

Transmission of IPv6 Packets over Near Field Communication

draft-ietf-6lo-nfc-00

Y-G. Hong, Y-H. Choi (ETRI),
J-S. Youn (DONG-EUI Univ.), D-K. Kim (KNU)
J-H. Choi (Samsung)

6lo WG Meeting@IETF 92 – Dallas, USA
2015.3.24

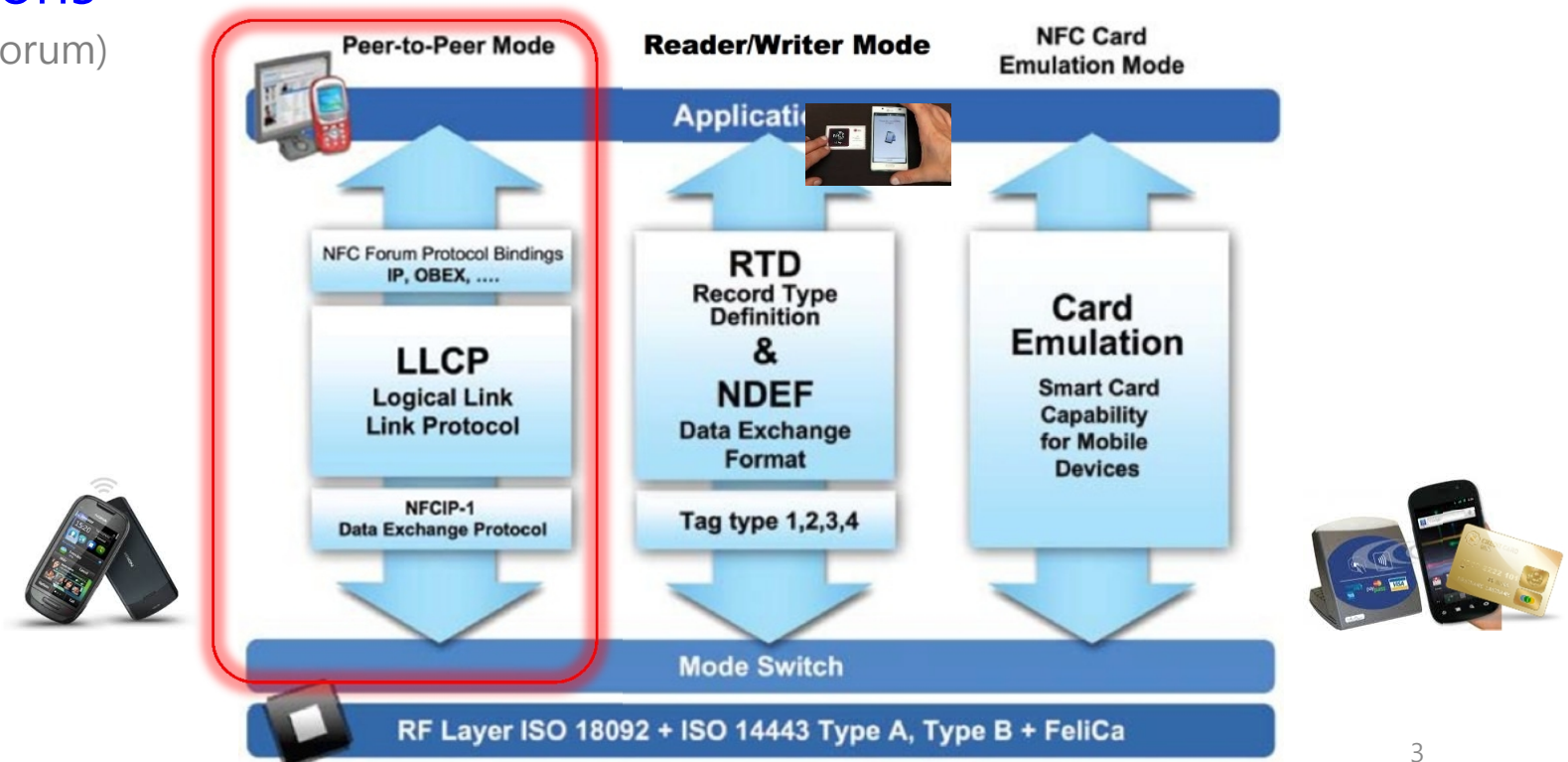
History and status

- Initial: draft-hong-6lo-ipv6-over-nfc-00 (July.7.2014)
 - Presented at IETF-90 6lo WG meeting
 - Positive feedback from Michael Richardson, Pascal Thubert, Suresh Krishnan
- Update 01: draft-hong-6lo-ipv6-over-nfc-01 (Aug.14.2014)
 - Suresh comment at IETF 90: It needs detail address mapping
 - Resolve: Add Uncast address mapping and multicast address mapping respectively
- Update 02 : draft-hong-6lo-ipv6-over-nfc-02 (Aug.26.2014)
 - Complete TBD parts
- Update 03 : draft-hong-6lo-ipv6-over-nfc-03 (Nov.27.2014)
 - Update NFC IID and multicast address mapping
