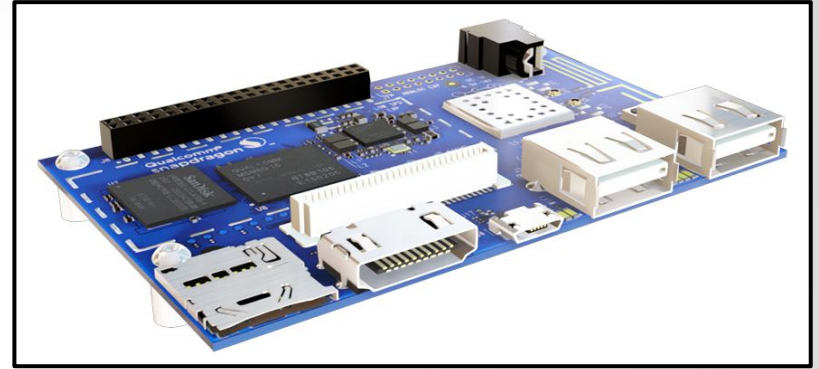


Lesson 1

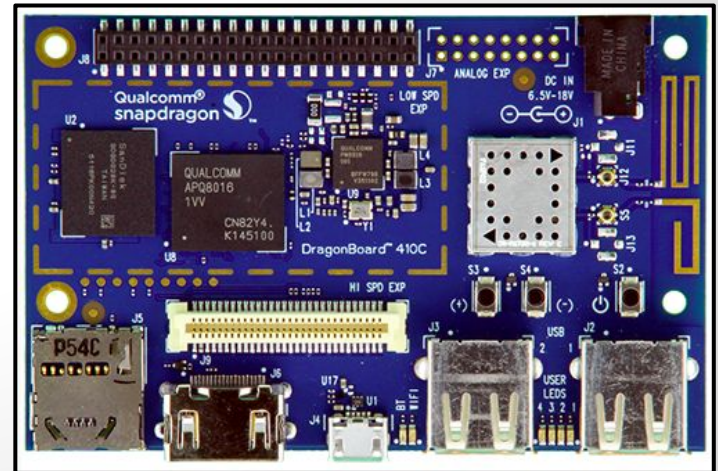
Acronyms and Terminology

Lesson 1 | Acronyms and Terminology

1 - Acronyms



2 - Frequent Terminology



1 - Acronyms

1 | Acronyms

IP - Internet Protocol - communication protocol for sending data/packets across a network (e.g. computer to computer)

VoIP - Voice over Internet Protocol - voice communication through IP

PSTN - Public Switched Telephone Network - traditional telephone service

SIP - Session Initiation Protocol - communication protocol that controls multimedia communications, phone/voice, video, messaging

ATA - Analog Telephone Adapter - device that connects an analog device (e.g. telephone, fax) to a digital network (VoIP)

ADC - Analog-to-Digital Converter - device that converts an analog signal to a digital signal

DAC - Digital-to-Analog Converter - device that converts a digital signal to an analog signal

1 | Acronyms

RTP - Real-time Transport Protocol - network protocol used to send audio and video over IP protocols; real time transfer of streams (phone calls, live tv)

API - Application Programming Interface - routines/protocols/tools for building software; basically defines functionalities of a component

UDP - User Datagram Protocol - a core piece of IP; sending messages or datagrams without any overhead (no transmission channel setup, thus unreliable). Use when dropping packets is favored over delayed packets

TCP - Transmission Control Protocol - another core of IP; opposite of UDP due to reliability (connection setup -- allows for error checking, ordered communication)

1 | Acronyms

SSL - Secure Sockets Layer - Predecessor of TLS, often paired as: TLS/SSL

TLS - Transport Layer Security - provides security to communication systems over a computer network, cryptography (encryption/decryption)

SCTP - Stream Control Transmission Protocol - transport-layer protocol; sends messages like UDP but with reliable, in-sequence transport like TCP

SDP - Session Description Protocol - format to describe streaming media initialization parameters

MSML - Media Server Markup Language - language used to control/manipulate media (audio/video) streams over networks

1 | Acronyms

XML - Extensible Markup Language - way to encode documents that humans and machines can read

IPvX - Internet Protocol version “X” - First accepted internet protocol was IPv4, its successor is IPv6

GTK - GIMP Tool Kit - tool kit used to create GUIs

GUI - Graphical User Interface - an interface (for interactions between the user and the software) using visuals (e.g. video games)

