Summarising Wireless Network Datasets

Charlotte Knight

September 2019

1 Abstract

Contents

2 Introduction

With such large quantities of wireless traffic now travelling through networks at ever increasing rates, processing of this data can be challenging. By introducing a summarisation step before any main processing the overall efficiency of information extraction from wireless network datasets may be increased. The aim of this project will be to create a summarised report from large datasets in order to enable more efficient onward processing of the data. This will mean using statistical approaches to maintain some identified information from the dataset while reducing the overall quantity of data that must be stored and processed. The output summaries created may utilise an existing format if my research can identify an appropriate one. My project will produce a command line application involving new approaches to summarisation to run over data collected in the CRAWDAD archives, the approach taken may (in this project or otherwise, dependent on time constraints) also be extended to work on datasets in real-time so as to eliminate the need for storing large datasets before summarisation.

3 Context Survey

3.1 CRAWDAD Usage

3.1.1 Research

When completing this research the focus has been on one particular dataset from the CRAWDAD archive, dartmouth/campus [11]. This dataset was chosen because it is one of the most popular datasets in the archive, having been cited by 374 papers at the time of writing [2]. The most frequently cited dataset however is cambridge/haggle, the reasoning for deciding not to focus on this instead is that the Cambridge dataset is comparatively small in size and would therefore benefit much less from the summarisation which this project hopes to provide.
The papers that have been selected for use in this research were chosen because they all cite the dartmouth/campus dataset. A Google Scholar [1] online search was used to retrieve the most "relevant" papers which used the chosen dataset, from these results the ones which have been most often cited in other work were selected. This selection process found papers which are relevant in the research community. As there are different versions of the dataset the search had to be repeated three times, once using the 2009 dataset, once with the 2007 dataset, and once with the 2005 dataset. For each search the five most cited results have been used. Table 1 shows a summary of the type of information each paper needed to use from the dartmouth/campus dataset. Papers in which the dataset was referenced but ultimately has not been used have been excluded.

		Properties Needed		
Paper	Topic	Device/AP Identification	Time of Transmission	Transmission Quality/Rate
Nextplace: a spatio-temporal prediction framework for pervasive systems, Scellato et al., 2011	Mobility	x	x	
Community-Aware Opportunistic Routing in Mobile Social Networks, Xiao, Wu, and Huang, 2014	Mobility	x	x	
On nodal encounter patterns in wireless LAN traces, Hsu and Helmy, 2010	Mobility	x	x	
Mobility models for systems evaluation, Musolesi and Mascolo, 2009	DTN	x	x	
Large-Scale Synthetic Social Mobile Networks with SWIM, Kosta, Mei, and Stefa, 2014	Mobility	x	x	
WAVEFORM DESIGN AND NETWORK SELECTION IN WIDEBAND SMALL CELL NETWORKS, Yang and Liu, 2014	Mobility	x		x
MAGA: A Mobility-Aware Computation Offloading Decision for Distributed Mobile Cloud Computing, Shi, Chen, and Xu, 2017	Mobility	x	x	
Flow-Based Management For Energy Efficient Campus Networks, Amokrane et al., 2015	SDN	x		x
Human behavior and challenges of anonymizing WLAN traces, Kumar and Helmy, 2009	Anonymizing WLAN Traces	x	x	
Automatic profiling of network event sequences: algorithm and applications, Meng et al., 2008	Profiling of Network Event Sequences	x	x	
Confidentiality of event data in policy-based monitoring, Montanari and Campbell, 2012	Policy-Based Monitoring	x		
Distribution of inter-contact time: An analysis-based on social relationships, Wei et al., 2013	Distribution of Inter-Contact Time	x	x	
Coverage and Rate Analysis for Facilitating Machine-to-Machine Communication in LTE-A Networks Using Device-to-Device Communication, Swain, Thakur, and Chebiyyam, 2017	Machine-to-Machine Communication	x	x	
Balancing reliability and utilization in dynamic spectrum access, Cao and Zheng, 2012	Dynamic Spectrum Access	x	x	
An Online Algorithm for Task Offloading in Heterogeneous Mobile Clouds, Zhou et al., 2018	Offloading	x	x	
State-of-the-Art Routing Protocols for Delay Tolerant Networks, Feng and Chin, 2012	DTN	x		x

Table 1: Table of the properties of CRAWDAD dartmouth/campus data used in various research projects in which it was cited. Papers are ordered by the number of other papers they have been cited by, with the most cited at the top.