- **WG document : draft-ietf-6lo-nfc-00** (Mar.3.2015)
 - Update Stateless address autoconfiguration (RFC7136)

What is Near Field Communication (NFC) ?

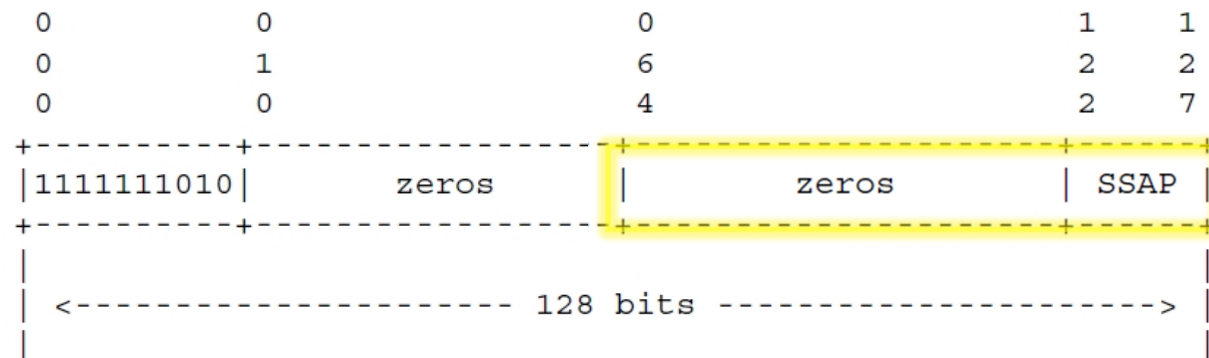
- NFC technology enables (Source: NFC forum)
 - simple and **safe two-way interactions** between electronic devices, allowing consumers to perform contactless transactions, access digital content, and connect electronic devices **with a single touch**.

- NFC Functions
(Source: NFC forum)



Update since IETF91 (1/2)

- Stateless address autoconfiguration
 - Uniqueness of NFC IID
 - NFC IID : 6-bit NFC LLCP address (SSAP or DSAP)
 - Data link connection : uniquely identified by (DSAP, SSAP)
 - Address configuration
 - 64-bit prefix + 64-bit IID(included 6-bit NFC IID)
 - 64-bit IID
 - 58-bit "0" + 6-bit NFC LLCP



Update since IETF91 (2/2)