ICE - Interactive Connectivity Establishment - similar to SIP; a communication protocol but where one can discover public IPs to communicate to/with

2 - Frequent Terminology

2 | Frequent Terminology

Protocol - set of rules for how systems communicate

Datagrams - basic piece of data to be sent

Packets - grouping transmitted data into nice blocks

Internet Protocol Suite - also known as IP; a communications protocol

Codecs - coder-decoder, encodes or decodes a digital signal

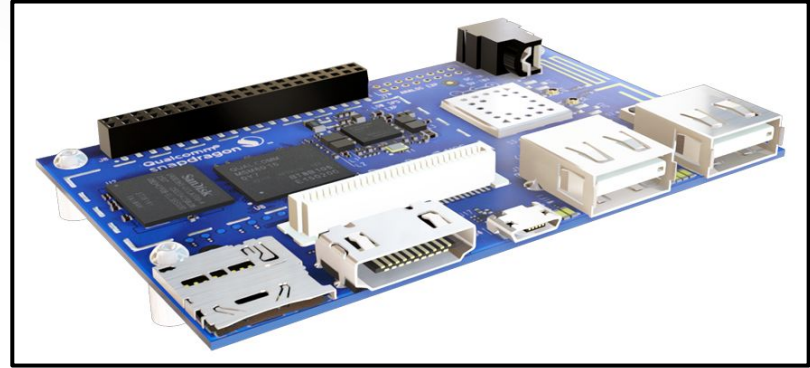
- encoded signals are for transmission, storage, or encryption
- decoded signals are for playback or editing

Containers - file formats, contains the codecs

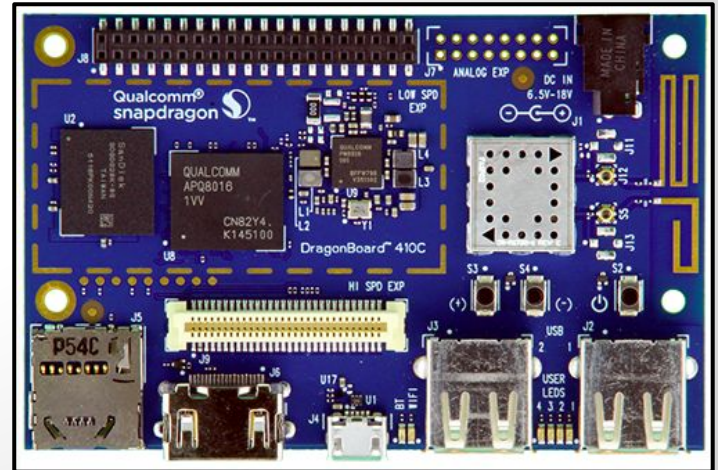
GStreamer - multimedia framework that allows for audio/video playback, editing, streaming, recording

Lesson 1 | Summary + A Look Back

1 - Acronyms



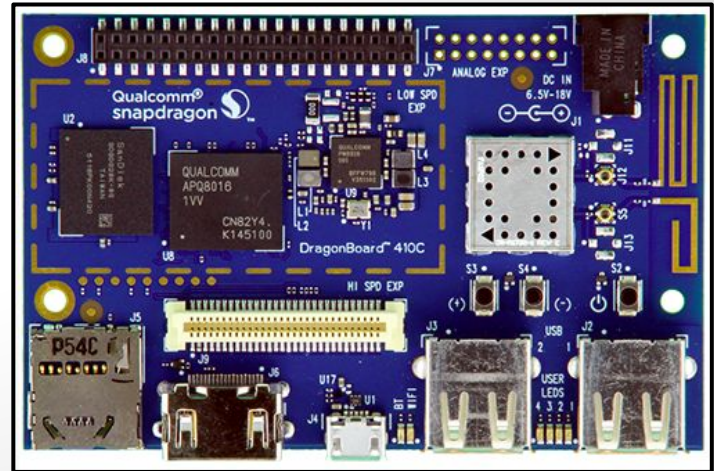
2 - Frequent Terminology



Lesson 2

Concepts

3 - Packet-switched Network



1 - Software Stack

1 | Software Stack

Front End

- more client sided
- UI (User Interface)
 - GUI, how user interacts with the software
- UX (User Experience)
 - user satisfaction; rating for the UI design

