Infrastructure Mode Based Opportunistic Networks on Android Devices

Andre Ippisch, Kalman Graffi
Technology of Social Networks Group
University of Düsseldorf, Germany
Email: {ippisch, graffi}@cs.uni-duesseldorf.de

***Abstract*—Opportunistic Networks are delay-tolerant mobile networks with intermittent node contacts in which data is transferred with the store-carry-forward principle. Owners of smartphones and smart objects form such networks due to their social behaviour. Opportunistic Networking can be used in remote areas with no access to the Internet, to establish communication after disasters, in emergency situations or to bypass censorship, but also in parallel to familiar networking. In this work, we create a mobile network application that connects Android devices over Wi-Fi, offers identification and encryption, and gathers information for routing in the network. The network application is constructed in such a way that third party applications can use the network application as network layer to send and receive data packets. We create secure and reliable connections while maintaining a high transmission speed, and with the gathered information about the network we offer knowledge for state of the art routing protocols. We conduct tests on connectivity, transmission range and speed, battery life and encryption speed and show a proof of concept for routing in the network.**

***Index Terms*—Opportunistic Networks, Delay Tolerant Networks, Routing, Smartphones, Smart Objects, Android**

I. Introduction

Nowadays we have a widespread adoption of feature-rich smartphones in society featuring powerful processors, high bandwidth communication possibilities and large storage space. Such smartphones can produce and store large files or file collections such as photo albums, videos and offline maps. Additionally, smart objects are conquering the market, including smartwatches, media players, TVs, eBook-Readers, printers, photo cameras or even cars. Those objects produce and store large files of their own or depend on large media files. Many usecases are not supported nowadays, for example the exchange of those large files between geographically close participants or between multiple smart devices owned by the same user. In general those files are uploaded into the cloud or shared with other users over the mobile network which can be time-consuming and results in additional costs due to limited bandwidth if the limit of data plans is exceeded. Mobile network infrastructure, however, can also be absent, simply in buildings or areas without reception but also long-term in developing countries without well-developed mobile networks or due to catastrophe situations. Additionally there can be intentional reasons like censorship, for example the caused lack of Internet connectivity as happened at the Hong Kong protests in 2014 or after the Istanbul attack in 2016. All these factors seek for a solution in which files of any size can be transferred in a secure, fast and straightforward fashion from smartphone to smartphone or between other smart objects.

With local high bandwidth communication like Wi-Fi all smartphone and smart object devices can communicate without mobile network infrastructure. The combination of smartphones, smart objects and their users evoke the principle of an Opportunistic Network (OppNet). An OppNet [1] is a delay-tolerant-network in which nodes can communicate with each other without having a route connecting them. The distribution of files among several users is possible with local communication, all functions to offer an easy and cost-efficient solution are already available in current smartphones. While some solutions for 1-to-1-communication and data exchange exist, those are complicated, non-automatic and not suitable for multi-hop communications. Non restrictive ad-hoc communication is not given on current smartphones and neither is out of the box data sharing over Wi-Fi, closest comes Wi-Fi Direct which can connect not only two smartphones but a group of smartphones as well but pairing is necessary. Files and other data that are gathered by smart objects are mostly distributed by Wi-Fi networks which makes a persisting connection and an infrastructure necessary at all time. With Opportunistic Networking the infrastructure can be replaced by a system in which data is collected when smart devices come close to each other.

In this work we seek to find a solution to the challenges that rise from the aforementioned demands. The application should run on off-the-shelf i.e. not rooted smartphones, perform all its operations in the background and without user interaction to connect and transfer data securely over large distances.

Therefore, we create a network application that connects Android smartphones over Wi-Fi, offers identification and encryption, and gathers information for routing in the network. By using Wi-Fi tethering hotspots we create secure and reliable connections while maintaining a high transmission speed, and with gathered information about the network we offer knowledge for state of the art routing protocols. Our work should be a solution to the following scenarios:

- With direct communication between two users there should be a direct connection between them and the transmission should be as fast as possible.

This work was partially supported by the BMWi Grant KF2846503KM4 - within the program Zentrales Innovationsprogramm Mittelstand (ZIM)

- With direct communication between several users, the content should be sent to at least one device, other devices get the data from the original sender or previous recipients.
- With multi-hop communication, in which one user wants to send data to some user that is not in range, other devices nearby act as data ferries to transport the message to the receiver.

We offer our work as an Android application, because Android offers an open-source mobile operating system that is the most widespread in the world with a 2016 market share of 86.8% [2] and with over one billion smartphones shipped. Many smart objects like the aforementioned are running with mobile operating systems as well, including Android.

A high cost-to-benefit-ratio is necessary for the user, for example many possible usecases for network usage which are provided both by self implemented and third party owned applications, like chat messengers or filesharing applications, and an API to connect the applications.

By running the application in the background, with no necessity for user interactions when it comes to connecting devices and exchanging data, and allowing the users to temporarily disable the network connection at all times, they stay in control of the device and are not distracted in their ordinary use of the device.

