Connecting the edges
The UMOBILE project

Sotiris Diamantopoulos
Research assistant,
Athena Research and Innovation Center

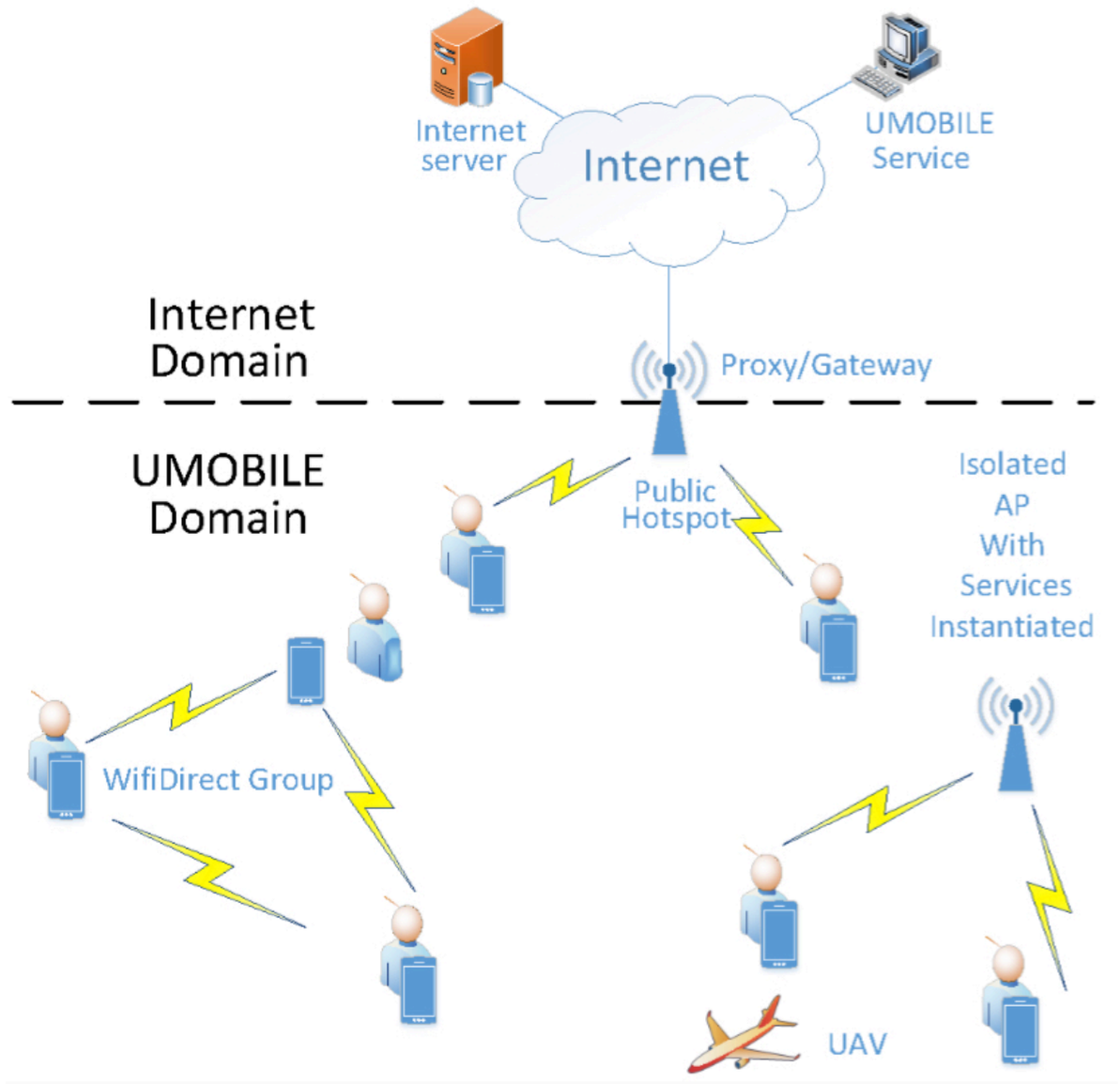
Main objectives

- Develop a consolidated information-centric, opportunistic and delay-tolerant communication platform
- Provide architectural support for the network edge, where mobility and connectivity disruptions are the norm
- Drive the Internet towards a communication platform for universal coverage
- Drive new application models and services

