What is NFC?

- NFC means Near Field Communication
- It is a short-range wireless communication
- It operates at 13.56 MHz
- Data rates from 106 kbits/s to 424 kbits/s
- Range of about 4cm
- Modes:
  - Tag Read/Write
  - Card Emulation
  - Peer to Peer (LLCP)
    - OBEX over LLC
    - IP over LLC
What is the difference between NFC and RFID?

- RFID uses many frequencies including 13.56 MHz
- NFC Tag read/write and some RFID 13.56 MHz are the same thing
- NFC Card Emulation and NFC P2P are not part of RFID
- Other RFID frequencies are not part of NFC
- So, NFC and RFID are not the same thing
- But there is an intersection between them
What is the difference between NFC and RFID?

- **NFC**
  - NFC Card Emulation
  - NFC P2P

- **RFID**
  - RFID 135kHz
  - RFID 2.45GHz
  - RFID 433MHz

NFC and RFID overlap in the area of NFC Tag Read/Write, which operates at 13.56MHz.
NFC and Bluetooth

- Both are short-range communication technologies.
- NFC is much shorter range than Bluetooth.
- NFC setup is faster.
- NFC is partially compatible with RFID.
- NFC power consumption is lower.
- Bluetooth has a much higher throughput.
- Bluetooth has a lot of high level profiles defined.
- So, NFC and Bluetooth are meant for different use cases.
- NFC can be used to simplify Bluetooth pairing.
# NFC Use Cases

![NFC Use Cases Diagram](nfc-forum.org)

Source: nfc-forum.org
NFC today's importance

- File Sharing
- Mobile Payment
- NFC Smart Poster
What is NDEF?

- It means NFC Data Exchange Format
- “NDEF is a lightweight, binary message format that can be used to encapsulate one or more application-defined payloads of arbitrary type and size into a single message construct.”
- “Each payload is described by a type, a length, and an optional identifier.”
- “NDEF is strictly a message format, which provides no concept of a connection or of a logical circuit …”
- It is possible to exchange a NDEF using NFC Tag Read/Write or P2P (LLCP)
The current NFC support for Linux

Android support

- Libnfc-nxp library, provided by NXP to Google.
- 70000 lines of code.
- Strongly tied to the NXP hardware (pn544, pn65n).
- HCI support only.
- No community, exclusively maintained by NXP and Google.
- 100% userspace, complete frames are sent to /dev/pn544.
- Feature rich (target and reader mode, LLCP and SE support).
The current NFC support for Linux

Open NFC

- opennfc library, provided by Inside Secure.
- Targeted for Android, although not the default Android stack.
- 100000 lines of code.
- Strongly tied to the Inside Secure hardware.
- Slightly better architecture for additional HW support.
- No community, no mailing list, rare tarballs release only.
- 100% userspace, complete frames are sent to a device entry.
- Feature rich as well.
The current NFC support for Linux

libnfc

- Libnfc library, about 10000 lines of code.
- More community oriented, although sponsored by il4p.fr.
- Forums, SVN repo, documented website.
- Userspace implementation.
- Work in progress, missing features.
The current NFC support for Linux

What is missing?

- Vendor neutrality.
- HW independent support.
- Open development process.
- A proper kernel/user space split.
- A proper POSIX API.
The new NFC subsystem

User Space

Control commands handler

Raw data exchange handler

Core

Device Drivers

LLCP handler

socket AF_NFC imsock tmsock

generic netlink

socket AF_NFC llcpsock

Linux NFC Subsystem
Main Architecture points

- Hardware Independent
- New Hardware requires a new device driver
- POSIX API
- Generic netlink for control commands
- Sockets for data exchange
NFC control commands and events

- NFC_CMD_GET_DEVICE
- NFC_EVENTDEVICE_ADDED
- NFC_EVENTDEVICE_REMOVED
- NFC_CMD_START_POLL
- NFC_CMD_STOP_POLL
- NFC_EVENTIM_TARGETS_FOUND
- NFC_CMD_IM_GET_TARGET
- NFC_EVENT_TM_ACTIVATED
- NFC_EVENT_TM_DEACTIVATED
NFC Sockets - AF_NFC

- **Initiator mode socket**
  - connect – select and activate a target
  - write – send commands
  - read – receive responses

- **Target mode socket**
  - bind – bind the socket to an NFC adapter
  - accept – wait for being activated by an initiator
  - read – read initiator commands
  - write – answer initiator commands

- **LLCP socket**
  - Similar to TCP sockets
Current Status

- Reader mode supported
- Card emulation WIP
- NXP PN533 device driver
- HCI support WIP
- NCI patches being reviewed (contributed by Ilan Elias from TI)
- HCI and NCI support means NXP and TI hardware support
What's next for the NFC subsystem?

HCI and NCI support

- HCI is an ETSI defined spec.
  - Mostly defined to help smartcards integration.
  - Lots of vendor specific extensions.

- NCI is the NFC Forum answer.
  - Currently a draft.
  - Much more generic and NFC oriented.
Linux NFC Subsystem

NFC CORE

NFC Sockets

NFC Netlink

nfc_allocate_device()
nfc_register_device()

nfc_hci_allocate_device()
nfc_hci_register_device()

nfc_nfc_allocate_device()
nfc_nfc_register_device()

NFC HCI LAYER

NFC HCl based driver

NFC NCI LAYER

NFC NCI based driver

NFC Driver

Linux NFC Subsystem
What's next for the NFC subsystem?

Card emulation mode

- POSIX API, card emulation is NFC's server side.
- One sock_addr_nfc structure for both modes.
- Add card emulation mode to the polling loop.
- Which RF technology do we want to poll?
- bind, listen, accept, recv, poll: Your typical networking API.
What's next for the NFC subsystem?

LLCP (Logical Link Control Protocol) sockets

- Asynchronous Balanced Communication using a symmetry mechanism.
- Protocol multiplexing.
- Connectionless and Connection oriented data transport.
- New NFC socket protocol.
- Simple NDEF Exchange Protocol (SNEP) on top of LLCP
- Google Push protocol (NPP).
- No more manufacturer specific tag commands.
- All NFC advantages without the manufacturer's legacy.
- IP and OBEX bindings.
What's next for the NFC subsystem?

User space daemon

- NFC events and properties exported through D-Bus.
- Adapters, targets and tags objects.
- Tags reader and emulation support.
- SNEP and NPP integration
- Many similarities with BlueZ
- Should be open sourced by next week.
What's next for the NFC subsystem?
What's next for the NFC subsystem?

Secure Elements

- NFC adapter and secure element communication.
- Transaction entirely handled by the SE itself:
  - Smartcard applets
  - Proprietary protocol for the APDUs (not NFC defined).
- New simple netlink API:
  - Enabling and disabling the SWP link.
  - HCI events reports, new netlink events.
- Several potential user interaction media:
  - SIM Application Toolkit (STK), link with telephony stack.
Thanks!
#linux-nfc at freenode

Linux NFC subsystem developers:
Lauro Ramos Venancio <lauro.venancio@openbossa.org>
Aloisio Almeida Jr <aloisio.almeida@openbossa.org>
Samuel Ortiz <sameo@linux.intel.com>