3.1.2 Summary of Results

The usage of the Dartmouth University CRAWDAD dataset is primarily regarding network mobility and social interaction/encounters. As such, the most often needed information seems to be identifiers for both mobile devices and access points, and the times of connections. I found that the majority of the papers I looked at used the movement [13] or syslog [12] tracesets as these are most tailored towards mobility research.

There are also some less frequent topics of research such as software defined networking and delay tolerant networking using the dartmouth/campus dataset. These uses seem to require a wider variety of information from the data, however these instances are much less frequent than those mentioned above. these less common cases are the only ones which mention bandwidth and quality of connection.

3.2 Formats

3.2.1 Existing Formats for Aggregation of Network Traces in Mobility Research

A lack of published information on the intermediate formats used while analysing network traces for mobility research has been found during this survey. This is likely due to the encounter data not being the final outcome of the research taking place and therefore not being considered important enough to write up.

Through varied searches of DTN, Mobility, and SDN research I have found only two examples of well documented formats for storing data on device encounters. The first of these, The ONE Simulator [9], uses several reporting options to store device encounter data. The other documented format (from [26]) that was found was an association matrix. These two sources and and analysis of the information found in them is set out in the following section of this report.

3.2.2 The ONE Simulator [9]

The ONE is a simulator which generates data intended to mimic a network of mobile nodes. It then reports this data using various reporting modules, three of these modules focus on data regarding encounters between devices.

The first and most simple of these reports contains information about the dispersion of the total number of encounters experienced by the nodes in the network. It consists of two fields, one containing the number of encounters, and the other containing the count of nodes that have experienced that number of encounters. This contains no information relating to the unique nodes between which the encounters occur, or any temporal information such as duration of the encounters.

The second format provides information on the uniqueness of the encounters recorded, but loses detail about the total number of encounters in the dataset. This format also contains two data fields, one containing the values from 0 to

	Level of Detail (Complete, Most, Some, or None)		
Format	Endpoints	Duration	Frequency
TotalEncountersReport - The ONE [9]	Some	None	Complete
UniqueEncountersReport - The ONE [9]	Some	None	Most
EncountersVsUniqueEncounters - The ONE [9]	Most	None	Complete
Association Matrix [26]	Complete	Some	Some

Table 2: Table of existing formats for storing data about device encounters and the level of detail they contain regarding the unique endpoints and length/frequency of the encounters.

1000; representing promilles. The second field contains the number of unique pairs encountering with frequency withing the corresponding promille. This has a benefit of being almost static in size as the number of nodes int he system increases.

The final report format from the ONE simulation which has been looked at is a combination of the two previously discussed reports. It has three fields; the first contains an identifier for each node, the second contains the total number of encounters that the node has had, and the third contains the number of unique nodes with which it has had an encounter. This still does not uniquely identify both devices in an encounter, nor does it provide and detail about the duration of the encounters.

3.2.3 Association Matrices

Thakur et al. use an association matrix to record the percentage of time each node spends in an encounter with each other node. A matrix is created for each node, each column in the matrix corresponds to the other endpoint of the encounter, and each row corresponds to a time interval. The entry in each cell represents the percentage of the time interval spent in an encounter with the columns node. This format contains the most information out of all those discussed here, however also takes more space. The space taken will increase at with the square of the number of nodes.

3.2.4 Summary of Findings

A very brief summary of the detail contained within each of the formats discussed above is given in Table 2. Despite association matrices storing the most useful information, the polynomial increase in size with the number of mobile nodes makes using them potentially ineffective in the context of this project. The aim is to summarize a large amount of data into a smaller, easier to process format. In many cases association matrices would decrease the size of the data, but by a dramatically lesser amount than the other formats discussed here. Additionally it would be possible for the association matrix format to increase the quantity of data; for instance if N nodes had $< N$ encounters each then the association matrix for each node would include at least one redundant column. The final reporting format discussed under the ONE simulation - EncountersVsUniqueEncounters - avoids this polynomial growth, with its size increasing only linearly with the number of mobile nodes in the network. It provides less complete information regarding the unique par of nodes between which the encounter occurred, there would however be no way to preserve this information while avoiding at least N^2 growth in size.