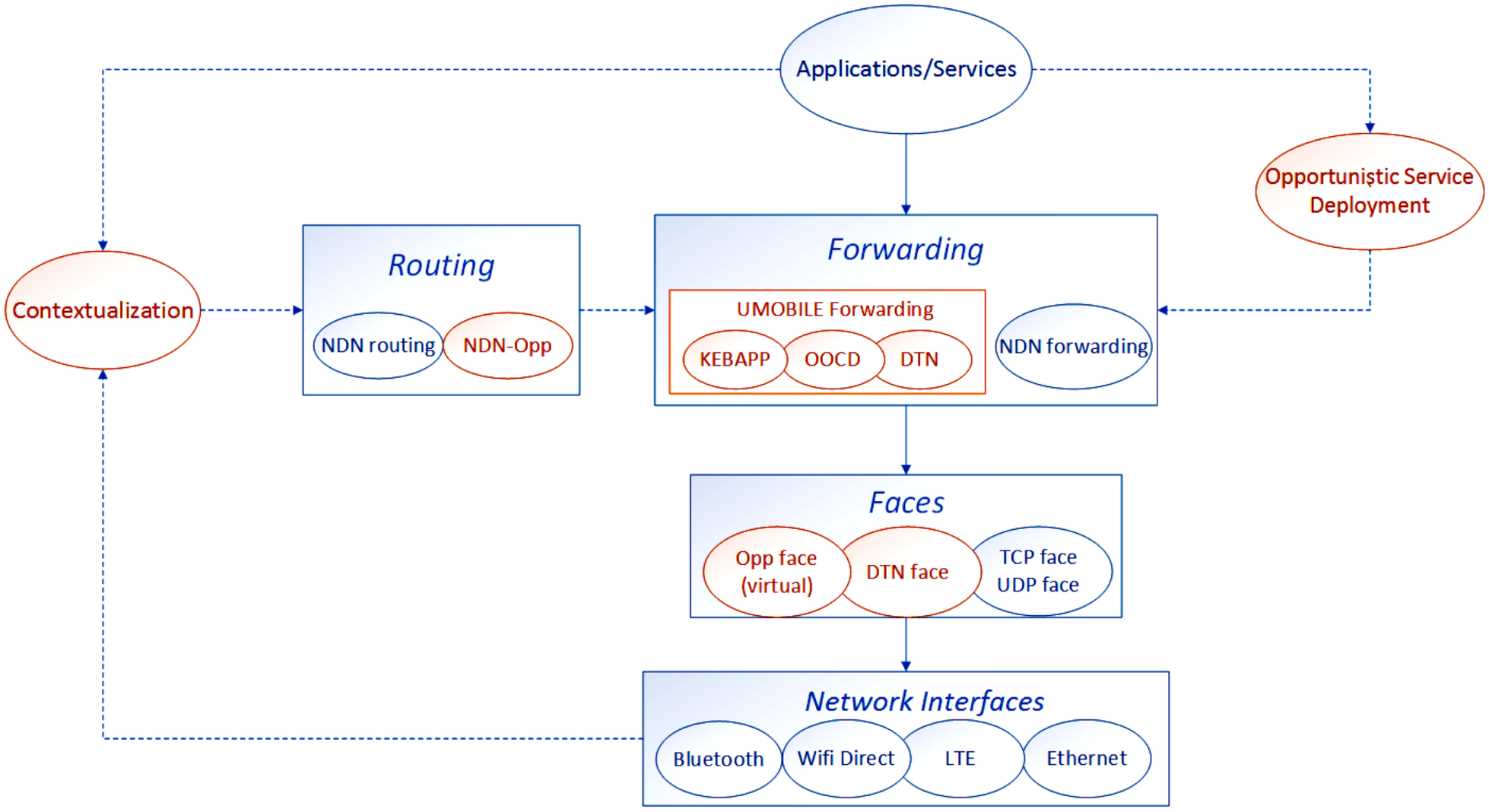
UMOBILE

High-level perspective and novelty

- Exploitation of all communication opportunities and better exploitation of network capacity
- Inherent support of disruptive communications, even between devices that are disconnected in space at any point in time
- Facilitation of user and service mobility
- Application/computation sharing
- User, usage and network contextualization
- Social-based routing



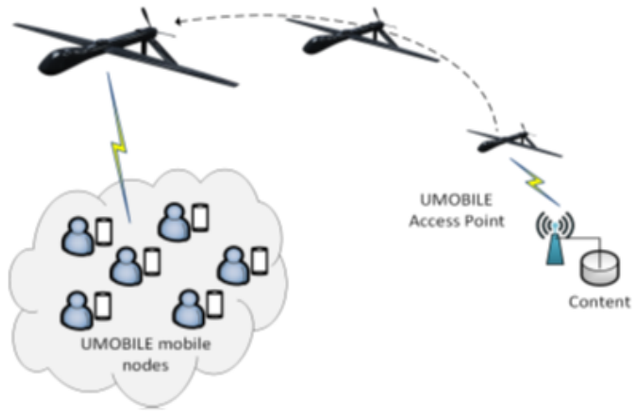
UMOBILE architecture



- Extending/modifying NDN for opportunistic and edge communications

Forwarding

- DTN tunneling
  - Reachability
  - Reliability



- Opportunistic off-path content discovery (OOCD)
  - Introduces a new table (D-FIB)
  - Points Interests towards the edge of the network, if Interests for same content recently received (=Data cached)
  - Cache hit increase through the discovery of locally available content

- NREP:
  - Introduces name-based push services with priorities (for disaster scenarios)
  - Messages spread through the network of mobile devices, based on their name, related priorities, TTL and the geographic area of dissemination

Routing

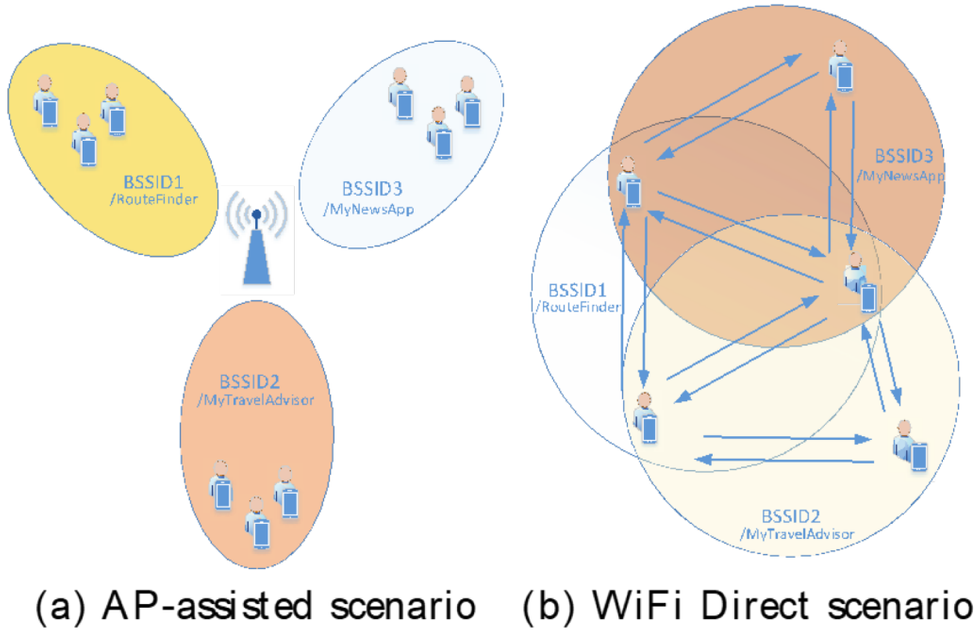
- NDN-Opp
  - Opportunistic communications (e.g. over Wi-Fi Direct)
- DABBER protocol (information-centric routing for opportunistic networks)