Also, for delivering data to devices that are not directly connected, a routing scheme is necessary, so that we evaluate OppNet routing protocols in current literature and combine advantages of several of those to provide routing that suits the given prerequisites and takes full advantage of modern smartphones.

To accomplish a marketable solution we not only design the OppNet application but also a usecase to provide a proof of concept. We put this into effect by implementing a filesharing application that can be used with the network, as this application covers several of the aforementioned aspects. We present an application with a functioning routing algorithm, the ability to exchange files with fast and secure transfer and show that user interaction is not necessary to fulfill the tasks.

This paper is structured as follows: First, in Section II we introduce related work to all following parts of the contribution which is divided into three parts. Link connection set-up and information gathering (Section III), providing collected information to be used in routing protocols (Section IV) and giving example applications for specific usecases (Section V). Concluding in Section VI, we evaluate the link connection layer and give results regarding the routing possibilities.

II. Related Work

OppNets are characterised by local, range-limited and mostly wireless communication of nodes and are described in detail in [3]. The movement of users in wireless networks causes frequent network partitioning and dynamic connection opportunities. In order to bridge connections between separated partitions, mobile helper nodes, so called data ferries, have been proposed as in [4]. A message ferry is a controlled node, which follows a store and forward routing approach to transport messages between geographical areas, in our case the owner of the smartphone acts as a data ferry.

Our work focuses on the link connection set-up which is achieved with infrastructure mode Wi-Fi connections. The information we gather from connecting and exchanging meta information can be used for routing. Based on connection and routing the whole network can be used for applications.

A. Link Connection Setup

Other applications like TeamPhone [5] use rooted Android phones, which are not actually off-the-shelf, to enable ad-hoc communication. Since we want to use the network on unrooted off-the-shelf devices our challenge was to find a different working solution which we found in the connections over Wi-Fi tethering hotspots. The link connection setup of our network application is similar to the more theoretical contribution of Wlan-Opp [6], which uses infrastructure mode Wi-Fi, too. We concentrate more on the practical implementation on Android.

B. Routing

In current literature there are various routing protocols suitable for our Android based OppNet application. These can be divided into different categories which include context free based, mobility based, and social context based routing protocol classes.

Context free routing is a routing principle in which peers have no information about other peers and their context to deliver a message. Ideally, low transmission delays and high delivery ratios can be obtained through flooding-based approaches. One controlled flooding algorithm is Spray & Wait [7] in which only the sender can create copies of the original message and connected peers forward the copies to the destination. A couple of extensions to this algorithm are Spray & Focus and Seek & Focus [8].

Mobility based routing is a principle in which the mobility and connectivity patterns are shared among peers to enable a targeted delivery of messages which improves routing in OppNets [9]. Notable routing schemes are PRoPHET [10] in which peers forward messages according to delivery probability, and Meeting and Visits [11] which additionally shares visits to geographical locations.

Social context based protocols use the social aspects of the peers to route through the network, additional context information is used to predict the delivery probability. Some approaches include that social activities and also geographical structures motivate the mobility of people, who are considered a target area in the field of social context based routing. Examples are dLife [12] and SPRINT [13] which are advanced approaches that use a distributed community detection algorithm like k-CLIQUE [14] to form communities.

C. Applications

Application areas for OppNets are typically found in sparse environments. Examples are the Sami Network Connectivity

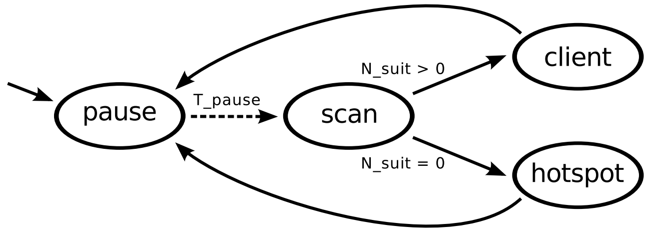


Fig. 1. The four stages of the connection process

(SNC) Project [15] in Northern Europe, which aims at providing Internet access for Sami population of reindeer herders who live in remote areas, or the project Zebranet [16] in Africa, which aims at tracking zebras using an OppNet. Disaster Recovery applications exist for rooted Android devices, the aforementioned TeamPhone is one example. However, if the application is not usable on unrooted off-the-shelf smartphones and smart objects, its extensive distribution is questionable.

III. Layer 1: Link Connection

A. Goal & Idea

For the connection between Android devices we want to be able to automatically connect to other devices in the surroundings without user interaction to forward data. The implemented ad-hoc standards which are offered by Android devices do not fulfill our demands as Wi-Fi Direct and Bluetooth offer no API to accept connection requests programmatically. Since the Android operating system does not ship with suitable ad-hoc functionality we enable a Wi-Fi tethering hotspot on Android devices, which serves as an access point to other devices which can connect to it. Advantages of using a Wi-Fi based connection technology are long range transmission and the use of the TCP protocol for reliability.