It seems that the most complete format of those discussed in which to store encounter data while also guaranteeing a reduction in the quantity of data stored would be the EncountersVsUniqueEncounters report format. This format could also be easily modified to add additional fields such as statistics regarding encounter duration. Any additional fields would need to be carefully considered and justified in order to keep the data quantity reduction as high as possible.

4 Requirements Specification

In order of priority the requirements of this project have been listed below. These are the same requirements as set out in the DOER document submitted at the beginning of this project, and have been described in greater detail here. Each of these requirements can be implemented separately from one another, and with the completion of all high priority requirements a working product will be completed. The high priority requirements combine to describe a minimum viable product to meet the overreaching aims of this project.
Implementation of the medium and low priority requirements will complete a more versatile system which could be applied to datasets with a wider variety of sizes and formats.

4.1 High Priority

- Reduce the quantity of data from the original dataset while maintaining any information identified as useful during research into how CRAWDAD datasets are used.
  - During the context survey it was found that the most used information from these datasets was the encounters between different devices. Mobile node encounters (including the nodes involved and duration) will need to be included in the summaries for this requirement to be met.
  - It is important that the total quantity of data is reduced in the summary compared to the original dataset. For this requirement to be met it should be shown that the output is guaranteed to contain less data than the input.

- Produce summaries of the initial datasets that can be processed more efficiently than the original data.
  - In order to meet this objective it is necessary for the summary format to be as simple as possible. Superfluous information and complex file types will need to be avoided.
- Use a commonly found format to output my summaries and justify why this format is appropriate in context.
  - In section 7 (Design) a complete output format will be specified and justification given based on the research in section 3.2.
  - Specification of the output summary format should include the file type that will be used, the fields that will be included, and the variable types that will be used. All of these decisions will need to be supported by relevant and reliable research.

4.2 Medium Priority

- Allow multiple summaries to be merged (this may allow extension into distributed processing).
  - This should allow two summaries which have been output by the minimum viable product to be given as arguments at run time and combined into a single summary.
  - The output of this will use the same format as the input summaries, as if created by running the basic program on the combined network traces.
- Summarise at least two different formats of input data to create a standard output summary.
  - The system will support summarising more than one input format, but the input format should be specified at run time.
  - Whichever format the input takes the same output format should be produced.
- Allow a summary to be updated by the addition of a single data entry (this may allow extension into real-time processing).
  - This requirement will be met if a summary can have added to it a single item of data (the exact definition of which will depend on the input format of the data, for instance a single data packet transfer in a tcpdump trace). With the resulting summary being identical to the summary which would have resulted from an initial input which included the new data entry.

4.3 Low Priority

- Process datasets with an unknown input format.
  - The format of the input data should not need to be specified at run time.
  - Multiple input formats should be accepted and and the system should be able to differentiate between them in order to process each correctly.
- Identify and report if a specific summary is likely to be unrepresentative of the input dataset due to aspects such as missing data or bias.
  - Depending on the information included in the summary, how representative it is may be effected for various reasons. Due to this it is important that any information used to determine whether a summary is unrepresentative is justified.

5 Software Engineering Process

6 Ethics

The potential ethical risk associated with this project is low since there will be no contact with participants (such as interviews or questionnaires), and no personally identifiable data is expected to be used. Datasets of wireless network activity downloaded from the CRAWDAD archive will be used in this project. The datasets which are to be summarised in this project are sanitised, meaning that details such as IP and MAC addresses have been changed to obscure the identities of the network users. Despite this there is a minor risk that identifying information may be accidentally extracted during the processing of this data. If any personally identifiable data is extracted it will be removed from any devices which it may have been stored on and the code which caused the mistake will be reviewed to prevent it from happening again. A full ethics application has been made for this project and has been given the approval code CS14642. The ethical application form has been appended to this document in **??**.

7 Design

7.1 Data Flow

7.1.1 Input and Processing

Multiple formats can be processed using the command line tool that has been developed during this project. Initially only TCPdump was considered, the tool took input in the form of binary PCAP files, this was then extended to allow syslog formats to be used as input. The output produced is in an identical format regardless of the input type. Data flow for the two input types is similar and detailed in figure ??.