Back End

- more server sided
- algorithms/data structures implementations
- things that the user can't see
 - how data is stored/used

Full Stack

- combination of both
- how the client and server interacts; the link between the two

1 | Software Stack



source: <http://www.sitepoint.com/full-stack-developer/>

1 | Software Stack

Some examples:

- You want to build a house
 - front end → designers, gardeners
 - back end → architects, contractors
 - full stack → you, consultants
- You want to build a Boggle game (<https://en.wikipedia.org/wiki/Boggle>)
 - front end → how the board looks, flow of the GUI, working UI elements, login screen, point system, etc
 - back end → how the dictionary is stored, how the game board is stored, how are usernames/passwords stored, etc
 - full stack → needs to know all of these; completes the connections

2 - Internet Protocol

2 | Internet Protocol

- Principal communications protocol in the Internet Protocol Suite
- Routing functions enables internetworking

Connects a computer network to other networks

- Delivers packets from source host to destination host

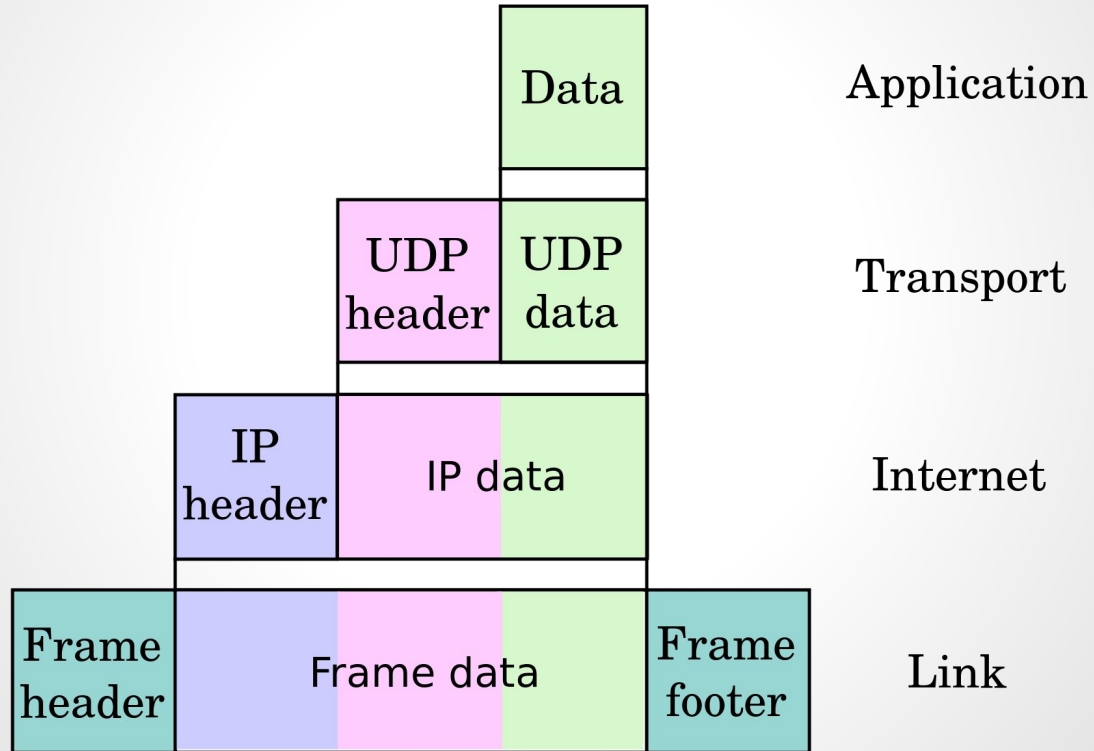
Done by checking IP addresses in packet headers

- Internet Protocol Version 4 (IPv4) was first major version of IP
- Internet Protocol Version 6 (IPv6), successor IP



2 | Internet Protocol

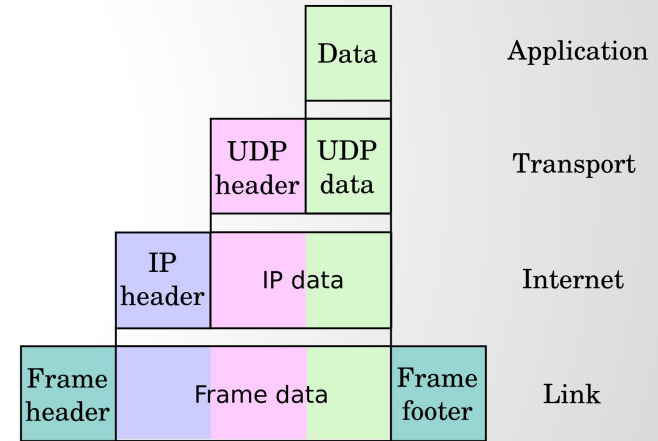
Datagram Construction:



2 | Internet Protocol

Datagram Construction:

- Every datagram has a header and a payload
- IP header is tagged with source and destination IP address, meta-data for routing
- Payload is actual transported data
- Nesting data payload in packet with header is called encapsulation

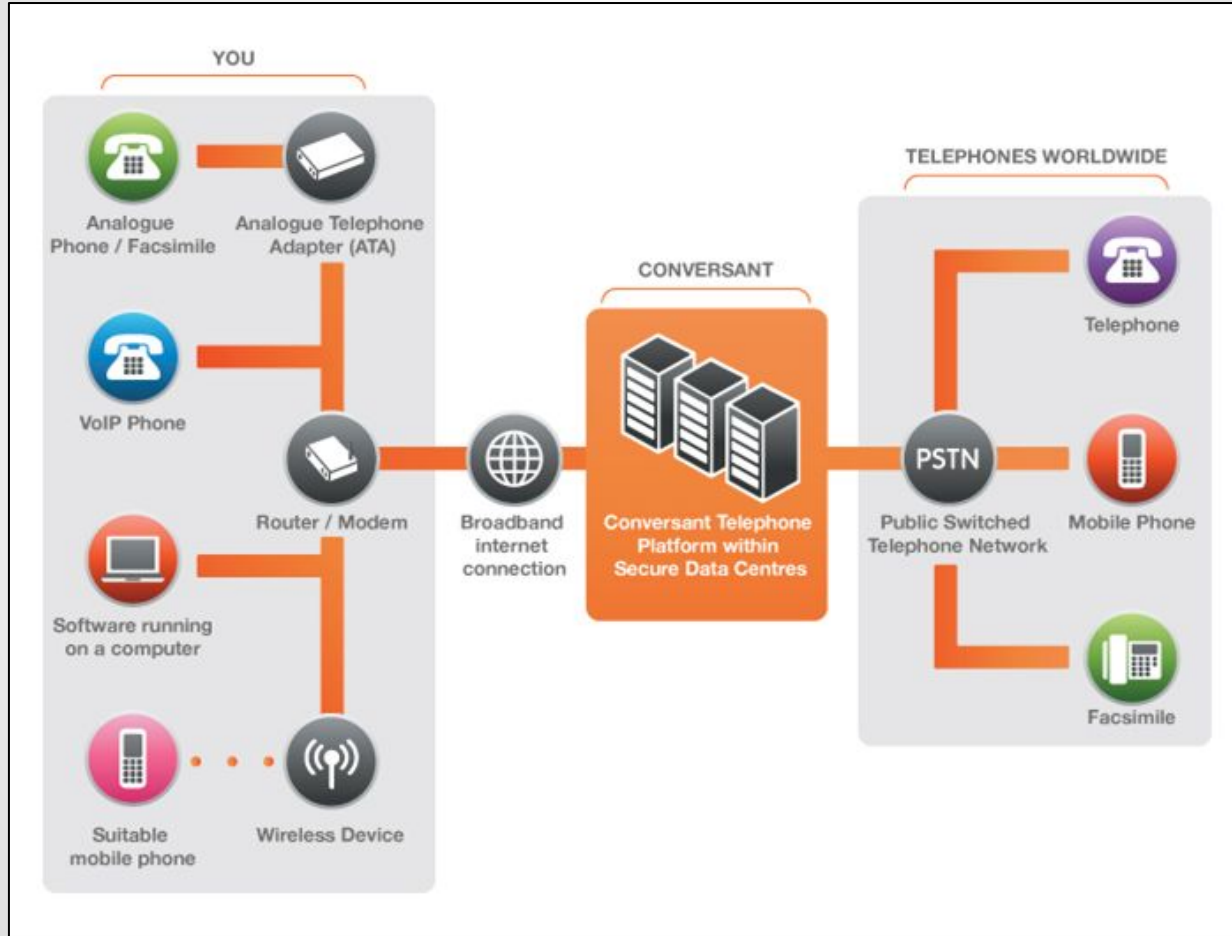


3 - Packet-switched Network

3 | Packet-switched Network

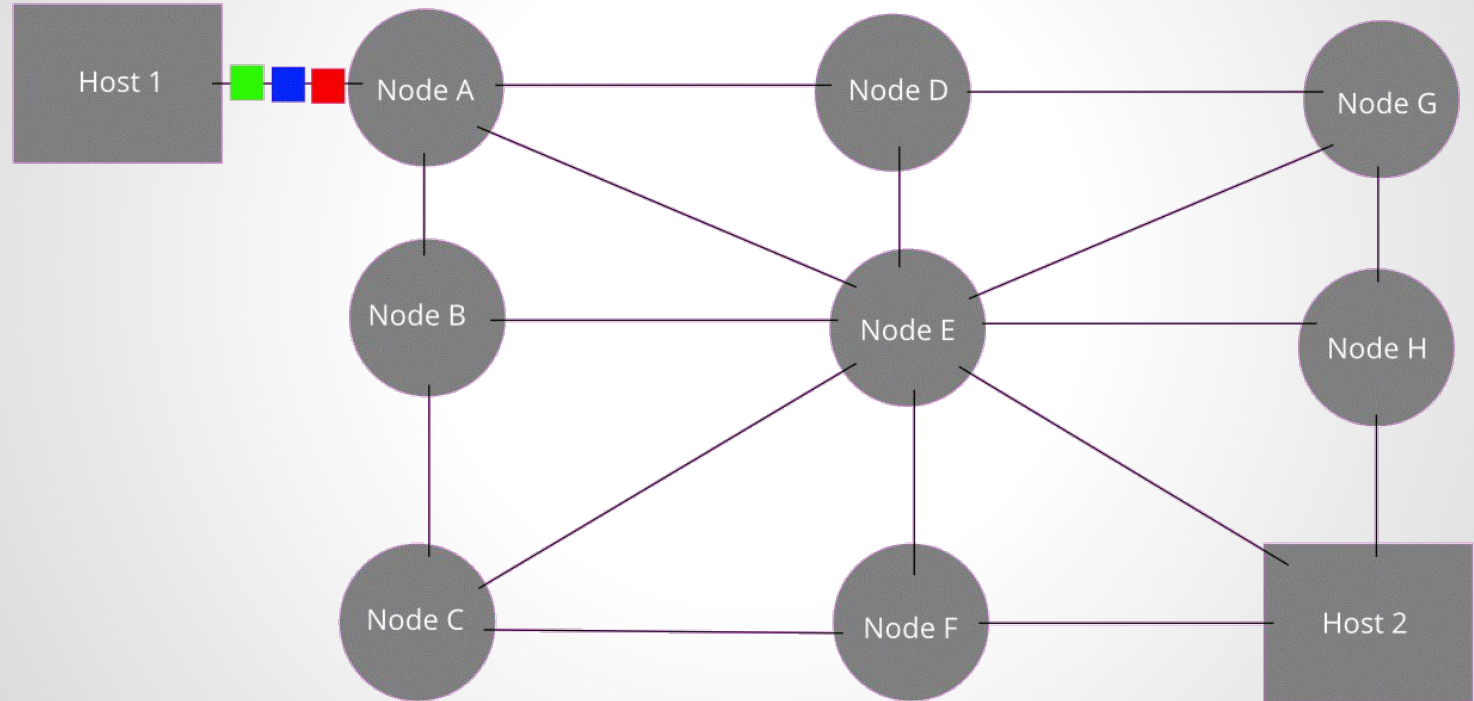
- Digital communication method using data blocks called packets
- A packet contains a header and a payload
 - header contains the destination
 - payload is the message (data)
- Packets travel through channels, going from node to node
 - channels are datapaths from one source to a destination
 - different messages can follow the same channel for transmission
 - nodes can be routers or switches

3 | Packet-switched Network



3 | Packet-switched Network

The original message is Green, Blue, Red.