B. Connectivity

Our network application manages the network connections and represents the layer directly on top of the transport layer. All functions are executed by a background service that includes functionality for initialisation and management of connections, both inter-device and inter-application connections, for file-handling and exchanging information about the network. With the Android operating system we can open Wi-Fi tethering hotspots, scan for Wi-Fi access points, and request Wi-Fi state information automatically and without user interaction. In our network two devices can connect if one device opens a Wi-Fi tethering hotspot and another device connects to it. The network application uses a sequence of stages for the connection process as seen as in Figure 1.

In the *SCAN* stage the network application scans its immediate vicinity for available Wi-Fi hotspots and sorts out all Wi-Fi access points that do not belong to the network. The protocol takes previous connections, signal strength and routing information of the remaining hotspots into consideration to calculate if suitable hotspots are available. According to the connection protocol a hotspot is considered suitable if the buffer contains data packets that would be routed over the hotspot or if the hotspot was not connected recently, signal strength can prioritize hotspots over others. If at least one suitable hotspot is available, the *CLIENT* stage is chosen with the most suitable hotspot. If no suitable hotspot is found, the *SCAN* stage can be repeated or the *HOTSPOT* stage is chosen. In *CLIENT* stage the device connects to the most suitable hotspot found in the *scan* stage. In *HOTSPOT* stage a hotspot is created by the network application to which other devices can connect. The hotspot is at least open for a elected span of time and stays open as long as there are devices connected to it and data is transferred. The *PAUSE* stage might be entered manually or automatically after client or *HOTSPOT* stage to give the device the possibility to connect to another Wi-Fi network or to the mobile network.

Even if devices, for example smart objects like Android OS cameras, can not open tethering hotspots, these smart objects can still connect to a device that hosts a tethering hotspot.

C. Identification of Devices & Security

When installing the network application for the first time a public-private-key pair is created for identification, authentication and encryption inside the network. The fingerprint of the public key is broadcast as part of the SSID when hosting a tethering hotspot so that other devices can identify the device even before connecting to it. To protect the transferred data from non-participants the devices use challenge-response authentication to exchange public keys for point-to-point encryption when connecting to each other. For end-to-end encryption the public keys of network participants can be used by third party applications. The network application offers to exchange public keys in form of QR code scanning to enable a secure exchange of keys for end-to-end-encryption.

We use timing-attack resistant XSalsa20 stream cipher with a 192 bit nonce [17] for encryption in general, the Elliptic Curve Diffie-Hellman (ECDH) on Curve25519 [18] for key exchange, and Poly1305 [19] for authentication.

The performance of these stream ciphers is important for the speed of transmissions between devices, the encryption of the stream has to be faster than the maximum transmission speed at any range to ensure that encryption does not slow down the transmission.

D. Partition of Tasks and Responsibilities

Since we want to make many different applications use our network the network application is located on the layers below the application layer. The network application uses TCP on the transport layer, and IP and MAC for direct connections between two devices, it includes routing and transport overlays to act as link between third party applications and the wireless physical layer, also it offers an API to let other applications serve as a real application layer. The network application manages all tasks beneath the application layer like routing, maintaining the connection or transmitting the data.

Table I presents the protocol stack in which network and API application are divided into two separate layers with regard to our model. So, for Device B the third party application is unnecessary for forwarding the data to the next device.

TABLE I
PROTOCOL STACK

OSI/our model	Device A	Device B	Device C
Application	API app		API app
Application/Transport	network app	network app	network app
Transport	TCP	TCP	TCP
Network	IP	IP	IP
Data Link	802.11 MAC	802.11 MAC	802.11 MAC
Physical	802.11 PHY	802.11 PHY	802.11 PHY

E. Offers

From current literature there are theoretical and practical contributions regarding the OppNet itself and many theoretical contributions examining routing protocols in OppNets and OppNet applications. We want to show that with a real life application that creates a real OppNet we can gather necessary information for current theoretical contributions.

What our network application is offering:

- public and private key creations and distribution for identification of devices before connecting, i.e. for routing decisions, while connecting, and for the transmission itself, therefore point-to-point and end-to-end encryption and signatures
- authentication with the GUI and decentralised web of trust that can be used by routing protocols
- exchange of connection, social and geographical information that can be used by various routing protocols
- internal information, such as battery life, geographical and mobility information (accelerator values)
- usage of signal strength and additional information gathered by Bluetooth for optimised choice of next connection, i.e. if devices are nearby that cannot open a hotspot due to restrictions the roles of hotspot and client devices can be allocated before the actual connection
- usage of end-to-end encryption and signature only when sending, therefore preventing unnecessary copies of messages and files on the device
- a datagram structure which is used as a container for raw application data that contains meta information like Time-To-Live and the path through the network

The contribution Wlan-Opp [6] shows some connection possibilities but not the big picture of application oriented connections and routing, so we change the link connection set-up by providing the following improvements.

