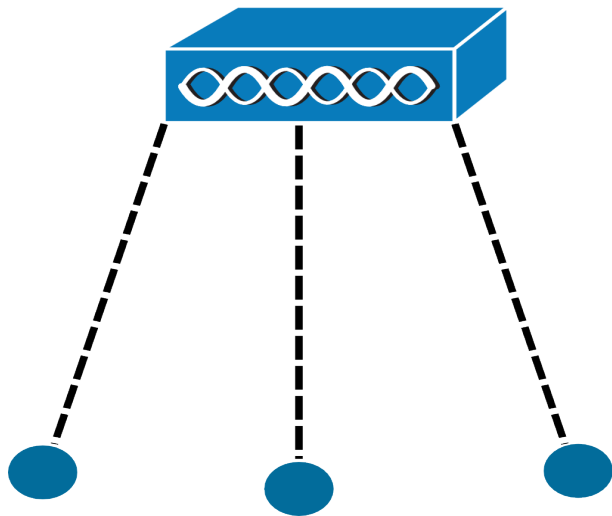
# Our Approach



- Mobile, ad-hoc mesh network
- All commodity equipment - you already have it
- Not all nodes need to run mesh routing software
- Provides services to associated clients
- Flexible enough to allow for improvisation

# Ad-Hoc Networking

- Takes place at OSI layers 1 and 2
- Built into 802.11
- Requires minimal configuration to bootstrap a network
- Nodes can move around physically while maintaining connectivity
- Nodes can relay traffic to extend range
- Any wi-fi enabled device can do it