Contextualization

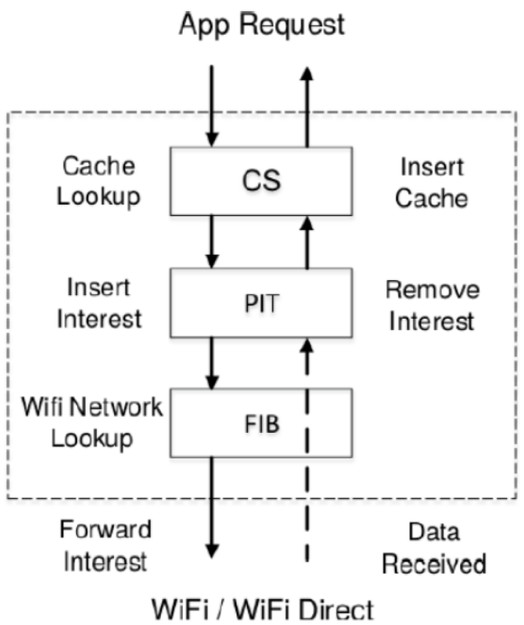
- Improves data dissemination through social awareness
- Passes information to other modules/apps

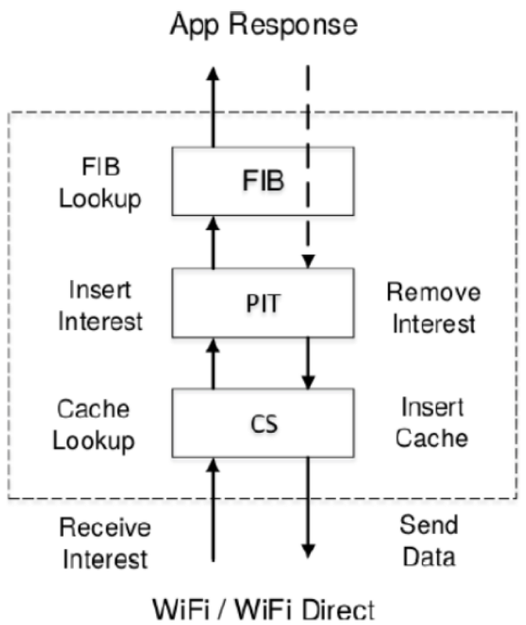
Northbound APIs

- Keyword-based mobile application sharing (KEBAPP)
- Application-centric computation and communication model
- Information discovery through application-driven and application-defined, hierarchical namespaces



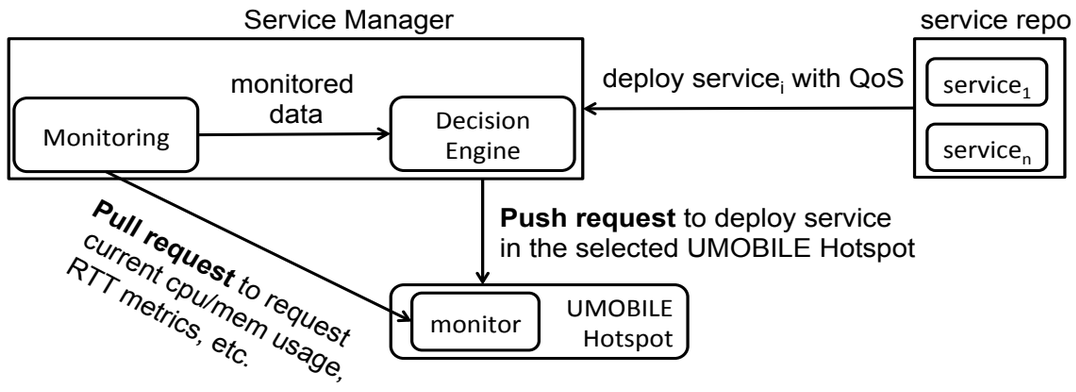
(a) AP-assisted scenario (b) WiFi Direct scenario

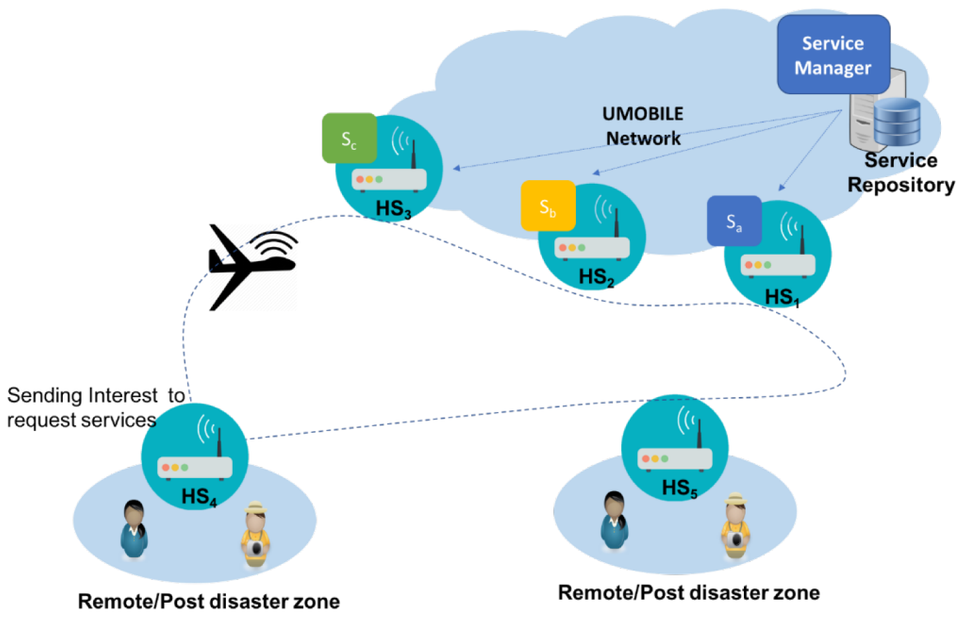




Edge service deployment

- Application-level, QoS mechanism to overcome latency and availability constraints
- UMOBILE hotspots
- Core network, isolated nodes