- our scan stage takes more information for access point choice into consideration, the network application repeats the scan phase more often to save battery
- an additional pause stage to allow alternative connections (for the case of available mobile network or Wi-Fi infrastructure)
- with additional Bluetooth connections even devices that are both in the hotspot stage can communicate with each other to arrange themselves

In conclusion our OppNet layer utilises existing and well defined mechanisms already included in Android. This allows us to build a reliable and functioning method to connect devices with each other in an Opportunistic Network.

IV. LAYER 2: ROUTING

While in mobile ad-hoc networks a disconnect is considered rare and data can be sent over several hops immediately, disconnects are highly usual in OppNets, therefore reliable routing is not possible because the nodes might never be connected to each other at the same time. We look at the characteristics of smartphones and OppNets to find a solution for routing in an Android-based Opportunistic Network.

Smartphones come with huge storage space, are equipped to track time and location like GPS and cell tower localisation and most smartphone users tend to be in social communities, thus usually there is software installed that can be used for gathering information for social context. Using this information, such as we proposed in [20] and [21], allows to come to optimized forwarding decisions based on the current network situation. For gathering mobility information we use the connections between devices to exchange meta information about the network. The content of the meta information is not explicitly defined since we focused on the link connection setup and want to present the additional information that can be obtained by the devices' connections.

For our network application we want to keep all options open regarding routing protocols and focus on gathering information both from the device itself and from the connections with other devices. For the purpose of testing and giving a proof of work we use a routing principle which is a hybrid of some basic routing principles for OppNets.

For the purpose of spreading messages in general we use Spray and Focus [8] with a Time-To-Live value that is given with each data packet. The *History of Encounters and Transitivity* of PRoPHET [10] and the *Meetings* principle of HiBOp [22] are used to forward the data packet according to higher probability and in addition some social context information of the device owners can be saved and used to allow targeted distribution of data packets and to calculate probabilities.

Also, GPS and location pattern information, if available, are used for connection decisions, for example by location based rules for hotspotting which can help saving battery. Those information are used by routing protocols, too, for example when trying to carry data packets from one community to another community like mentioned in dLife [12] and SPRINT [13].

A. Exchange of Meta Information

In this Section we want to present the routine of meta information exchange between devices when they connect to each other. The connections between devices are used not only for data packet transmission but also for meta information exchange that can be used by routing protocols to target data packets in the right direction. Figure 2 shows a scenario in which five devices are lined up in a row and only the movement of devices B and D enables an OppNet for those devices and meta information is exchanged with a two-hop-count. For example, the meta information `CDB` at device A includes all information that device C has gathered about its connections to device D. This information packet was obtained

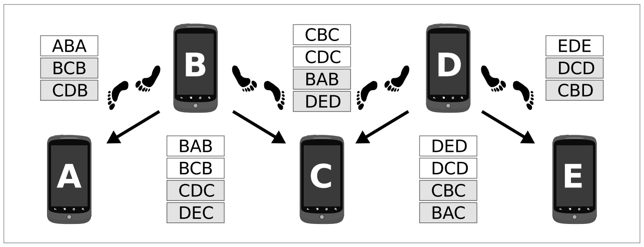


Fig. 2. Meta information exchange

by device B and forwarded to device A, where Device A can use it to calculate that through device B there is a high probability to reach devices C and D.

With the information that can be gathered by our network application we provide knowledge about the device's state and the network and neighbouring nodes. We use principles of established routing protocols to give a proof of concept and to use basic routing in our network. However we do not want to compete with state of the art routing protocols but instead want to motivate the usage of those in our network, since we show that most information that is necessary is available in our network. Therefore, existing routing protocols can be easily adapted in our Android based OppNet.

V. Layer 3: Applications

We want to offer an application-driven Opportunistic Network and one important part are the applications that can use the network. The network application provides an API that can be used by third party applications for their network usage. The API defines a datagram structure for the network to be used by other applications for sending and receiving data in the form of data packets that are forwarded through the network.

To provide a proof of concept and some initial ideas for applications, we implemented usecases in the form of a filesharing and a chat application. OppNets on Android are not recommended for time critical applications, but if no infrastructure is given, not the time is the important factor but the fact that messages are received by the recipient at all, also, when distributing large files, limited data plans make it costly to use the mobile data network. If we are reliant to the OppNet because of remote areas, emergency situations or censorship we are still able to provide networking even for applications that are normally time critical.

The OppNet based filesharing application, for example, is able to send large files to other devices and offers a distribution function for requesting files from other devices. Data packets can both be broadcast, and routed to a specific receiver. This receiver is identified by its public key which can be requested from the network application over the API. Those public keys, once obtained, can be mapped with contacts for further use, authentication can be done by manually verifying the fingerprint. Further options include the presetting of a routing protocol and the choice of an encryption method. Symmetric and asymmetric encryption is provided, asymmetric encryption uses the public key of the receiver, for symmetric encryption a passphrase can be chosen.

