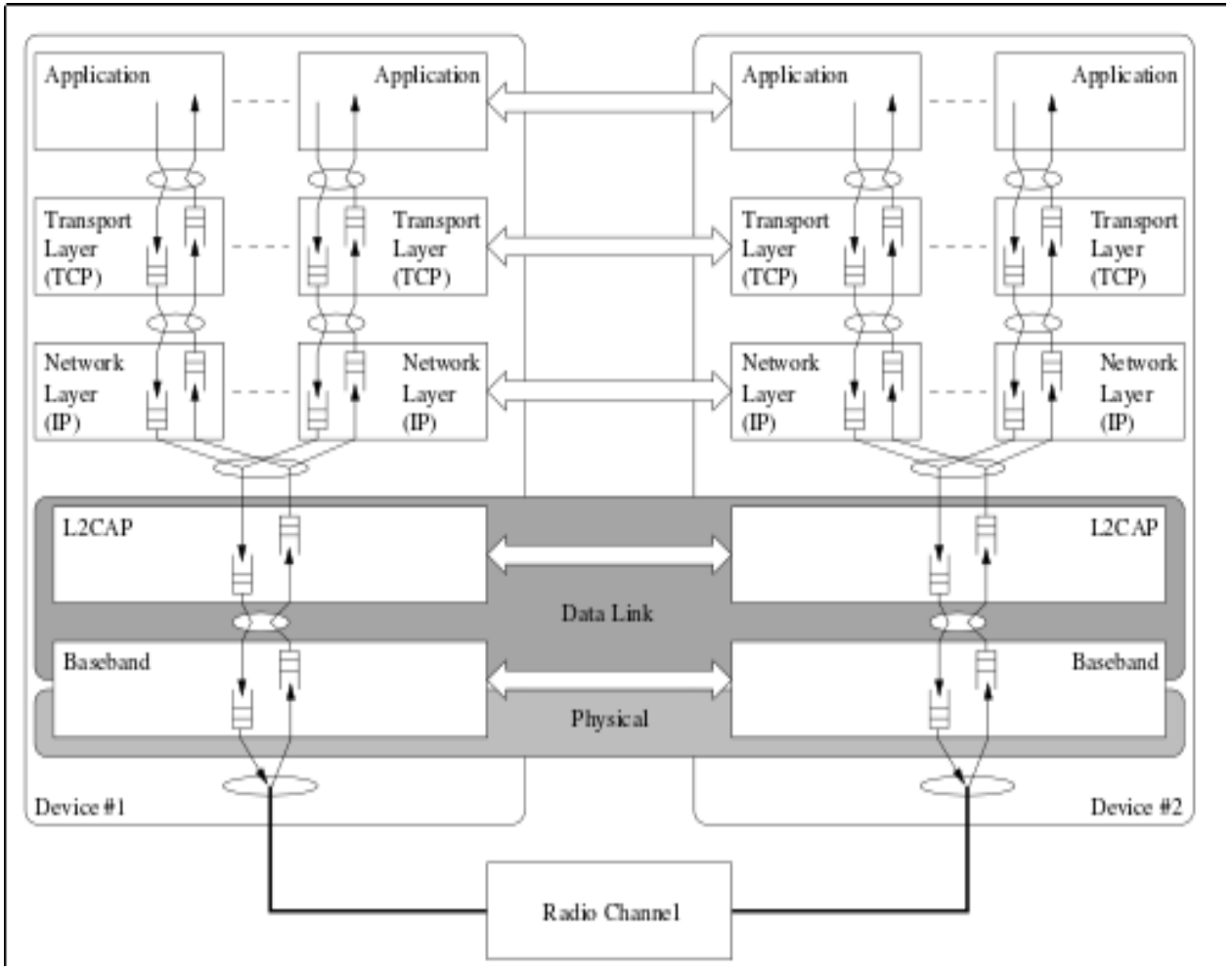
Networking 2

IP over Bluetooth



IP over Bluetooth Part 1: Setup

- Connect your Pis directly to the monitor, keyboard, and mouse, login, and start X
  - No VNC or SSH connections
- We will make a point-to-point connection between 2 Pis:
  - Organize into groups such that each group has 2 Pis
- Install plugable bluetooth USB adapter in each Pi
- Install bluetooth, bluez, bluez-compat, bluez-tools, and bridge-utils packages on each Pi:

  sudo apt-get install bluetooth bluez bluez-compat bluez-tools bridge-utils
  - This will take awhile...
- Start the bluetooth daemon on each Pi:

  sudo service bluetooth start

IP over Bluetooth Part 1 (cont): Setup

- View and set properties on BT adapter on each Pi

  sudo bt-adapter -i

  - record the adapter hardware address

  sudo bt-adapter --set Discoverable 1

  sudo bt-adapter --set Pairable 1

- Check for nearby adapters/devices on each Pi

  hcitool scan

  - make sure you see the adapter that you hope to pair with

- Although not manditory, you can remove the ethernet cable from the client Pi; the host Pi requires a ethernet connection

IP over Bluetooth Part 2: Pairing

- This is a time sensitive operation!
- On the Pi designated as the client, type the following but do **NOT** hit ***enter:***

  sudo bluez-simple-agent hci0 01:01:01:01:01:01

  - Replace ***01:01:01:01:01:01*** with the host Pi's MAC address
- On the Pi designated as the host, enter the following:

  sudo bluez-simple-agent
- As soon as you have entered the host Pi's *bluez-simple-agent* command (directly above), enter the client Pi's *bluez-simple-agent* command typed earlier
- When prompted, use ***1234*** as the pass-code on both Pis
- The host Pi's terminal is locked by the *bluez-simple-agent* command; after pairing is complete, it's ok to abort it (ctrl-C, ctrl-C) or move to a new window

IP over Bluetooth Part 2 (cont): Pairing

- On the host Pi, view and set the properties for each client Pi (we have only 1 client Pi)

  sudo bt-device -l 01:01:01:01:01:01

  - Replace ***01:01:01:01:01:01*** with the client Pi's MAC address
    - should see the client HW address as an "Added device"

  sudo bt-device --set 01:01:01:01:01:01 Trusted 1

IP over Bluetooth Part 3: Create pan0

- Enter the following commands on the **host Pi only**:

  sudo brctl addbr pan0

  - this creates pan0; verify this using the ifconfig command

  sudo brctl setfd pan0 0

  sudo brctl stp pan0 off

  sudo ifconfig pan0 192.168.2.1 netmask 255.255.255.0

  sudo bt-network -s nap pan0

  - The host Pi's terminal is locked by this command; move to a new window; do NOT abort it

IP over Bluetooth Part 4: Create bnep0

- Enter the following commands on the **client Pi only**:

  sudo pand -c 01:01:01:01:01:01

  - Replace ***01:01:01:01:01:01*** with the host Pi's MAC address
    - this creates bnep0; verify this using the ifconfig command

  sudo ifconfig bnep0 192.168.2.2 netmask 255.255.255.0

    - Verify the IP address of bnep0 using ifconfig
- On the client Pi, verify the connection to the host Pi:

  ping 192.168.2.1

- On the host Pi, verify the connection to the client Pi:

  ping 192.168.2.2

- You should be able to *ping*, *SSH*, *HTTP*, etc between the client and host.

What did we just do?

- We created a Personal Area Network (PAN) of Bluetooth devices capable of providing services overTCP/IP. The network is refered to as a *piconet*.
  - Limited by Bluetooth transmission range (up to ten meters)
- Pan0 is a Network Access Point (NAP), a service that routes network packets between PAN Users (PANU) using the Bluetooth Network Encapsulation Protocol (BNEP)
  - The Bluetooth Network Encapsulation Protocol (BNEP) emulates Ethernet
  - A separate bnep interface is created on demand for each client device that connects (bnep0, bnep1...bnepN)
  - The Bluetooth host providing the NAP service acts as an Ethernet bridge that forwards BNEP packets between each of the connected Bluetooth devices (PANUs), including the Ethernet of an external network

