Flexible ping utility

Ericsson

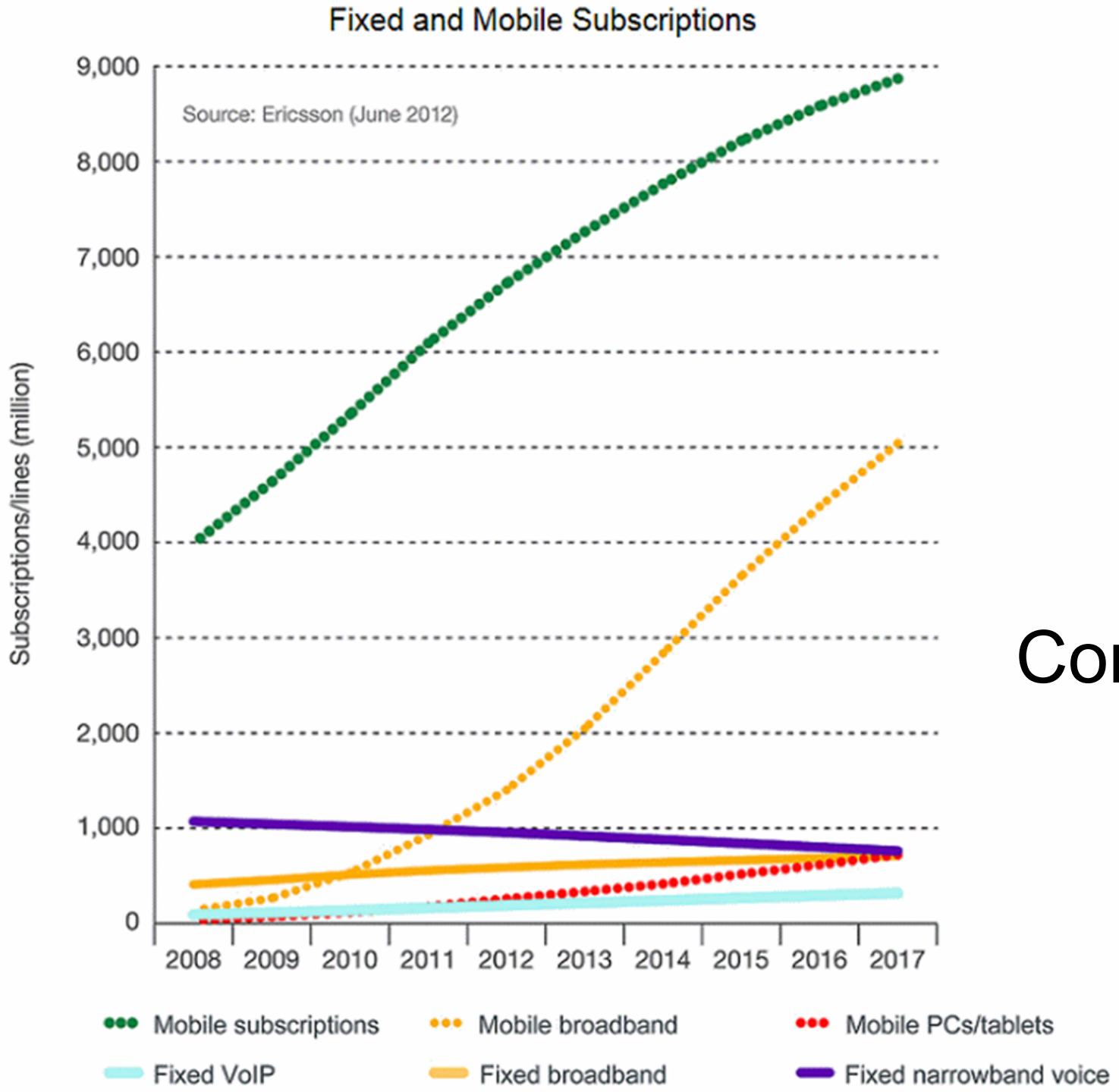
- Founded in 1876
- Produced landline telephones popular from 1950-1990
- Created Bluetooth in 1994
- Launched a joint mobile-phone venture with Sony in 2001

Jordgubbe (2005) *Bust of Lars Magnus Ericsson*

A Networking Giant

- Headquartered in Stockholm, Sweden
- In 2012, largest mobile telecommunications maker with market share of 38%
- Operations in over 180 countries
- Employ over 100,000 people world-wide
- Have produced everything from telephones and network cables to power modules and radar systems

Ellgaard, H. (2007) *1953 Ericofon "Cobra"*



The Future of Connectivity

What's Ping?

- Written originally by Mike Muuss in 1983
- The premiere utility for troubleshooting problems in an IP network
- Sample uses of Ping:
  - Ping your computer: is TCP/IP working?
  - Ping local router: is it running?
  - Ping somewhere else: can you connect to it?

- # of pings to be sent
- Adjustable buffer size in bytes
- Minimum, maximum, and average reply times
- Lost packets
- Time granularity in milliseconds

Basic Ping Features

- Existing ping utilities do not offer enough flexibility
- Our application will reduce the amount of time Ericsson engineers spend making scripts.

ePing

- Includes all basic ping features
- More comprehensive statistics
- Send pings on commanded intervals
- Random interval

ePing Features

- Adjustable size by payload or total IP datagram
- Random payload size
- Increasing payload size
- Pings excluded from summary
- IPv4/IPv6 functionality
- Option for .csv file output

ePing Features

Some basic UDP functionality:

- Send UDP packets
- Configurable size, port, and recipient

UDP listening mode:

- Listen on chosen port for UDP ping
- Return UDP datagram to sender

More UDP functionality, time permitting

ePing Features

- The final product will:
  - Be platform independent
  - Run on the command line without installation
  - Be released open source

Project Overview

Technologies

- Written in C++
- Using Git/Github in place of SVN/Trac
- Additional document sharing done with Dropbox
- Wireshark for network testing

- Prep that packet. Wrap it in protocol headers to keep it warm and safe.
- “Hey, are you there?” (starts timer) ECHO_REQUEST
- Sent through the network…
- Received (hopefully)
- Sends reply ECHO_REPLY
- Received (hopefully)
- Protocols taken-off
- Read the packet.
- Did we lose any?
- How long did it take?

Ping deconstruction

ICMP & TCP/IP

- Acronyms! OSI model! Layers!
  - **Application** – protocols for specific data communication services (FTP, SSH)
  - **Transport:** Host to host communications
  - **IP:** connects local networks
  - **Link**: communication tech for a local network

  - ICMP – primarily for errors and diagnostic functions, and works directly on top of the IP Layer.

- Raw sockets allow you access to the raw, unprocessed packet rather than letting your kernel handle TCP/IP
- Mostly used by kernel hackers, or security experts. Sometimes you might even want badly formed packets
- Puts more responsibility on the program

RAW SOCKETS

src: linuxchix.org/content/courses/security/raw_sockets

- Learning C++
- Learning Git / version control
- Learning Network Programming

Learning Experience

Difficulties Encountered

- Testing and Debugging
  - Platform Independence
  - Network firewalls
- Overseas sponsor
  - 7 hours ahead
  - Great to work with

- Fix our ICMP checksum function
- Implement various features / extensions requested in requirements document
- DNS Lookup for domain names' IP addresses

Future Work : short term

- Implement UDP ping send/listen modes
- Create a GUI
- Implement IPV6 functionality
- Pick a good name for the project
- Release as open source!

Future Work : long term

Thanks for your time!

Questions?