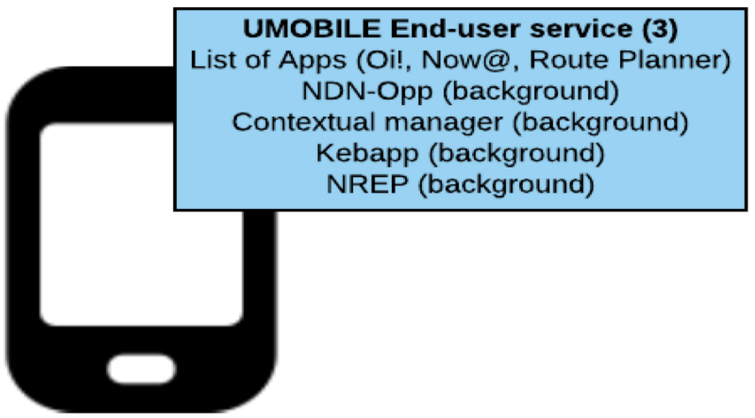


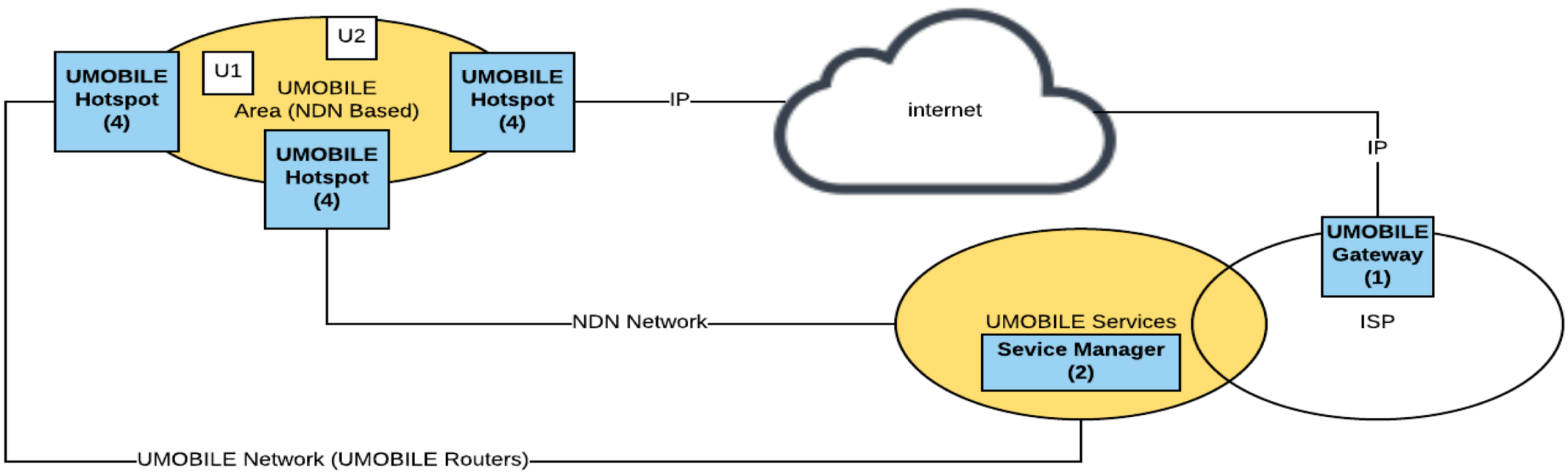


UMOBILE as a Whole

Main Elements, End-to-End Perspective

UMOBILE Gateway (1)
UMOBILE Service Manager (2)
UMOBILE End-user service (3)
UMOBILE Hotspot (4)





Questions?

- Thank you :)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

ATHENA Research & Innovation Information Technologies

UCL

UNIVERSITY OF CAMBRIDGE

copelabs Cognition and People centric Computing Labs

tecnalia Inspiring Business

tekever AUTONOMOUS SYSTEMS

Senceptior

fon

AFA SYSTEMS

Back-up Slides

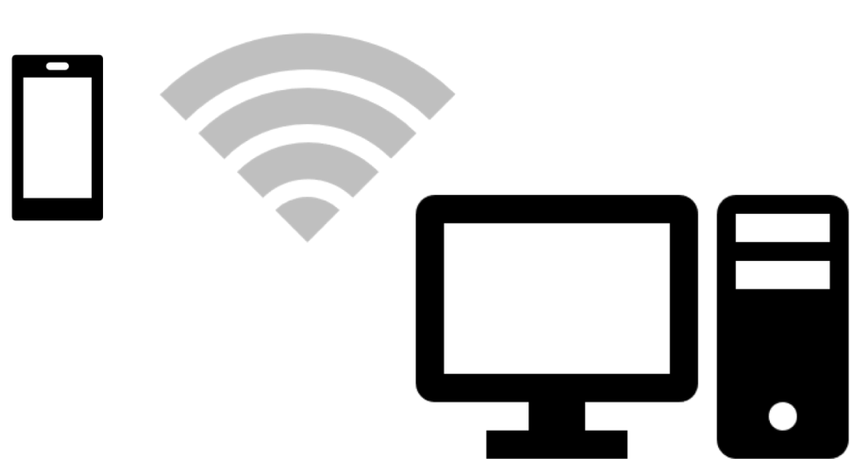
DTN tunneling

Problem statement

- NDN originally developed for low-delay, highly-connected network environments
- Intermittent connectivity: no support out-of-the-box
- Additional mechanisms are needed
- Breadcrumbs routing limitations