TABLE II
All Android devices for testing in detail

Alias	Name	Model	Manufacturer	Type	Cellular	Version	CPU	Memory
$\mathbf{N}_1$, $\mathbf{N}_2$	Galaxy Nexus	GT-I9250	Samsung	Phone	yes	4.3	1.2 GHz 2c ARM	1 GB
$\mathbf{O}_1$, $\mathbf{O}_2$	One	A0001	OnePlus	Phone	yes	4.4.4	2.5 GHz 4c Krait	3 GB
P	Google Nexus 4	E960	LG	Phone	yes	5.0.1	1.5 GHz 4c Krait	2 GB
Q	Google Nexus 7	1A019A	Asus	Tablet	no	4.4.4	1.5 GHz 4c Krait	2 GB
S	Google Nexus 9	OP82200	HTC	Tablet	yes	5.1.1	2.3 GHz 2c Denver	2 GB

There are many more possible applications that can be used with OppNets, in Section II we presented some existing ones.

VI. Results & Evaluation

In the following Section we present and discuss the results we gathered from testing our application.

One of the goals of our user- and application-driven Opportunistic Network is the adaption by smartphone users. For an easy adaption users must not be detracted from their familiar usage of their smartphone. The application is running without user interaction in the background. The application uses Wi-Fi for its connections, the transmission range depends on the signal strength of the device and on the interference around the devices. A large transmission range between the devices promotes high probability of a connection and successful data transmission. The application also offers security in the form of symmetric and asymmetric encryption for point-to-point and end-to-end encryption. Although the disk space on Android devices is normally high it still has a limit. All third party applications encrypt their data packets on the fly while transmitting it to the next device to avoid multiple copies of data on the device. We measure the encryption and decryption speed on multiple devices and compare it to the transmission speed of the network connection to see if the encryption speed is a bottleneck for the transmission. We also test if those connections work as desired and how meta information, that can be used as routing information, is exchanged.

Most typical scenarios for the usage of our network, which include mobile, static and mixed environments, depend on connectivity, routing, battery life, and transmission range and speed. We used the Android devices listed in Table II to test the dependencies presented in the following Sections.

A. Battery Life

Smartphone users do not want to be detracted from the familiar usage of their smartphone. Our application is running in the background and without user interaction even when the screen is turned off which already saves battery by default. We measure how much energy is consumed when using the network application which mostly uses Wi-Fi either when hosting a hotspot or searching for access points nearby.

The most energy is consumed in the phase in which the tethering hotspot is enabled because in addition to the transmissions the device is sending out beacons. We test the battery life span of the devices with our running network application and no other applications running except for default system applications.

Additionally we test the life span of the same devices with a running music streaming player, which is similar in several

TABLE III
BATTERY LIFE SPAN

Device	network app	music streaming app	without apps
N	27.05h	7.61h	228h
P	22.25h	9.37h	196h
S	47.01h	15.06h	400h

aspects since it uses Wi-Fi for streaming, has no need for user interaction, and runs in the background and with the screen turned off.

Table III lists the battery life span of the test devices. We can see that the battery life span of our test devices when running our network application is at least twice as long as with a running music streaming application. The battery life span still outreaches the wake time of typical users, who charge their phone at least once a day, therefore our application only slightly affects the charging habits of the users.

B. Transmission Range and Speed

The transmission range of the smartphones that use our network application depends on the signal strength of the device itself and on the interference around the devices. A large transmission range between the devices promotes high probability of a connection and successful data transmissions.

Dependent on different distances, we measure the transmission range and speed of multiple pairs of devices to still be able to maintain a connection and exchange handshake, meta information and a fixed sized data package of five megabytes.

In two different scenarios, once the measurement took place outside at an open area without Wi-Fi access points nearby and therefore almost no interference to create an almost ideal scenario, and once in the university building within hallways and with walls between the devices as well as some Wi-Fi access points nearby to create a building scenario.

We test with the devices **N**, **O** and **S** as the hosts of the tethering hotspot and use three devices as client devices.

Figure 3 shows the successful transmissions and their data rates for different ranges and combinations of test devices both for outside and inside measurements.

The difference of the maximum ranges between different devices takes their origin in the choice of manufacturers to keep the transmission range low to save energy and battery. This is because the common usecase for tethering is the forwarding of Internet connectivity for which users usually need only a distance of a few meters. Nevertheless both outside and inside there is enough range for the devices to connect to other devices in the surroundings. Even with a higher distance there is still a modest data rate that allows devices to exchange handshakes and even small files in a short time.

The results are promising due to the fact that devices can be connected even at high distance and the data ferry principle can be used to carry data packets in areas where members of communities are nearby. Inside the area of communities the distance between devices can be over 30 meters and therefore data can be transmitted directly or over multiple hops in short time.