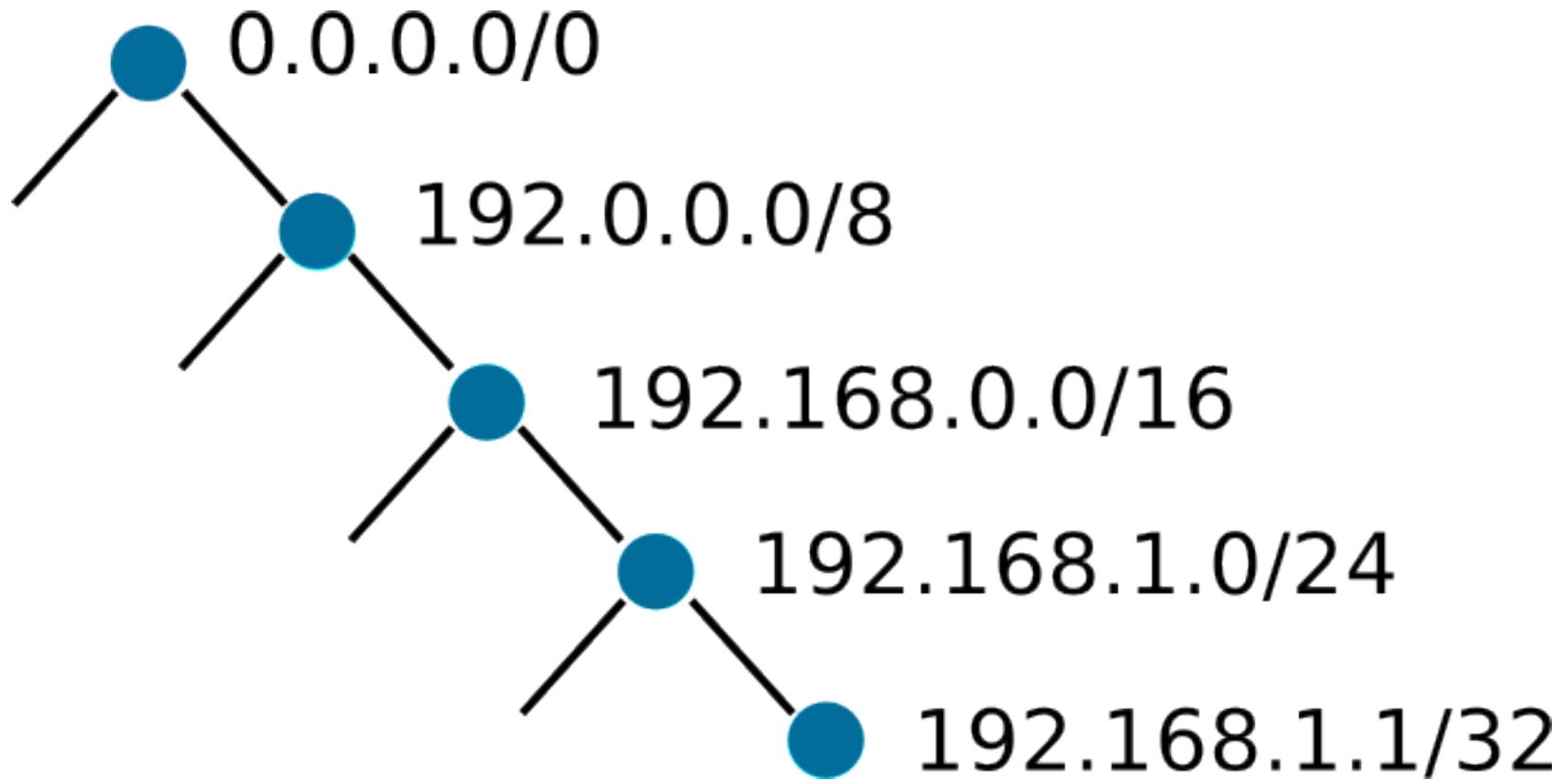


# Mesh Routing

- Takes place at OSI layer 3
- Nodes perform true IP routing
- A number of protocols exist
  - By 'a number' I mean around 70
  - [https://secure.wikimedia.org/wikipedia/en/wiki/List\\_of\\_ad\\_hoc\\_routing\\_protocols](https://secure.wikimedia.org/wikipedia/en/wiki/List_of_ad_hoc_routing_protocols)
- Not all protocols have the same features, solve the same problems, or are efficient



# How Routing Works



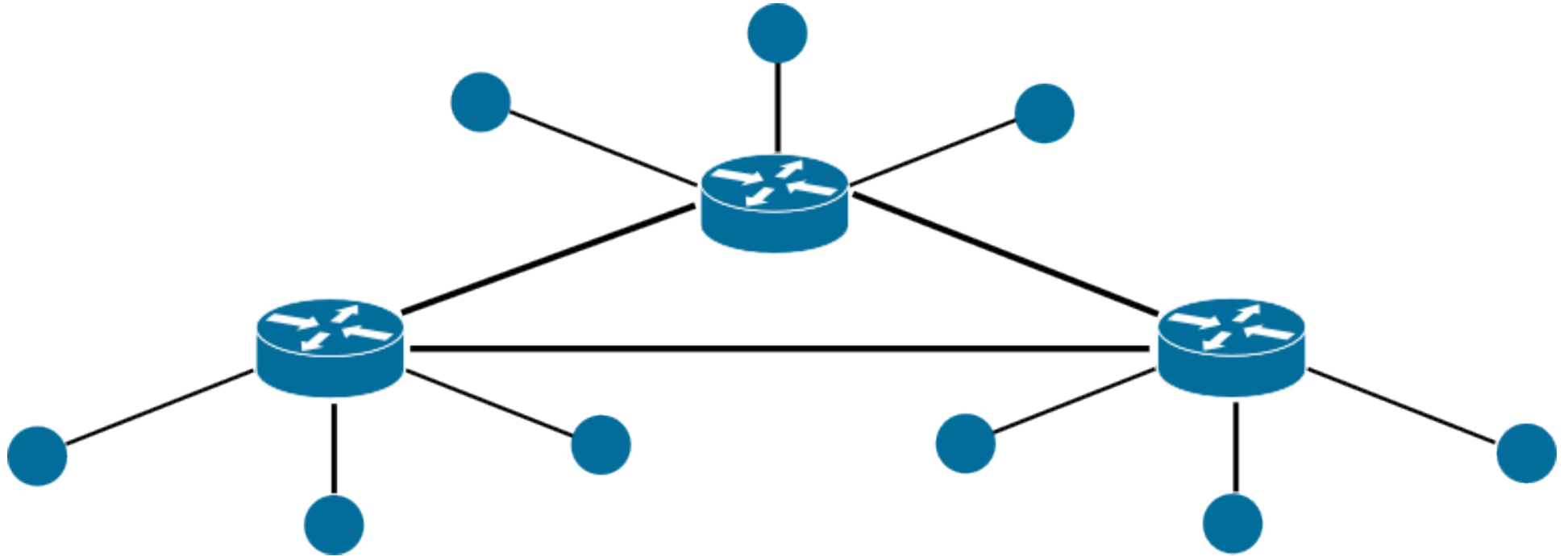
# But wait! Isn't the Internet a decentralized network?