Breadcrumbs

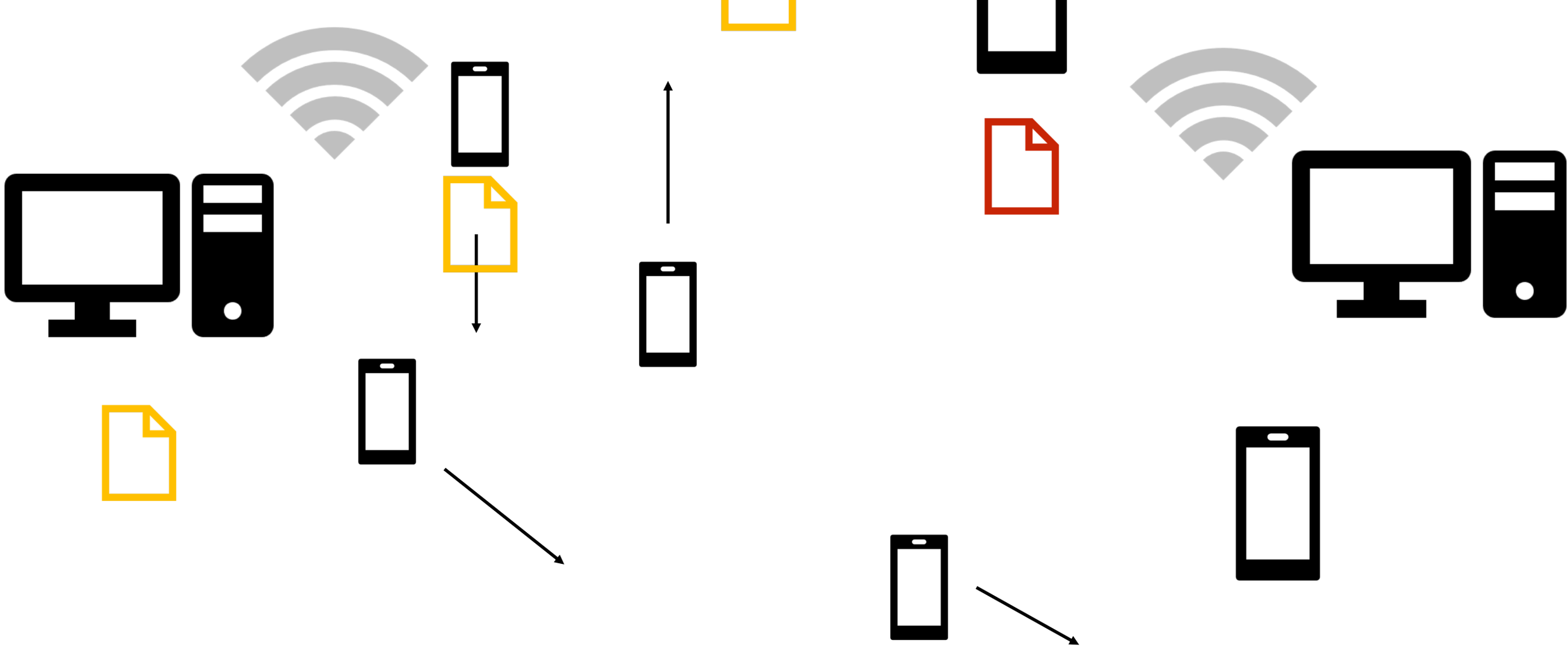
- End-to-end connectivity assumption (typical in wired networks)
- If not, upstream node is usually different than downstream



Opportunistic mobile networks

Interest 1

Data



Opportunistic mobile networks

Interest 1

Data

Problem statement

Goal: Enhance NDN to provide intermittent connectivity support

Solution 1:

- Try to build delay/disruption tolerant functionality into NDN

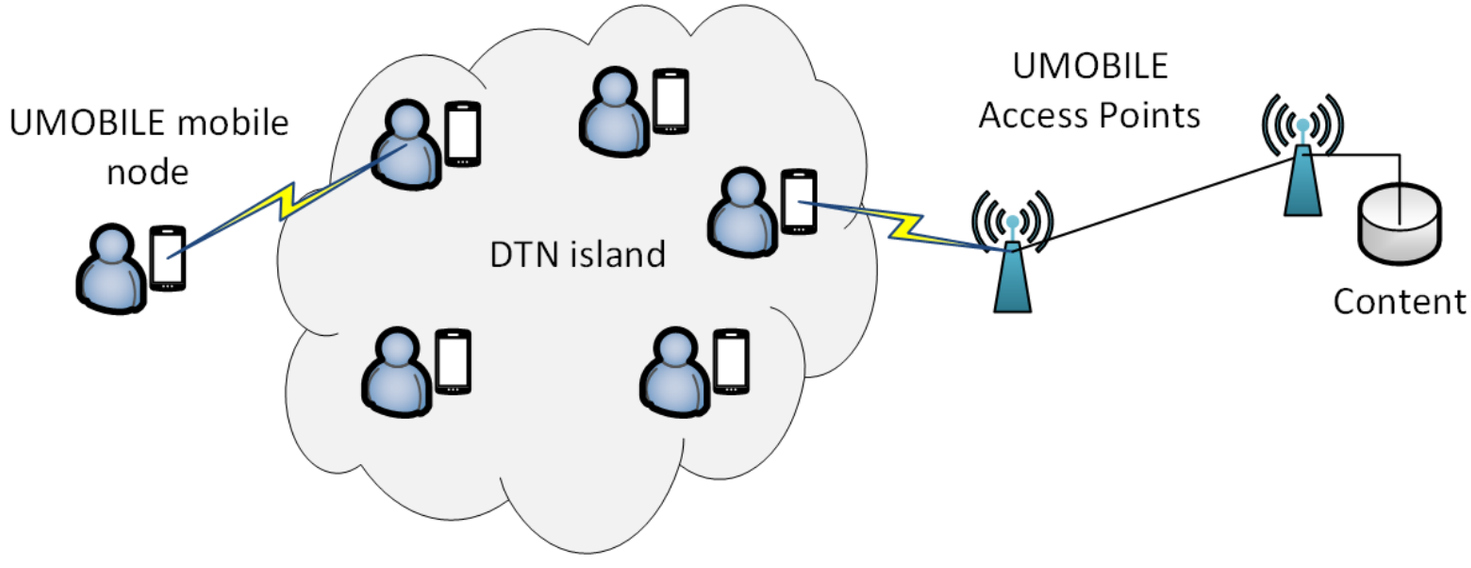
- Example:
  - Data packet is lost when transferred back. Consumer has to send interest again, but not efficient (very large delay until next contact).
- One solution is to have the data mule retransmit the Interest packet, assuming responsibility for content retrieval.
  - Just arrived at some form of the DTN custody transfer mechanism
- Could perform Interest flooding, polling or both – inefficiency problems again
- Forward Interests only to nodes most probable to deliver them/return the Data
  - After a while we will start mimicking DTN routing algorithms
- Accept unsolicited Data: altering NDN core communication primitives, possible security problems

