Spiny Android

Tom Tantillo, Abhishek Prakash, and Daniel Obenshain

The original Android logo is from Android.com.

Android Phones

- Android is a new, popular smartphone platform.
- On average, there are 350,000 new Android phones activated per day.
- Android phones are reported to be easier to work with than iPhones.

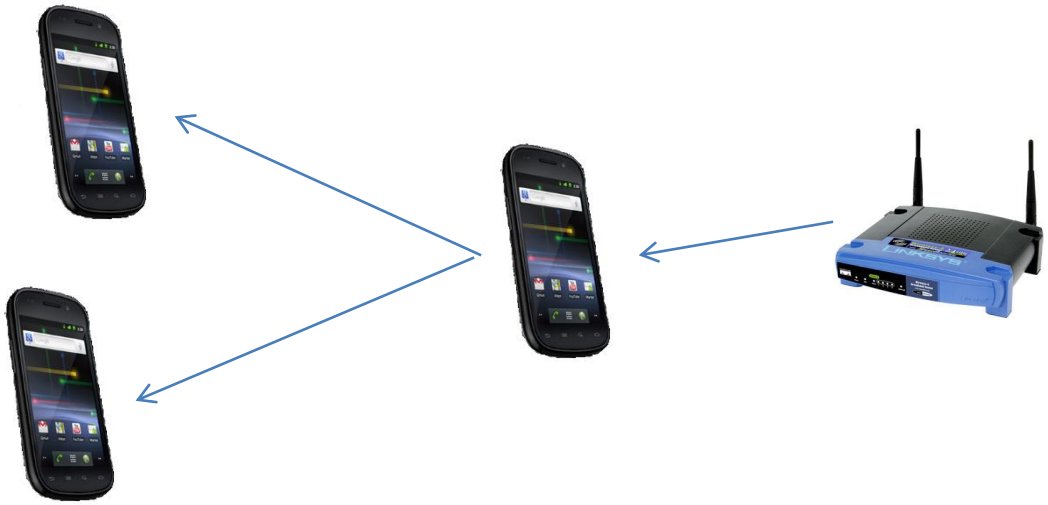
Phandroid. 15 February 2011. Retrieved 2011-02-25.

Motivation

- You’re on a subway car.
- You’re bored.
- You don’t have any bars.
- Your friend next to you does.
- Wouldn’t it be nice to be able to use his connectivity?

Motivation

- A mesh network of cell phones would allow users to share connectivity.
- If any user has connectivity, they all do.



A Mesh Network

- We wanted to make an app to run on android phones.
- This app would connect the phones via any means possible (WiFi, Bluetooth, etc.) and would share connectivity among them.

Android Programming

- Normally, Android Apps are written in Java.
- They run in a Dalvik Virtual Machine, which protects them from each other and protects the phone from bad Apps.

Android Programming

- This did not work for our purposes.
- Instead, we wanted to write raw C code that could access all parts of the phone.
- We rooted the phones and found a way to compile C code so that it would run on the Android.

Porting Spines

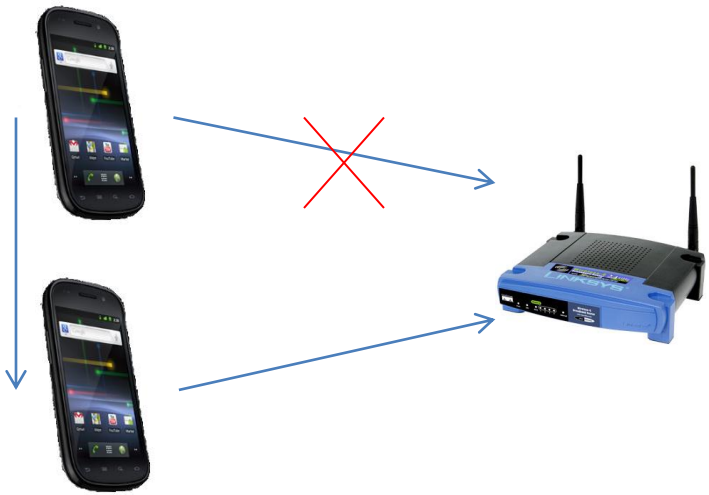
- In order to port the spines library, we cross-compiled it to the ARM architecture and linked it to all the appropriate libraries in the Android kernel.

Porting Spines

- After doing all this, we had the following:
- If the phone was not rooted:
  - We can start spines as a daemon while the phone is plugged in to a laptop.
  - It will keep running.
- If the phone was rooted:
  - We can start spines from the phone itself.
- This is because the terminal application is an Android App.

Raw Sockets

- We can use raw sockets to capture packets.
- We can also use them to insert packets.
- In this way, we can artificially force packets to go on our Spines network, even though Internet connectivity is fine.



Proof of Concept Part 1

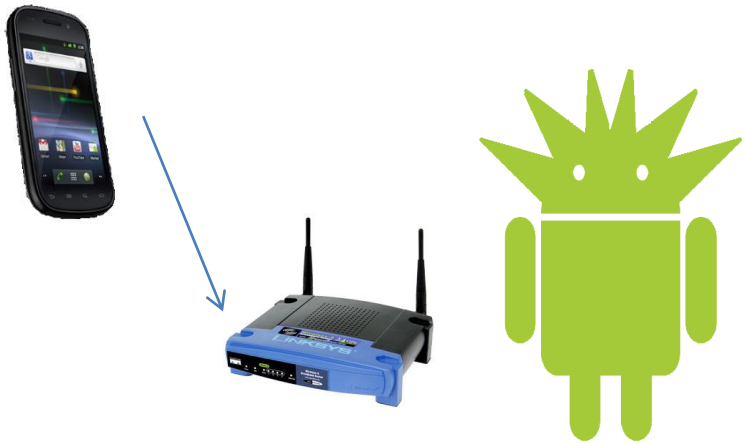
- First phone (with browser)
- We capture packets with raw sockets, prevent them from leaving (with IP tables), loop them back via local interface (127.0.0.1)

Proof of Concept Part 2

- Send the packet on spines to the gateway phone

Proof of Concept Part 3

- Gateway phone uses NAT table to send the packet to the Internet
- Get the reply, sends it on spines
- This part is not completed



Proof of Concept Part 4

- Phone 1 gets the reply packet via spines
- It sends it again on its local interface so that the browser application can get the packet.

Proof of Concept

- We would like to be able to browse on one phone, while all those packets travel over our network and out the other end to the Internet and all reply packets travel back the same way.