NDN-Android: NDN Networking Stack for Android Platform
What is the difference with IP?

Internet Protocol (RFC791): Focused on delivering packets to destination *node*

- Destination address
- Source address
- Meaningful static identifier

NDN: Focusing on retrieving *data*

- Interest contains desired name
- Data has unique name
- Data has to be secured
  - Sign Data with certificate
Example: enable a camera to publish data

- NDN protocol stack
  - NFD: for network connectivity
- Routing configuration
  - Discovery of local hub & prefixes
  - Local data prefixes propagation
- Identity/Certificate
  - Sign Data with the certificate corresponded to identity
NDN apk: NDN Stack for Android

- NDN Forwarding Daemon
- Security Tools
- Management GUI
- NDN Tools
- NDN Certificate Management
- WiFi Direct Communication
What’s Inside NDN.apk (integrate NFD into apk)

- **Platform-specific native-compiled NFD**
  - (-) No “nfd” binary
    - No main() function as a starting point
  - (+) A bundled shared library
    - No starting point

- **libnfd-wrapper.so** adds starting point to create NFD thread from Android Java code via JNI interface
  - Runs as Android Service
  - NFD management tools, other tools and UIs are implemented in Java

No need to root Android and can be deployed through Google Play store

Cannot have direct access to hardware
- TCP, UDP, WebSocket faces
Enabling NDN on Android

• Download and install NDN.apk from Google Play
  • https://play.google.com/store/apps/details?id=net.named_data.nfd
  • Or compile from source
    • https://github.com/named-data-mobile/NFD-android

• Start the app
• Start NFD

• + (Auto) Configure name reachability
• + Enable local or global NDN connectivity
• + (Auto) Configure security identities
Local NDN Communication

- Enables NDN communication over WiFi Direct
  - Faces and routes are configured and maintained automatically
Global NDN Communication

• Create Faces and configure routes automatically (in progress) or manually
  • Face and route configuration recover automatically when network is disconnected and reconnected

• Discover nearby NDN hub & maintain the connectivity automatically (working in progress)
Identity Management

• Every application should have corresponding identity (namespace) and the corresponding certificate for this namespace

• Applications could to manage sub-identities and their certificates (working in progress)
Currently in progress

- Create GUI versions of other NDN tools
- Implement NDN over Bluetooth
NDN Android Development

- Install NDN.apk from Google Play

- Use jNDN as part of standard Android app development process
  
  - Template: https://github.com/cawka/ndn-skeleton-apps/tree/master/jNDN