C. Encryption and Decryption Speed

The application offers security in the form of asymmetric encryption for point-to-point and end-to-end transmissions. The encryption takes place when the transmission is about to begin to avoid using too much disk space.

We measure the encryption and decryption speed on multiple devices and with different file sizes and compare it to the transmission speed of the network connection to see if encryption speed is a bottleneck for the transmission.

Figure 4 shows the required time to encrypt and decrypt transmitted data, in this example, to compare it to the transmission results before, a five megabyte file, which resembles a high resolution photo or an audio file. Comparing the values we can see that devices with higher specifications regarding CPU, memory and battery have higher transmission speeds but can also encrypt faster.

As the network application is encrypting data packets while transmitting it is important that the encryption does not slow down the transmission. In comparison to the transmission rate we conclude that no transmissions in our tests were slowed down due to encryption, except for the constant delay that is used to encrypt the first bytes in the transmission stream.

D. Connections & Routing Information

The exchange of meta information between devices is the requirement for target-oriented routing in an Opportunistic Network. In this test we want to see if devices connect when they are in range of each other, if those connections work as desired and how meta information is exchanged. Due to our intent not to connect to devices, that have been connected shortly before, we also take a look at the connection order of the devices.

Figure 5 shows the connections of test device pairs in a fixed span of time. Because of the protocol that prevents devices from connecting to each other after a recent previous successful connection no two devices connect to each other

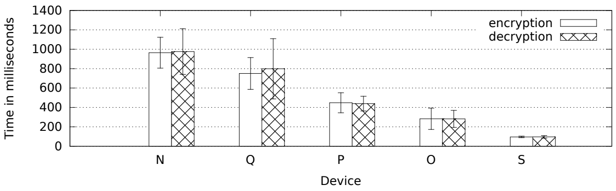


Fig. 4. Encryption and decryption test with five megabyte of data

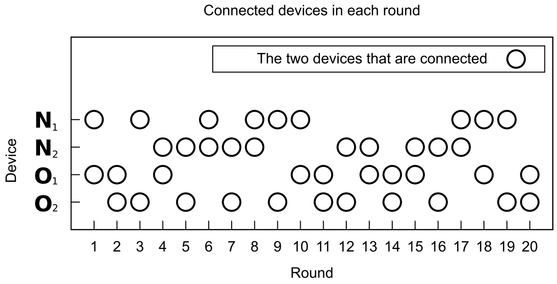
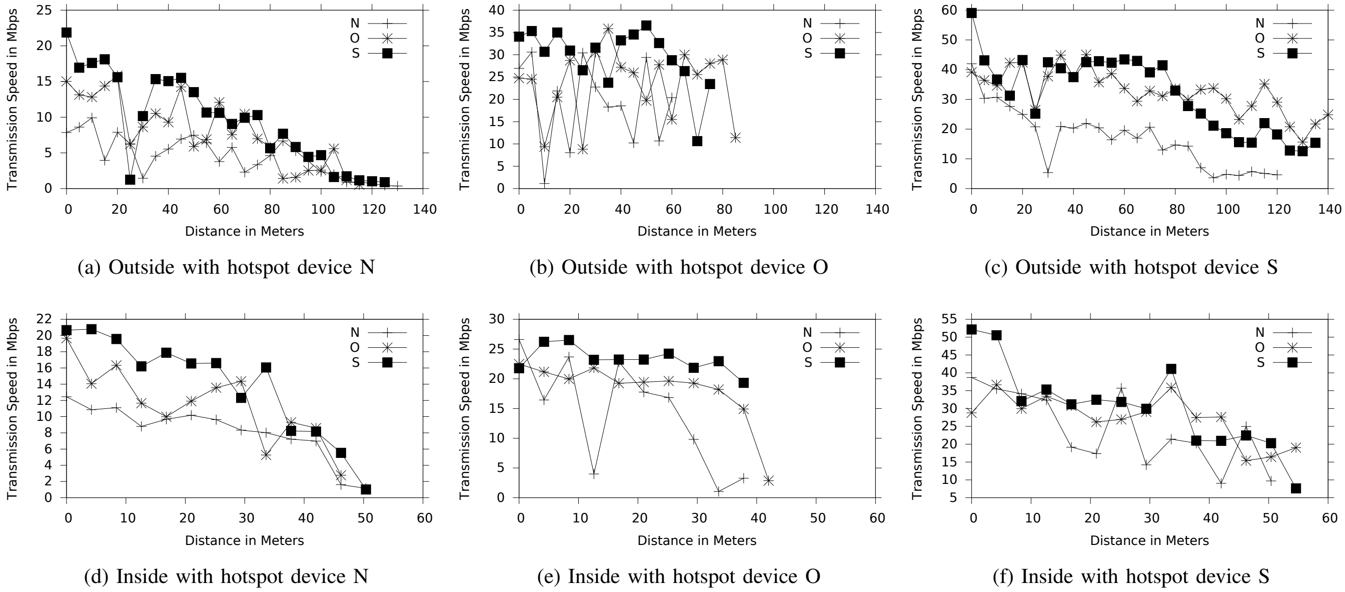