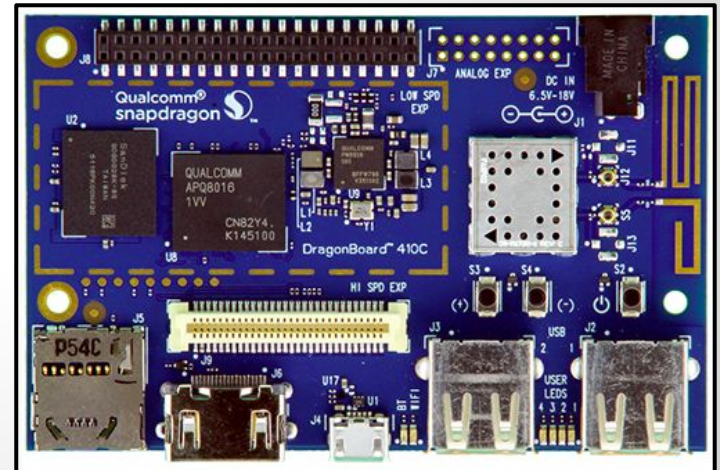
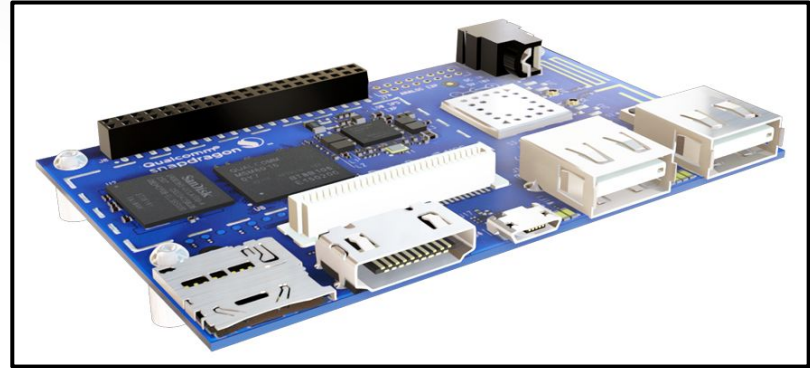


Lesson 2 | Summary + A Look Back

1 - Software Stack

2 - Internet Protocol

3 - Packet-switched Network



Lesson 3

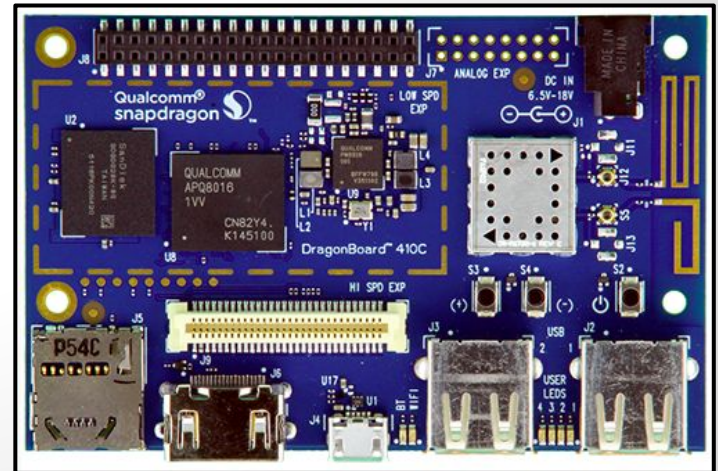
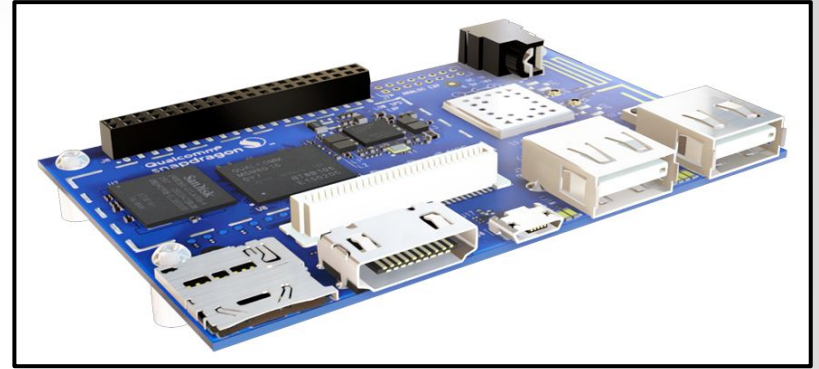
Supplemental Material