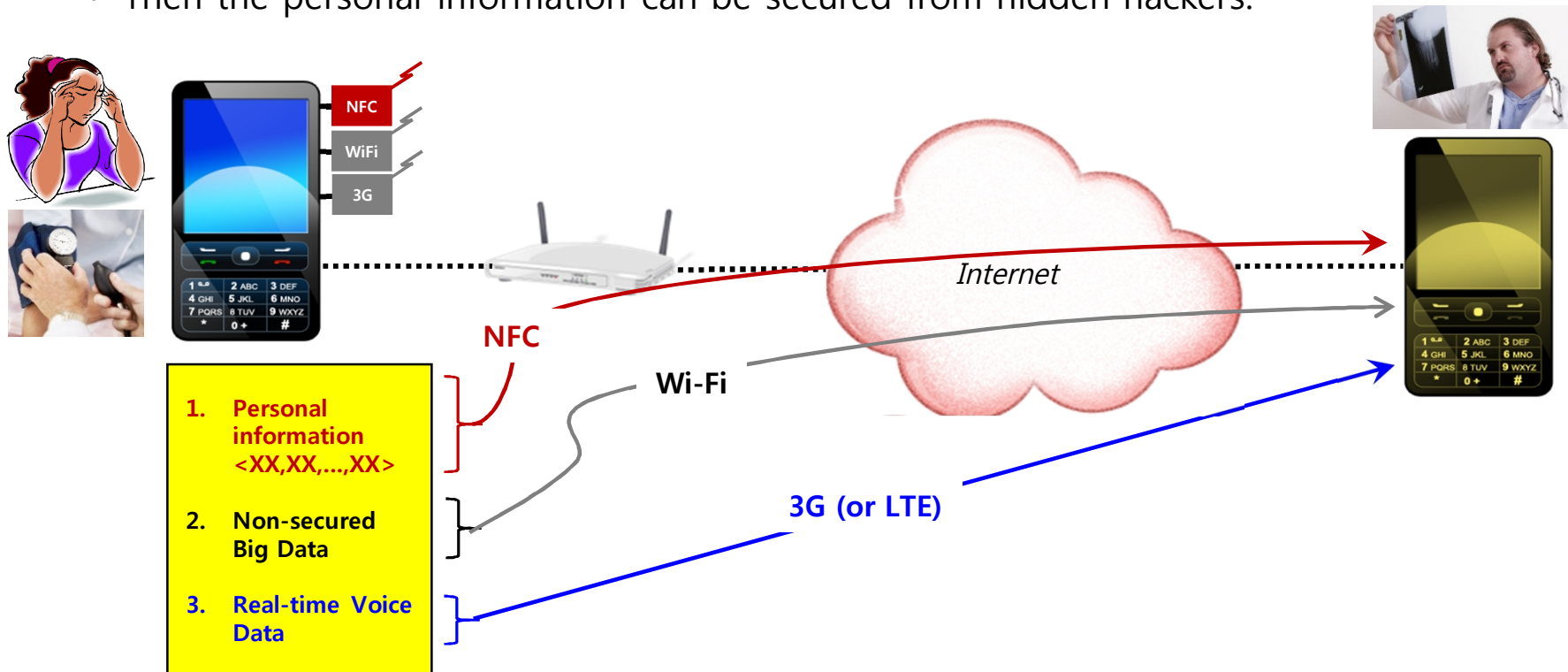
- Multicast address mapping
 - All IPv6 multicast packets -> NFC Destination Address, 0x3F (broadcast)
- 0x3F (broadcast) address
 - Must not be used as a unicast NFC address of SSAP or DSAP
- When represented as a 16-bit address in a compressed header
 - Formed by padding on the left with a zero

```
      0                               1
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+---+
| Padding(all zeros) | 1 1 1 1 1 1 |
+---+---+---+---+---+---+---+---+---+
```

With a Possible Example for NFC Use

- **Scenario**

- A User wants to securely provide his personal information (e.g., certificate or results of blood pressure measurement, etc.) to his doctor via an app. in a mobile phone.
- The app usually sends non-secured big data through WiFi, but it can securely send the personal information to his doctor by using only NFC.
- Then the personal information can be secured from hidden hackers.



Implementation of IPv6 over NFC

• TEST-BED

- ACR112U NFC Chipset & LG Optimus (NFC-enabled)
- Ubuntu Linux OS 12.4 (Kernel ver. 3.8)

• Current Status

- Header compression & Packet Fragmentation : done (but, verification required)
- **(new) NFC network interface driver** for IPv6 packet transmission : 30% done
- > **should be simplified (direct binding to IPv6-over-NFC layer)**
- **(new) IPv6 address configuration** for the NFC IF. : 50% done



```

opened device /dev/nfc0
ifconfig nfc0 inet 127.0.0.1 up
ifconfig nfc0 down
ifconfig nfc0 hw ether 12:33:44:55:66:77
ifconfig nfc0 up
route add -net 127.0.0.0 netmask 255.0.0.0 dev nfc0
ifconfig nfc0
  