Fig. 5. Connections



(a) Outside with hotspot device N

(b) Outside with hotspot device O

(c) Outside with hotspot device S

(d) Inside with hotspot device N

(e) Inside with hotspot device O

(f) Inside with hotspot device S

Fig. 3. Outside and inside measurements with several test devices

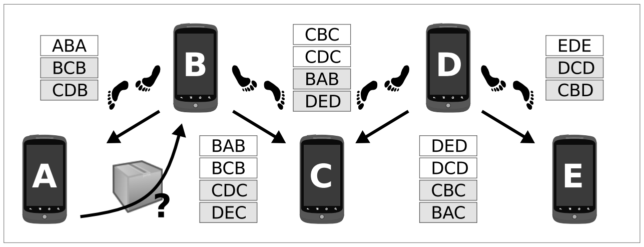


Fig. 6. Example for the connection between device A and B

twice in a row. In round 20 all devices are covered with meta information about the four shown devices and their connections to each other. As we can see always two devices that have not been connected before with each other create the next connection pair.

Additionally, we use the setting proposed in Figure 2 to see if the exchange of meta information in a two-hop setting works as expected. There are several routing protocols, like those mentioned in Section II, that can use the exchanged meta information for routing decisions.

In the following scenario, that can be seen in Figure 6, we only use the meta information for primitive targeted routing.

We want to show that a device knows if a data packet should be forwarded or not when connected to another device assuming that two hop path information is provided by the meta information. As an example we look at the cases in which device A is connected to device B and has a data packet for different recipients.

- If A has a packet for B then it is transmitted because B is the destination.
- Packets for C are transmitted because the meta information (BCB) covers a path to C.
- Packets for D are transmitted because the meta information (CDB) covers a path to D.
- Packets for E are not transmitted because no meta information at device B indicates a path to E.

Equivalent results are obtained with packets that are forwarded from other devices and additional tests with higher hop counts increase the information set but enhance the amount of data that has to be saved.

With this test we present a proof of work that meta information is exchanged as desired between the connected devices. With the meta information gathered by other devices and additional values obtained by the device itself like battery state, location or data packet information, most state of the art routing protocols we mentioned before can be adapted in our network.

In this Section we presented the results of testing our network application, the evaluation regarding the link connection setup which was our focus in this work and some proof of work regarding the routing possibilities that can be used in the network. In the next Section we draw conclusions from our results and give some insight into future work.

VII. Conclusion and Future Work

In this work we propose a network application for Android which forms an Opportunistic Network for the transmission of user data from one smartphone device to another regardless of being connected directly or over a multi-hop and time-independent neighbourhood. We use the infrastructure mode of Wi-Fi for direct connections and provide an identification scheme to enable multi-hop routing.

For both scenarios, using Opportunistic Networking as an addition to the normal networking and for enabling networking after emergency or censorship situations we give the smartphone user the ability to share and receive data using the most convenient way. The application runs in the background and does not disturb the user which is elementary for adaption by smartphone users. This way, the user has the ability to run other applications, make and receive phone calls, and to use the mobile data network if it is available.

We showed that our application offers meta data information for already existing state of the art routing protocols. Our network application can be used by other applications for OppNet connectivity as can be seen in our example Android application that can be used to send and receive files.

We implemented both a symmetric and asymmetric encryption for files whereby the symmetric encryption makes use of a passphrase and is therefore useful for spontaneous file transmissions. The asymmetric encryption uses public and private keys to encrypt files that are scheduled for a long range transmission and will be transmitted to different devices on the path to the destination. Encryption and decryption of files is offered efficiently via our API and is processed on-demand, thus it does not use unnecessary resources.

By providing own Android applications that can run as application layer, we provide some usecases for the network. Those applications show that application principles that are considered time-critical can be used in an Opportunistic Network if the situation does not allow access to infrastructure.

We come to the conclusion that our applications operate well in all tested environments, yet, due to different smartphones, we can not make general assumptions on connection times and transmission speed. Summarised, the conducted data promotes an efficient and reliable usage of our Opportunistic Networking technology, which provides a foundation for the future usage on Android devices.

For future work we will test the developed environment on different smart objects like Android OS powered smartwatches, too. The next step is to analyse other routing possibilities and test the enhanced application with more test devices and in additional environments. Considering secure routing in such networks, lightweight and organic solutions, such as [23], [24] are promising. Having an open and modular approach on our framework, this can easily be achieved and used in many scenarios by a multitude of applications. Also the extension of exchanged meta information is planned and will be tested with test devices and in a simulator [25]. The choice of several parameters that are exchangeable in the network application will be tested in the simulator, too. All future results are used to improve the parameters of the network application and to offer realistic values for different simulators.