Solution 2: Integrating DTN

- **First thoughts:** DTN an already well-mature technology, with several existing implementations
- **Idea:** Instead of implementing delay/disruption-tolerant functionality from scratch in NDN, leverage the existing implementations
- **Solution:** DTN as an NDN underlay
- **Practically:** Create a DTN face in NFD, in order to tunnel NDN packets through DTN islands (or isolated data mules)

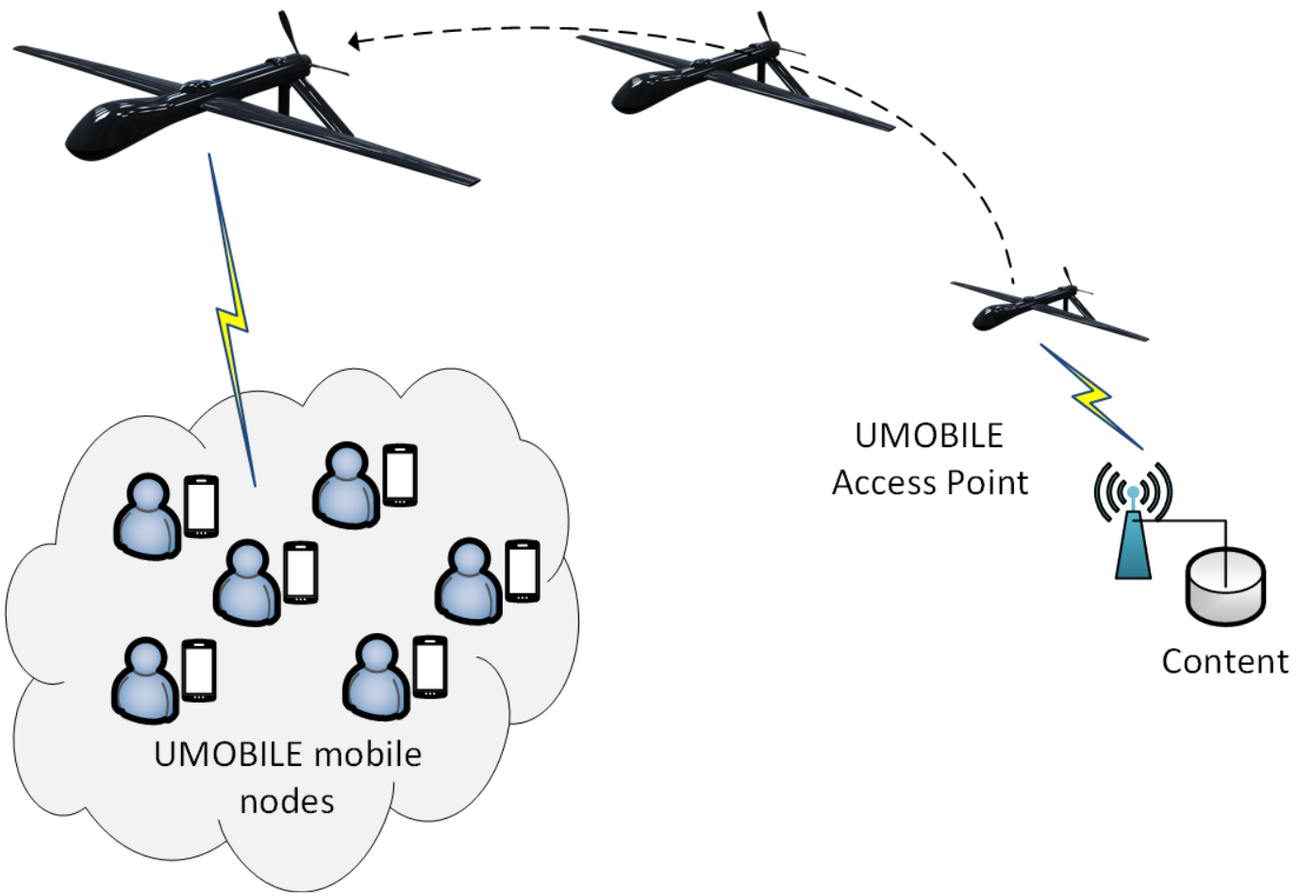
DTN can… (1)