Doesn't the Internet interpret censorship as damage and route around it?

Not really.



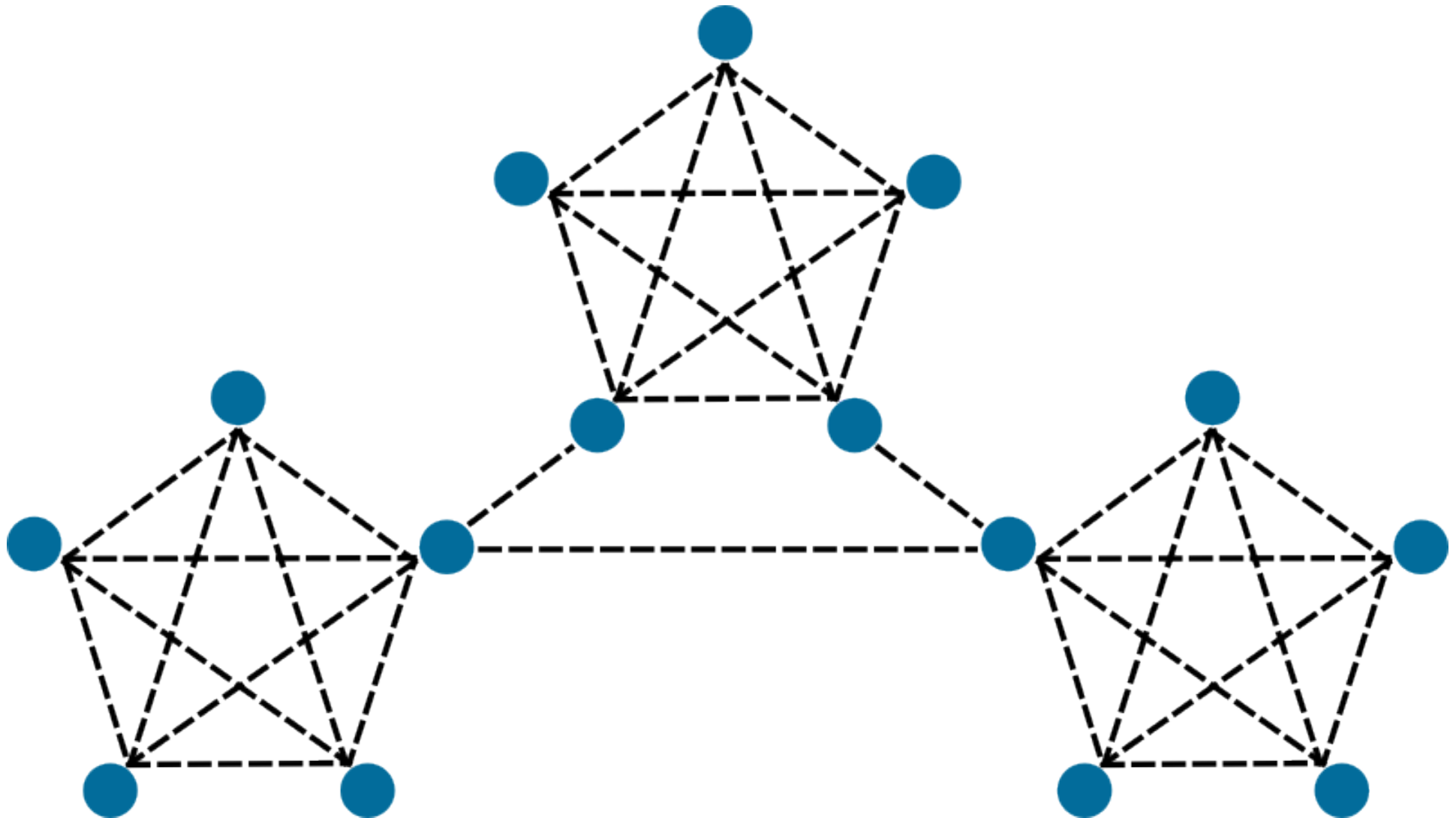


The Internet is a partial mesh. It's mostly hierarchical. Lots of networks have routers which are also single points of failure.

Lots of networks don't have redundant links, which also constitute single points of failure.

Just ask /San [Jose,Carlos]/ in March of 2009.

What we need is a fully connected mesh (or something close to it).