The first stage of processing is to extract associations between devices and access points from the raw input. These associations will later be compared with each other to determine encounters between devices, these comparisons are expensive and so it is important that as much information as possible is removed before they are made. The intermediate format used to store association information is a comma separated value file with fields of source id, destination id, start time, end time, and AP flag. The AP flag is set to 1 only if the destination address of the association can be identified as an access point. Methods used to identify access points are discussed later in this report.

TCPdump output PCAP files are very strictly structured. This allows for them to be processed without any additional information from the user. However the initial parsing is expensive with respect to time since so much information is contained within them. In comparison to this, syslog files are relatively quick to parse for associations, but the user must include a configuration file to specify the format used. Details such as defining features of association end points and the format of device identification values need to be given to the tool before it can extract associations. The expected format of a configuration file for use with syslog is detailed in figure ??.

Tying together the components of this data flow is a BASH script. Temporary files are used for storing associations in the intermediate stages, these files are then read by the MatLab script which compares associations to find encounters. A final output file is then given as a comma separated values file.

7.1.2 Summary Output

The final output specifies the endpoints of encounters, the average time of encounters between the given end points, and the number of encounters found between the given end points. Only one entry in the CSV file is present for each unordered pair of end points.

The data sets which I am using in this project have in the past frequently been used for research into mobility and encounter patterns between users [22] [28] [8] [19] [10] [14] [27]. To maximise the usefulness of this project it is intended that the output from this summarisation tool will be able to help in identifying whether a dataset is appropriate for use in mobility research, and to provide a standard format between multiple input formats which can be used for onward processing.

7.2 Associations

7.2.1 Identifying Access Points

7.2.2 Initiation

7.2.3 Timeout

7.3 Encounters

7.4 UI

8 Implementation

8.1 Stage One; Decompressing Files

The datasets downloaded from the CRAWDAD site are initially compressed (as gzip files). A directory of compressed pcap files can be given as a parameter to the summarization script. Each file in the given directory will be individually decompressed using gzip [] and piped into the first stage of the processing. This unfortunatly causes associations which span multiple pcap files to be split at the temporal boundaries of the files.

8.2 Stage Two; Extracting Associations

Libpcap is a library written in C/C++ which provides functions for the analysis of network traffic, including reading and extracting information from pcap files such as those produced in tcpdump traces. Tcpdump was the utility used to capture the CRAWDAD dartmouth dataset on which this project is initially focused. There are several wrappers of libpcap written for different languages such as pycap [21] for Python and jpcap [6]. These are generally not very regularly maintained and have very little documentation compared to Libpcap with C/C++. When initially reading in pcap files as input C++ will be used to extract the necessary information. In this stage of processing only information about the times of associations and MAC addresses of involved devices should be kept. Most of the processing will be done in subsequent stages, the main aim of this stage is to remove as much unnecessary information as possible, and to convert into an appropriate format for the next stage.

In this stage of processing a hash map of ongoing associations is built up during the reading of all packets from a pcap file. The map is updated every time a packet is read which signals the beginning of an association or the end of one. In the case of the beginning of an association, the map is updated by setting the MAC address pair to the two addresses between which the packet is sent (this is used as the key), and start and end times of the association are set to be the timestamp on the packet. Every time that a packet is read between a pair of devices the end time of the pairs current association is updated to the packets timestamp. Each packet is also checked to find whether the destination node of the pair is an access point. Since packet-specific information will be discarded after this stage it is necessary to identify access points with a flag. When the association ends, the values stored in the map are output in string delimited by commas. The values which are output are the source and destination MAC addresses, and start and end times of the association, and a flag (0 or 1) value identifying whether the destination device is an access point. This output is used as input in the subsequent processing stage.

8.3 Stage Three; Finding Encounters

The third stage of processing is written using MatLab. This decision was made due to the efficiency of the MatLab language when manipulating large tables of data (such as are used during this project) and my previous experience with the language in comparison to others similar such as Mathematica or Maple. Consideration was also given to continuing to develop this stage of processing using C++, similarly to the previous stage. This could potentially increase the efficiency and speed up processing time, however my familiarity with C++ is fairly limited and therefore the efficiency gained would likely not be worth the cost in development time by using C or C++.