References

[1] L. Pelusi, A. Passarella, and M. Conti, “Opportunistic networking: data forwarding in disconnected mobile ad hoc networks,” *IEEE Communications Magazine*, vol. 44, no. 11, pp. 134–141, November 2006.

[2] Android and iOS Squeeze the Competition. [Online]. Available: http://www.idc.com/prodserv/smartphone-os-market-share.jsp

[3] M. Conti, S. Giordano, M. May, and A. Passarella, “From opportunistic networks to opportunistic computing,” *IEEE Communications Magazine*, vol. 48, no. 9, pp. 126–139, Sept 2010.

[4] W. Zhao, M. H. Ammar, and E. W. Zegura, “A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks,” in *Proc. of ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2004.

[5] Z. Lu, G. Cao, and T. F. L. Porta, “TeamPhone: Networking Smartphones for Disaster Recovery,” *CoRR*, vol. abs/1606.03417, 2016. [Online]. Available: http://arxiv.org/abs/1606.03417

[6] S. Trifunovic, M. Kurant, K. A. Hummel, and F. Legendre, “WLAN-Opp: Ad-hoc-less opportunistic networking on smartphones,” *Ad Hoc Networks*, vol. 25, Part B, 2015.

[7] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks,” in *Proc. of the ACM SIGCOMM Workshop on Delay-tolerant Networking (WDTN ’05)*, 2005.

[8] ——, “Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-copy Case,” *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 77–90, February 2008.

[9] M. Grossglauser and D. Tse, “Mobility Increases the Capacity of Ad-hoc Wireless Networks,” in *Proc. of Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’01*, vol. 3, 2001, pp. 1360–1369 vol.3.

[10] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic Routing in Intermittently Connected Networks,” *Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, 2003.

[11] B. Burns, O. Brock, and B. Levine, “MV Routing and Capacity Building in Disruption Tolerant Networks,” in *Proc. of Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’05*, vol. 1, March 2005, pp. 398–408 vol. 1.

[12] W. Moreira, P. Mendes, and S. Sargento, “Opportunistic routing based on daily routines,” in *Proc. of IEEE Int. Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM ’12)*, June 2012.

[13] R.-I. Ciobanu, C. Dobre, and V. Cristea, “SPRINT: social prediction-based opportunistic routing,” in *Proc. of IEEE Int. Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM ’13)*, 2013.

[14] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, “Distributed Community Detection in Delay Tolerant Networks,” in *Proc. of ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture (MobiArch ’07)*, 2007, pp. 7:1–7:8.

[15] A. Lindgren, A. Doria, J. Lindblom, and M. Ek, “Networking in the Land of Northern Lights: Two Years of Experiences from DTN System Deployments,” in *Wireless Networks and Systems for Developing Regions*, 2008.

[16] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, “Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet,” *SIGARCH Comput. Archit. News*, vol. 30, no. 5, pp. 96–107, Oct. 2002.

[17] D. J. Bernstein, “Extending the Salsa20 nonce,” in *Workshop record of Symmetric Key Encryption Workshop*, vol. 2011, 2011.

[18] ——, “Curve25519: new Diffie-Hellman speed records,” in *Public Key Cryptography-PKC 2006*. Springer, 2006, pp. 207–228.

[19] ——, *The Poly1305-AES Message-Authentication Code*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 32–49.

[20] S. Sati, C. Probst, and K. Graffi, “Analysis of Buffer Management Policies for Opportunistic Networks,” in *IEEE ICCCN ’16: Proceedings of the International Conference on Computer Communications and Networks*, 2016.

[21] K. Graffi, K. Pussep, S. Kaune, A. Kovacevic, N. Liebau, and R. Steinmetz, “Overlay Bandwidth Management: Scheduling and Active Queue Management of Overlay Flows,” in *IEEE LCN ’07: Proceedings of the International Conference on Local Computer Networks*, 2007.

[22] C. Boldrini, M. Conti, J. Jacopini, and A. Passarella, “HiBOp: a History Based Routing Protocol for Opportunistic Networks,” in *Proc. of IEEE Int. Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM ’07)*, 2007, pp. 1–12.

[23] P. Mogre, K. Graffi, M. Hollick, and R. Steinmetz, “AntSec, WatchAnt and AntRep:Innovative Security Mechanisms for Wireless Mesh Networks,” in *Proc. of the IEEE International Conference on Local Computer Networks (IEEE LCN ’07)*, 2007.

[24] P. S. Mogre, K. Graffi, M. Hollick, and R. Steinmetz, “A Security Framework for Wireless Mesh Networks,” *Wireless Communications and Mobile Computing*, vol. 11, no. 3, pp. 371–391, 2011.

[25] K. Graffi, “PeerfactSim.KOM: A P2P System Simulator - Experiences and Lessons Learned,” in *Proc. of the IEEE International Conference on Peer-to-Peer Computing (IEEE P2P ’11)*, 2011.