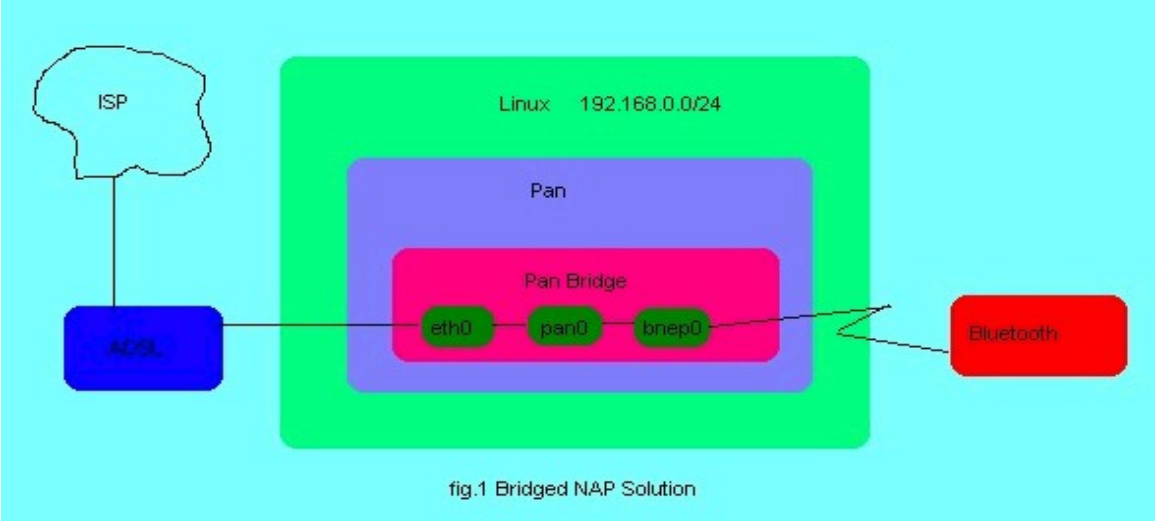


fig.1 Bridged NAP Solution

The rest of the story

- The procedure used to create pan0 and bnep0 is similar to that described in this article
- You can join up to 7 Pis (1 host & 6 clients) in a PAN using this procedure.
- The personal area network, pan0 (with NAP) and the connection, bnep0, are lost when the Pi shuts down and must be re-established by repeating the steps in Parts 3 and 4.
  - The pairing of the two Pi's remains after shutdown
- The system can be configured to recreate pan0 and bnep0 at startup as described in this article
  - I HAVE NOT TESTED THE PROCEDURE DESCRIBED IN THIS ARTICLE

A little about Bluetooth

- Bluetooth is a low cost, short range, low power radio technology used to connect devices
  - Started in 1994 by Ericson
  - Fast growing
  - Intended for mobile devices
  - Uses short-wavelength UHF radio waves from 2.4 to 2.485 GHz
  - Each Bluetooth transceiver is given a unique 48-bit Bluetooth Device Address
  - Typically used for point-to-point connections in a master/slave configuration
  - It is a packet-based protocol
- A great tutorial from SparkFun

Bluetooth communication: Discovery and Pairing

- Two Bluetooth devices that want to communicate must use the same frequency hopping sequence
  - Devices in the *inquiry (i.e., discoverable) state* send short ID packages with a predetermined hopping pattern at a high repetition rate---only possible to detect devices in this ***discoverable*** state
  - When a device detects an ID packet, it waits a random period and then responds with a Frequency Hop Synchronisation (FHS) packet
- Next a device must take on the master role and connect to a device in a slave role (i.e., ***pairing***)
  - The master transmits the slave's access code (DAC) in different hop channels and listens until it receives response from the slave
  - When the master detects a response it sends its own FHS and sets up an L2CAP link via the Link Manager.
  - After this the Service Discovery Protocol (SDP) will determine the device's capabilities.

Bluetooth communication: BNEP

- BNEP emulates an Ethernet segment
  - Hides the underlying master-slave piconet topology
  - It runs over L2CAP
  - It reuses the Ethernet packet format
  - It encapsulates the IP packet in BNEP headers and then in an L2CAP header and sends it over the L2CAP connection