This stage of processing has three main aims, firstly to extract a list of access points. Having a list of access points is important since encounters need to occur through access points (as described in the 'Design' section). The second aim is to match up the timings of associations between mobile nodes and access points such that encounters can be identified. The final aim is to then find the average duration and frequency of encounters between each distinct pair of mobile nodes. Initially the implementation of this stage of processing became drastically slower at high numbers of access points, this issue has been solved such that an almost constant behaviour in execution time if found as the number of access points increases (shown in figure 1. The limiting factor is now the total length of the input file for this stage.

9 Evaluation and Critical Appraisal

10 Conclusions

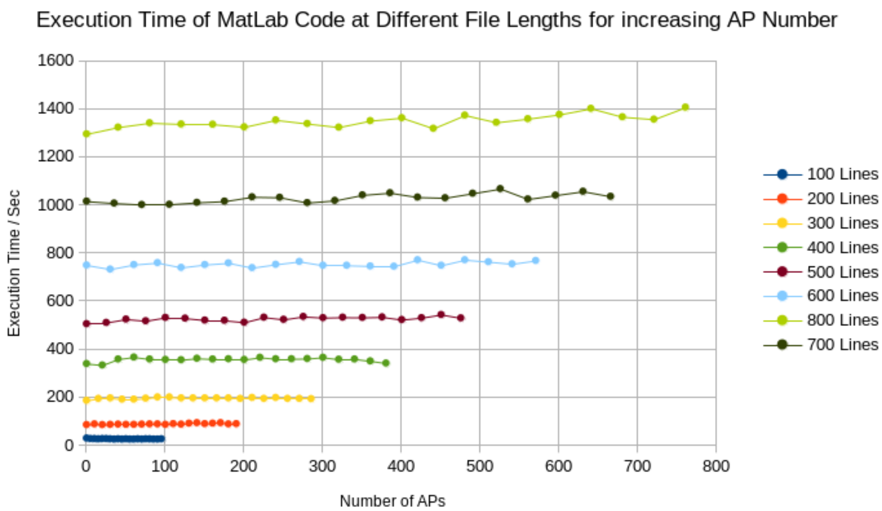

Figure 1: This figure shows the dependence of execution time on the number of access points in a file of associations at varying file lengths.

References

[1] URL: `https://scholar.google.com`.

[2] *About CRAWDAD*. accessed: 20.09.2019. 2014. URL: `http://crawdad.org/about.html`.

[3] Ahmed Amokrane et al. "Flow-Based Management For Energy Efficient Campus Networks". In: *IEEE Transactions on Network and Service Management* 12 (Dec. 2015), pp. 1–1. DOI: `10.1109/TNSM.2015.2501398`.

[4] G. Baudic, Tanguy Pérennou, and Emmanuel Lochin. "Following the Right Path: Using Traces for the Study of DTNs". In: *Computer Communications* 88 (May 2016). DOI: `10.1016/j.comcom.2016.05.006`.

[5] Lili Cao and Haitao Zheng. "Balancing reliability and utilization in dynamic spectrum access". In: *IEEE/ACM Transactions on Networking (TON)* 20.3 (2012), pp. 651–661.

[6] Patrick Charles. *Network Packet Capture Facility for Java*. accessed: 12.30.2019. June 2013. URL: `https://sourceforge.net/projects/jpcap/`.

[7] Zhenxin Feng and Kwan-Wu Chin. *State-of-the-Art Routing Protocols for Delay Tolerant Networks*. Oct. 2012. URL: `http://arxiv.org/pdf/1210.0965.p`.

[8] Wei-jen Hsu and Ahmed Helmy. "On nodal encounter patterns in wireless LAN traces". In: *IEEE Transactions on Mobile Computing* 9.11 (2010), pp. 1563–1577.

[9] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. "The ONE Simulator for DTN Protocol Evaluation". In: *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. Rome, Italy: ICST, 2009. ISBN: 978-963-9799-45-5.

[10] S. Kosta, A. Mei, and J. Stefa. "Large-Scale Synthetic Social Mobile Networks with SWIM". In: *IEEE Transactions on Mobile Computing* 13.1 (Jan. 2014), pp. 116–129. DOI: `10.1109/TMC.2012.229`.

[11] David Kotz et al. *CRAWDAD dataset dartmouth/campus (v. 2009-09-09)*. Downloaded from `https://crawdad.org/dartmouth/campus/20090909`. Sept. 2009. DOI: `10.15783/C7F59T`.