Lesson 3 | Supplemental Material

1 - PSTN: The Traditional Phone

2 - Linphone Source Code

3 - Codec Selection



1 - PSTN: The Traditional Phone

1 | PSTN: The Traditional Phone

- Circuit-switched network
 - telecommunication network which two nodes form a direct channel (circuit) to communicate
- Telephone lines, cell networks, satellites, etc. all connected through switching centers
- Huge, many devices are connected!
- Originally fixed-line analog telephones (landlines)
- Transition to digital phones (and Internet Telephony/VoIP)

1 | PSTN: The Traditional Phone

- About the old telephone:
 - converting sound into a signal for transmission
 - microphone or *transmitter*
 - earphone/speaker or *receiver*
 - *ringer*
 - contacts or *addresses*
 - yet less efficient than VoIP (e.g. bandwidth usage, cost)

So, if VoIP is better, why is the PSTN still around?

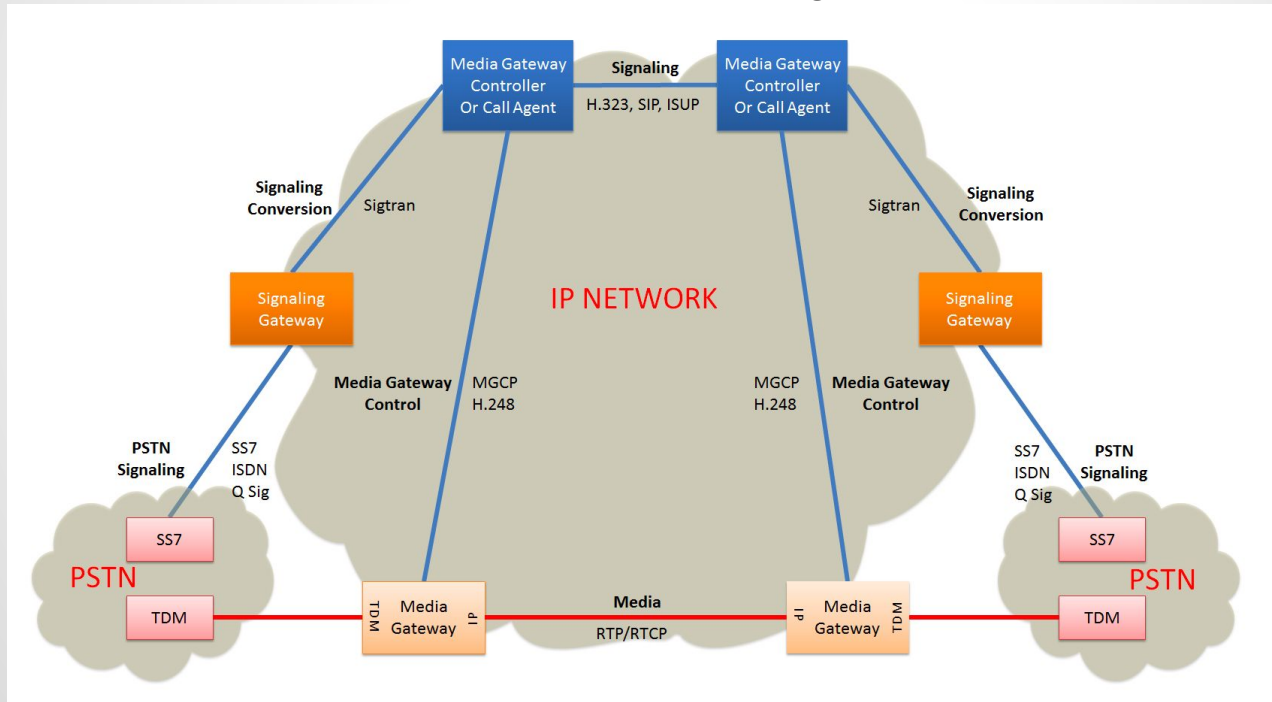
1 | PSTN: The Traditional Phone

PSTN probably isn't going anywhere anytime soon:

- infrastructure has been there for a long time
 - the PSTN is very large; has had a long time for growth
- tons of analog phones still out there
- need a way to communicate with those analog devices

1 | PSTN: The Traditional Phone

- PSTN-VoIP gateways introduced
 - devices to transform signals so the phones understand each other
 - thus PSTN will be here for a while longer



2 - Linphone Source Code

2 | Linphone Source Code

- open source VoIP software
- will talk about it more later
- tons of features; probably more than what you need
- great resource to look at