```

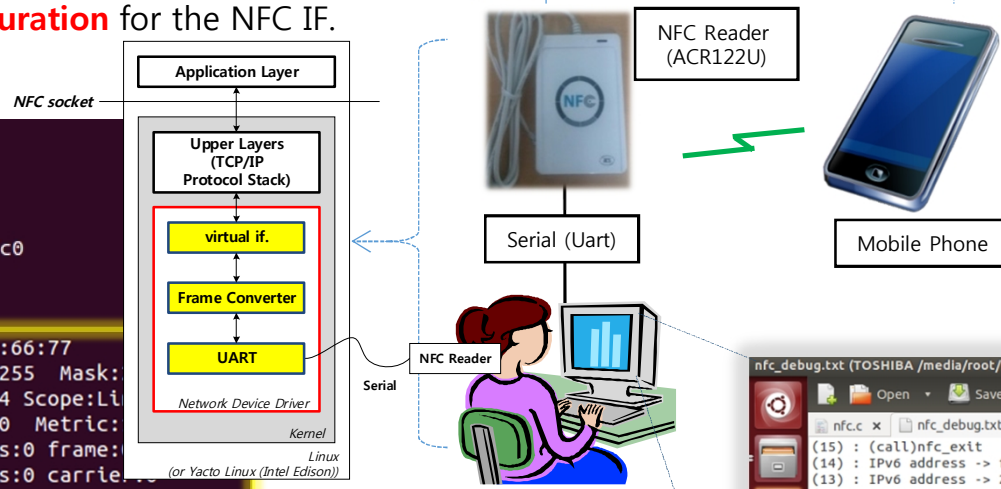
```

nfc0 Link encap:Ethernet HWaddr 12:33:44:55:66:77
inet addr:127.0.0.1 Bcast:127.255.255.255 Mask:
inet6 addr: fe80::1033:44ff:fe55:6677/64 Scope:Li
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:
RX packets:0 errors:0 dropped:0 overruns:0 frame:
TX packets:0 errors:0 dropped:0 overruns:0 carri
collisions:0 txqueuelen:500
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
  
```

```

user@instant-contiki:~$
user@instant-contiki:~$
user@instant-contiki:~$ ping6 -I nfc0 fe80::1
PING fe80::1(fe80::1) from fe80::1033:44ff:fe55:6677 nfc0: 56 data bytes

Recv Data[86]: 33, 33, FF, 0, 0, 1, 12, 33, 44, 55, 66, 77, 86, DD, 60, 0,
0, 0, 0, 20, 3A, FF, FE, 80, 0, 0, 0, 0, 0, 10, 33, 44, FF, FE, 55, 6
6, 77, FF, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, FF, 0, 0, 1, 87, 0,
5, 9C, 0, 0, 0, 0, FE, 80, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 1, 1, 12, 33, 44, 55, 66, 77,
  
```



network IF.
for the NFC Reader

```

nfc_debug.txt (TOSHIBA /media/root/TOSHIBA) - gedit
(15) : (call)nfc_exit -Wed Nov 5 21:06:52 2014
(14) : IPv6 address -> ff02::1::ff00:3 -Wed Nov 5 21:06:52 2014
(13) : IPv6 address -> 2002:9be6:1df1::9be6:1df1 -Wed Nov 5 21:06:52 2014
(12) : fe80::200:ff:fe00:3 -Wed Nov 5 21:06:52 2014
(11) : Header_Flag : 011000100110011 -Wed Nov 5 21:06:52 2014
(10) : (call)Header_compression -Wed Nov 5 21:06:52 2014
(9) : (call)nfc_init -Wed Nov 5 21:06:52 2014
(8) : (call)nfc_open -Wed Nov 5 21:06:52 2014
(7) : (call)register -Wed Nov 5 21:06:52 2014
(6) : (call)register -Wed Nov 5 21:06:52 2014
(5) : (call)register -Wed Nov 5 21:06:52 2014
(4) : (call)register -Wed Nov 5 21:06:52 2014
(3) : (call)register -Wed Nov 5 21:06:52 2014
(2) : (call)register -Wed Nov 5 21:06:52 2014
(1) : (call)register -Wed Nov 5 21:06:52 2014
(0) : (call)nfc_init -Wed Nov 5 21:06:52 2014
  
```

Next step

- **Updates for “draft-ietf-6lo-nfc-01”**
 - Considerations of Header Compression (Section 4.5) for NFC
 - Considerations of Fragmentation and Reassembly (Section 4.6) for NFC
 - Document re-organization for parts related to IPv6 Addressing
- **Implementations**
 - Verification for Header Compression & Packet Fragmentation
 - Simplification & Enhancement for NFC Network Device Driver