(a) Enable opportunistic forwarding between ICN nodes



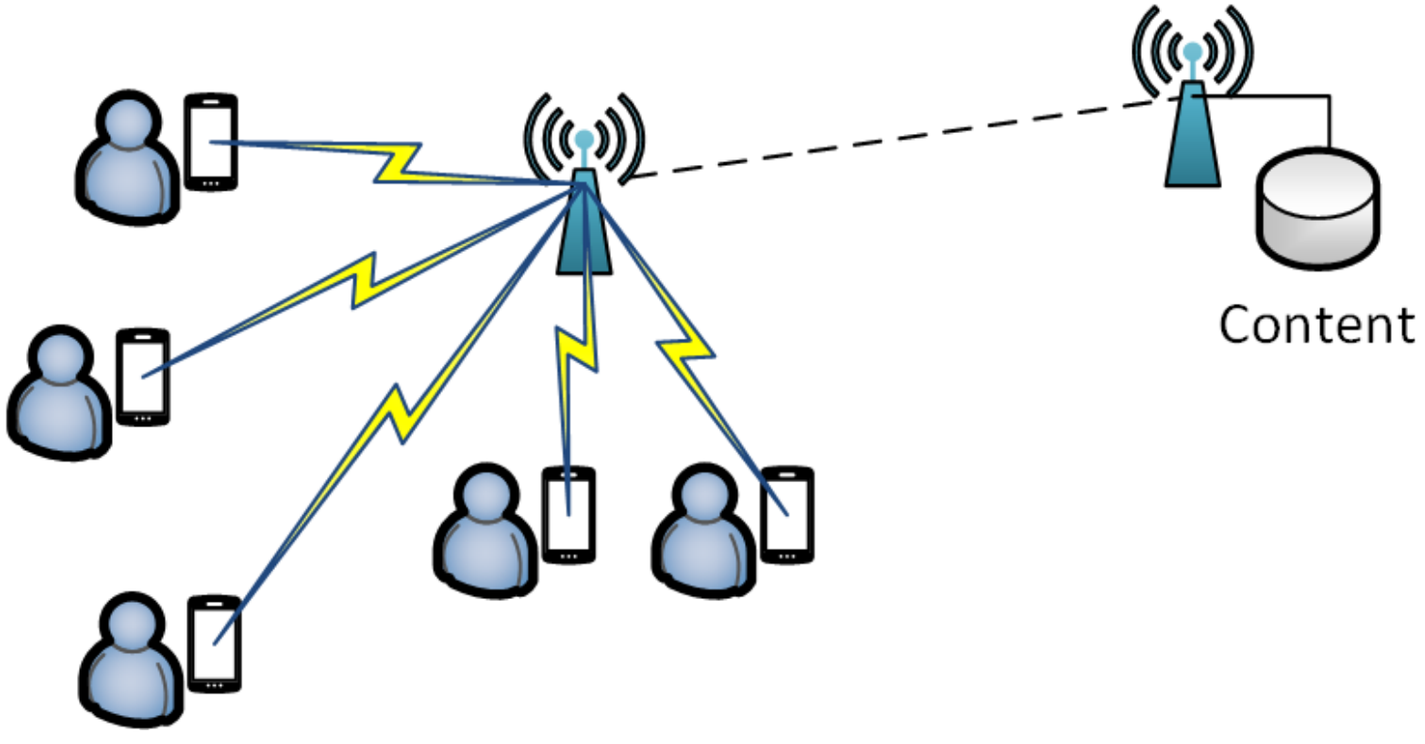
DTN can… (2)

(b) Accommodate delays



DTN can… (3)

(c) Increase reliability (e.g. by offloading cellular data to DTN)