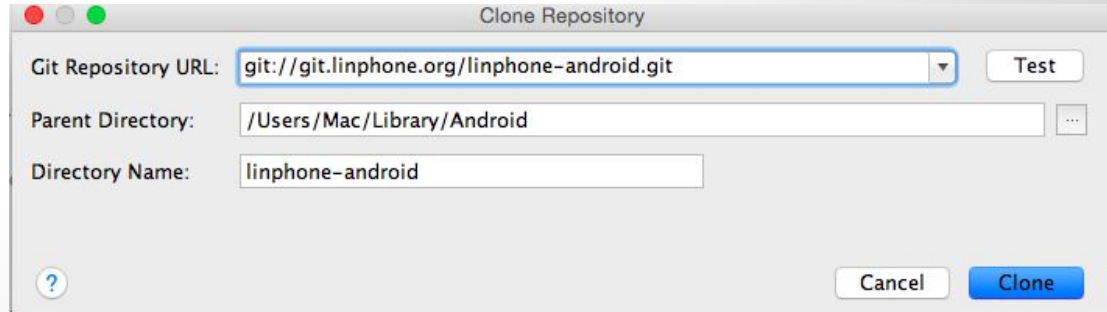
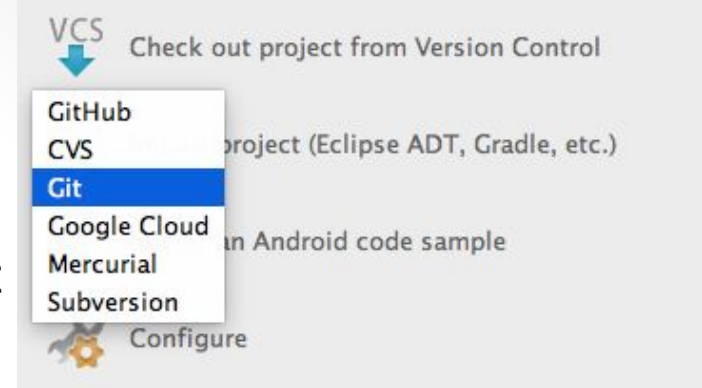
To download:

1. <http://www.linphone.org/technical-corner/linphone/overview>
2. Go to “Downloads”
 - a. Android: copy git link and use Android Studio VCS
 - b. Ubuntu: clone the desktop repo using the terminal

2 | Linphone Source Code

Android:

- copy git repo link off the downloads tab
 - look for android
 - `git://git.linphone.org/linphone-android.git`
- launch Android Studios
- check out from Version Control
- paste in the URL
- click “clone” to start the cloning and the project should show up!



2 | Linphone Source Code

Ubuntu:

- launch the terminal
- move to desired directory
- copy the entire clone command in the terminal and hit enter
 - `git clone git://git.linphone.org/linphone-desktop.git --recursive`
 - remember to get the desktop version!!
- the download should begin; once it's done you should see all the source files

2 | Linphone Source Code

How does this help?

- see what the developers were thinking
- great example if you want to build your own VoIP app
- You will be able to:
 - navigate through source files and see their interactions
 - see the important objects/classes/methods/functions needed for:
 - phone calls
 - video calls
 - text messaging, etc.

3 - Codec Selection

3 | Codec Selection

Codecs - encoding/decoding digital data stream or signal:

- FFmpeg, DivX, H.264, etc
- the same codec used to encode a signal must be used to decode

Why do we need multiple codecs? Why not just one? Or none?

- audio and video codecs → different forms of data
- uncompressed files → takes up lots of space, slow downloads, slow streaming
- allows for different forms of compression
 - lossless vs lossy

Can't watch a video or listen to audio?

- likely due to missing codecs

3 | Codec Selection

- Codecs *are not* containers
- Containers:
 - have audio/video codecs
 - are file formats
 - .avi, .mov, .mp4, .wav, etc
 - are bundles of files
 - contains the audio/video data and whatever else the file bundle needs (e.g. subtitles)

3 | Codec Selection

Remember, codec used to encode data must be used to decode it (correctly)

So which one to use?

- Depends!
 - Lossy or lossless?
 - High or low speed network (streaming)?
 - Small or large file size?
- Many factors as well
 - resolution (pixels)
 - number of bits to use
 - fps (video) or channels/sampling rate (audio)

Lesson 3 | Summary + A Look Back

1 - PSTN: The Traditional Phone

2 - Linphone Source Code

3 - Codec Selection