[12] David Kotz et al. *CRAWDAD dataset dartmouth/campus (v. 2009-09-09)*. Downloaded from `https://crawdad.org/dartmouth/campus/20090909/syslog`. traceset: syslog. Sept. 2009. DOI: `10.15783/C7F59T`.

[13] David Kotz et al. *CRAWDAD dataset dartmouth/campus (v. 2009-09-09)*. Downloaded from `https://crawdad.org/dartmouth/campus/20090909/movement`. traceset: movement. Sept. 2009. DOI: `10.15783/C7F59T`.

[14] Udayan Kumar and Ahmed Helmy. "Human behavior and challenges of anonymizing WLAN traces". In: *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*. IEEE. 2009, pp. 1–6.

[15] Lewis M Mackenzie. *networkpages*. accessed: 12.30.2019. URL: `http://www.dcs.gla.ac.uk/~lewis/networkpages`.

[16] Xiaoqiao Meng et al. "Automatic profiling of network event sequences: algorithm and applications". In: *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE. 2008, pp. 266–270.

[17] Mirko Montanari and Roy H Campbell. "Confidentiality of event data in policy-based monitoring". In: *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*. IEEE. 2012, pp. 1–12.

[18] Mirco Musolesi and Cecilia Mascolo. "Car: context-aware adaptive routing for delay-tolerant mobile networks". In: *IEEE Transactions on Mobile Computing* 8.2 (2008), pp. 246–260.

[19] Mirco Musolesi and Cecilia Mascolo. "Mobility models for systems evaluation". In: *Middleware for network eccentric and mobile applications*. Springer, 2009, pp. 43–62.

[20] Anthony J Nicholson and Brian D Noble. "Breadcrumbs: forecasting mobile connectivity". In: *Proceedings of the 14th ACM international conference on Mobile computing and networking*. ACM. 2008, pp. 46–57.

[21] Pynetwork. *pynetwork/pypcap*. accessed: 12.30.2019. July 2019. URL: `https://github.com/pynetwork/pypcap`.

[22] Salvatore Scellato et al. "Nextplace: a spatio-temporal prediction framework for pervasive systems". In: *International Conference on Pervasive Computing*. Springer. 2011, pp. 152–169.

[23] Yan Shi, Shanzhi Chen, and Xiang Xu. "MAGA: A Mobility-Aware Computation Offloading Decision for Distributed Mobile Cloud Computing". In: *IEEE Internet of Things Journal* PP (Nov. 2017), pp. 1–1. DOI: 10.1109/JIOT.2017.2776252.

[24] M. Sun et al. "Efficient Articulation Point Collaborative Exploration for Reliable Communications in Wireless Sensor Networks". In: *IEEE Sensors Journal* 16.23 (Dec. 2016), pp. 8578–8588. DOI: 10.1109/JSEN.2016.2611594.

[25] S. N. Swain, R. Thakur, and S. R. M. Chebiyyam. "Coverage and Rate Analysis for Facilitating Machine-to-Machine Communication in LTE-A Networks Using Device-to-Device Communication". In: *IEEE Transactions on Mobile Computing* 16.11 (Nov. 2017), pp. 3014–3027. DOI: 10.1109/TMC.2017.2684162.

[26] Gautam S Thakur et al. "Gauging human mobility characteristics and its impact on mobile routing performance". In: *International Journal of Sensor Networks* 11.3 (2012), pp. 179–191.

[27] K. Wei et al. "Distribution of inter-contact time: An analysis-based on social relationships". In: *Journal of Communications and Networks* 15.5 (Oct. 2013), pp. 504–513. DOI: 10.1109/JCN.2013.000090.

[28] M. Xiao, J. Wu, and L. Huang. "Community-Aware Opportunistic Routing in Mobile Social Networks". In: *IEEE Transactions on Computers* 63.7 (July 2014), pp. 1682–1695. DOI: 10.1109/TC.2013.55.

[29] Yu-Han Yang and K. J. Ray Liu. *WAVEFORM DESIGN AND NETWORK SELECTION IN WIDEBAND SMALL CELL NETWORKS*. June 2014. URL: http://hdl.handle.net/1903/14834.

[30] Bowen Zhou et al. "An Online Algorithm for Task Offloading in Heterogeneous Mobile Clouds". In: *ACM Transactions on Internet Technology* 18 (Jan. 2018), pp. 1–25. DOI: 10.1145/3122981.