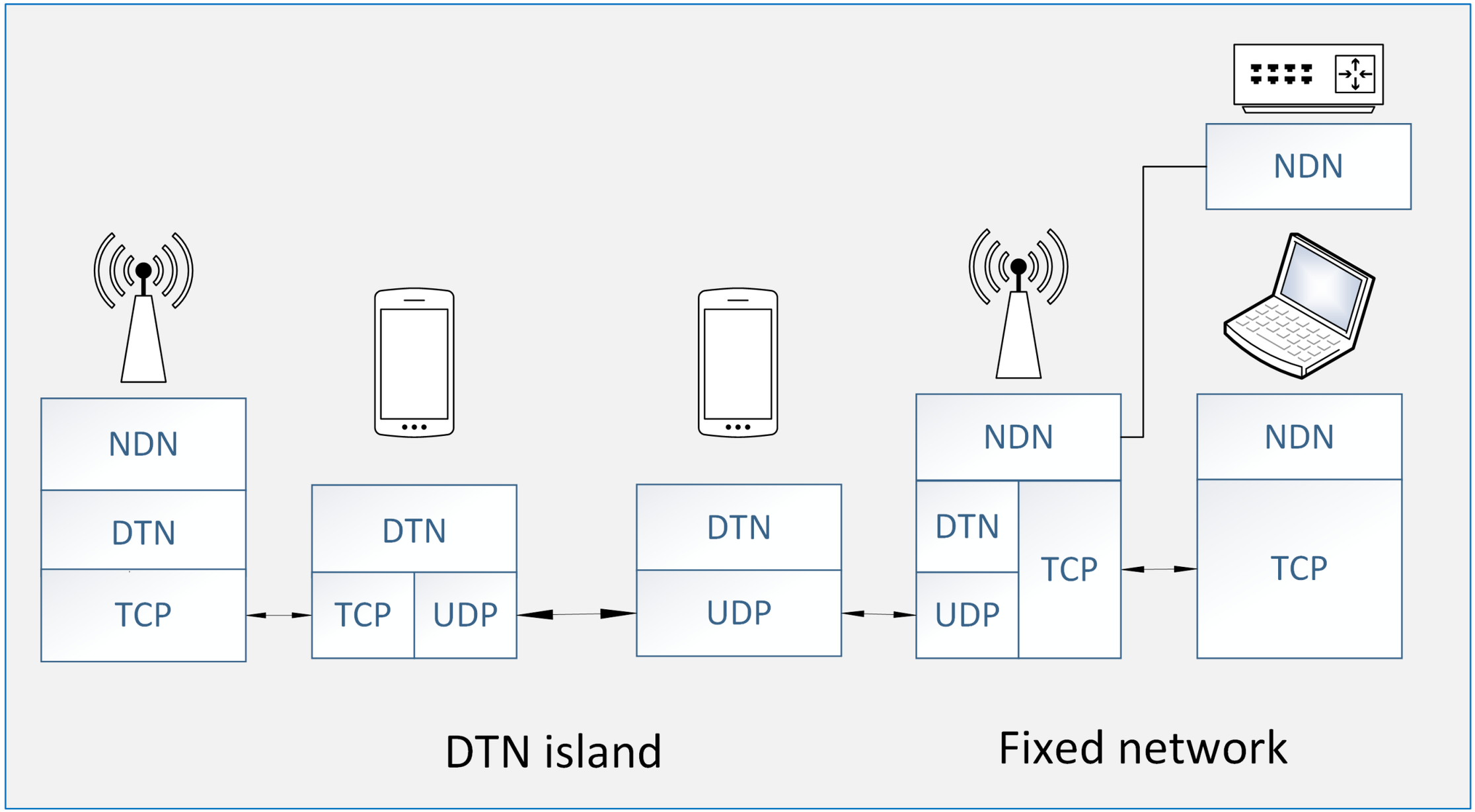


Approach tradeoffs

- On the plus side:
  - Compatibility with the original NDN architecture
  - Facilitates NDN deployment over existing DTN implementations
  - Intermittent connectivity handling is abstracted from NDN

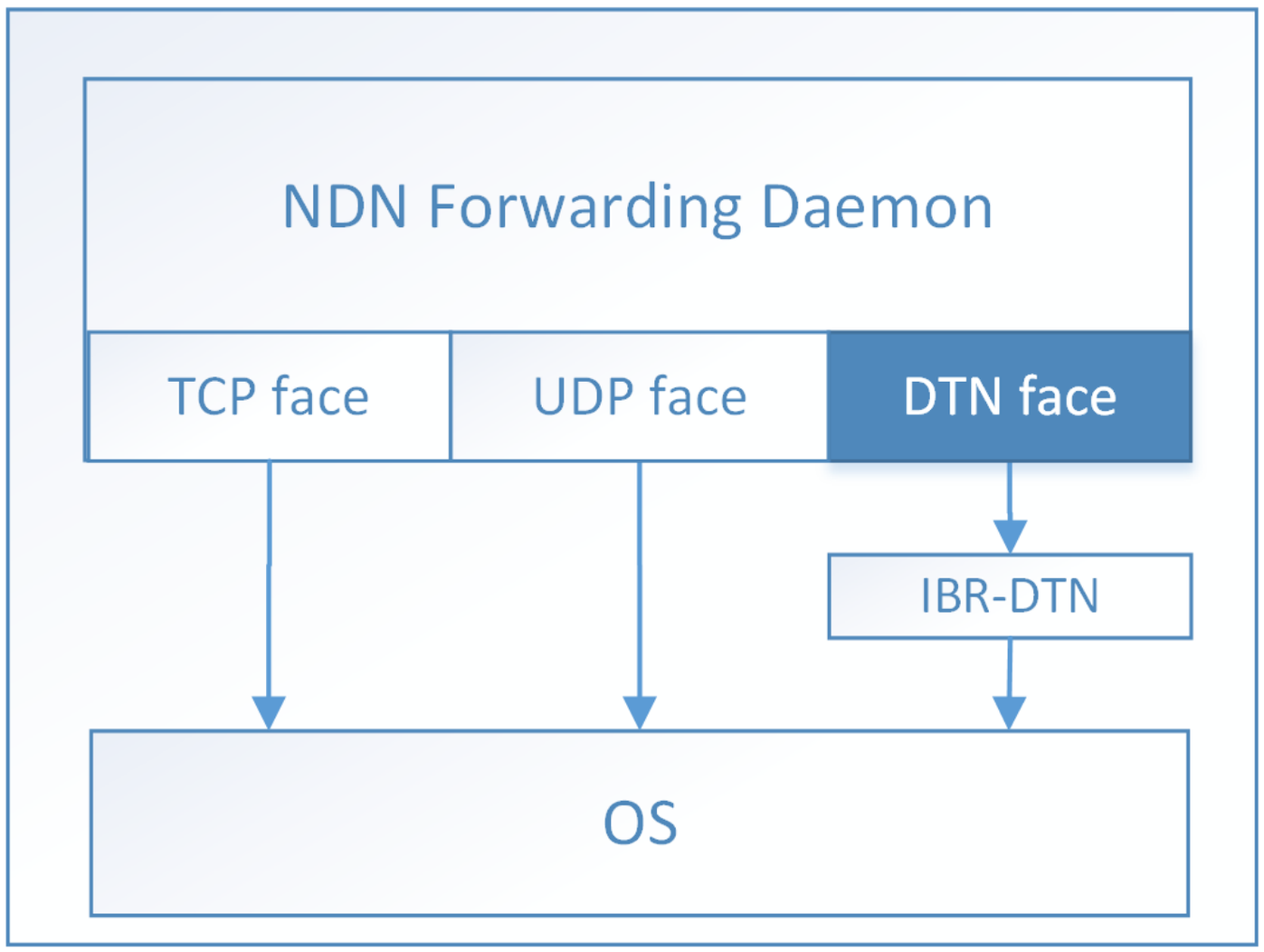
- On the minus side:
  - Not a native mechanism; NDN not delay tolerant by itself
  - Extra layers = overhead
  - Not a single extra layer: The bundle protocol also an overlay

Protocol stack



Implementation

- Linux/Android
- Used the IBR-DTN Bundle Protocol implementation as DTN transport
- Software stack:



Android implementation

- A closer look