# Ad-hoc wireless + mesh routing == Mobile ad-hoc mesh network



Image credit: [freshpaint.deviantart.com](http://freshpaint.deviantart.com)  
License: CC BY-NC-SA v3.0 Unported

We can already do this, but we need to make it easy.

# Design Goals

- Cheap, readily available equipment
- Improvisable
- Rapidly deployable
- Robust and reliable
- Secure
- Low maintenance



# Design Constraints

- Solving Katrina first, Egypt second
- A small group of geeks can deploy the solution to enable a larger community to connect.
- Not all nodes have (or need) the mesh software
- Sufficient tools available to potentially accomplish arbitrary tasks
- Minimal collusion required
  - Organically evolving network
  - ad-hocracy



# Wireless network

- Ad-hoc wireless clients
- Mesh nodes
  - Laptops or netbooks
  - Hacked routers
  - Desktop PCs
  - Plug computers
- Inter-mesh links
  - Cantenna
  - Dial-up
  - Something more exotic?

# Mesh routing

- Route traffic through the network
- Route traffic between networks
- Support both mesh and non-mesh nodes
- Protocols
  - OLSR
  - Babel
  - BATMAN-adv

# OLSR (Optimized Link State Routing)

- Not optimized for wireless
  - No link-quality awareness
- Routing loops are possible
  - Loop detection is just now being implemented
- Tries to propagate the full routing table to every node

# Babel

- Enhancement of OLSR
- Distance vector routing protocol
- Uses link quality to help determine optimal routes
  - Latest version is traffic-density aware
- Converges rapidly
- Loop avoidance
- No kernel code
- Works by managing the OS routing table
- Only supports IPv4 & IPv6 at layer 3
- Easy to start
- Minimal configuration - config files are four lines at most
- Easy to troubleshoot

# BATMAN-adv

- Better Approach To Mobile Ad-hoc Networking
- Also has link-quality awareness
- Also has loop avoidance
- Requires a kernel module
  - Included in kernel tree since v2.6.38
- Provides a virtual layer 2 interface
  - Works with any layer 3 protocol
- Very active community
- Works quite well
- Challenging to troubleshoot
  - batctl utility has a steep learning curve
  - Doesn't lend itself to rapid deployment
  - batctl isn't included with source for kernel module

# Resources provided by Byzantium Nodes

- Microblog
- Etherpad-Lite
- Web chat
- Telephony
- File dump
- Streaming audio server
- Whatever else the admin can set up.

All of these are existing software.

XMPP supports many of these features already.

# The Byzantium Live Distribution

- LiveCD/LiveUSB
- Based on Porteus Linux (<http://porteus.org/>)
  - Binary compatible with Slackware!
- Mesh routing software
  - Including source code
- Software development/debugging tools
  - GCC
  - Perl
  - Python
- Network troubleshooting/monitoring tools
- Resource hosting software
  - LAMP

# The Byzantium Live Distribution

- Utilities for easy setup and configuration
  - Dedicated web control panel
  - Porteus' scripts for manipulating the system
- Multi-lingual documentation
- Optional multi-lingual library (in progress)
  - CD3WD (<http://www.cd3wd.com/>)
  - /Where There Is No \*/i (<http://hesperian.org/>)
- Utilities for replication in the field



# Technical difficulties

- Node IP configuration
  - Pseudo-random RFC-1918 address (192.168/16)
  - arping used to detect duplicates
  - Assigns to mesh interface as a /32
- Client configuration
  - DHCP - dnsmasq
  - config files are generated by control panel
  - One wi-fi interface only? No problem!
    - IP alias interacts with clients – wlan0:1
  - Pseudo-random netblock generation/detection algorithm
  - All clients placed in a 10/24

# Technical difficulties

- Resource discovery
  - Bonjour/ZeroConf/mDNS
  - Graffiti - “For a good time, URL <http://www.byzantium.mesh/>”
  - QR Codes
  - P2P searchable resource directory service
- Inter-mesh links
  - Exotic != {easy, reliable}
  - May be specific to use case
  - Execution over style
  - Sometimes repurposing junk results in slightly higher quality junk
- Easy, intuitive user interface
  - Configuring mesh routing daemon
  - Configuring networking
- Distributing the solution before the crisis
  - Getting into hands of people before they need it

# Comments? Questions?

<http://wiki.hacdc.org/index.php/Byzantium>