How Broadcast Data Reveals Your Identity and Social Graph

Michael Faath, Rolf Winter, Fabian Weisshaar
University of Applied Sciences Augsburg
{michael.faath,rolf.winter,fabian.weisshaar}@hs-augsburg.de

***Abstract*—Networks rely on broadcasts and multicasts for some of the most basic services such as auto-configuration. In the recent past, application layer protocols have increasingly made use of the broadcast mechanism. Examples of these applications include Dropbox, Spotify or BitTorrent Sync. Given that broadcasts can be seen by every device in a broadcast domain, information that can be gleaned from this traffic is trivially accessible by a passive observer. Therefore, an obvious question is: what does broadcast and multicast traffic reveal about a device, a user or a group in a network?**

To answer this question, the broadcast traffic of two fairly large wireless networks was analyzed. One of these networks was the campus network of a university which was analyzed for a period of six months. Also, two SSIDs of the IETF meeting network in Yokohama in November 2015 were analyzed for a period of about 36 hours.

In addition to a general analysis of the composition of the daily broadcast traffic such as protocols observed, the number of devices, the peak times of user activity etc., a more in-depth analysis of a few protocols was carried out in order to identify users and their relation to each other. In other words, we used the available broadcast data to show that it is possible to generate a social graph of the network's users base, which e.g. helps to identify groups among students, their course of study, their online times and other personal information. We have verified the correctness of our inferred social graph by asking students to confirm our findings.

None of the observed broadcast protocols alone is to blame for the above and there is no easy technical solution to the problem while retaining the benefits of the broadcast protocols. However, there is a simple yet effective countermeasure against this kind of analysis which is non-technical and "only" requires changing user behavior.

***Index Terms*—Broadcasts, Dropbox, hostnames, privacy, traffic analysis**

I. Introduction

As RFC 919 notes: "The use of broadcasts, especially on high-speed local area networks, is a good base for many applications" [1]. While RFC 919 was written in 1984, it seems that many application developers have re-discovered the value of broadcast message exchange. Popular applications and services that utilize broadcasts today include Dropbox, Spotify, Steam and UPnP. Given that the used protocols all disclose information to every host on the subnet, it is trivial to collect this information. Protocol designers have certainly made sure that when presented with their protocol's information alone, a passive observer will not be able to make any good use of it. However, since other applications also broadcast information, these data sources can be combined to learn about devices, users and groups of users on the network.

In order to analyze what information today's broadcast traffic gives away, we have looked at all broadcasts sent on a large wireless campus network serving thousands of users at our university from October 2014 to April 2015 (note that we have analyzed both broadcast and multicast messages but will only refer to it as broadcast in the text to make it more readable). In addition, we have also analyzed the broadcast traffic on the IETF meeting network during the 94th meeting in Yokohama in November 2015 as a second data point to demonstrate that this analysis can be done at a public event. Due to restrictions of the IETF network and the much briefer measurement period the focus of this paper is on the first data point.

A. Methodology

We performed both experiments by connecting a computer to the network like any user would do, therefore we observed only traffic accessible by any participant of the network. The broadcast traffic from all users in the university's network was visible when connecting to one access point, therefore only one capture point was necessary to gather all data.

Our overall goal was to learn as much as possible not only about the devices that send out the broadcast messages, but also about their users and the social graph of these users relying exclusively on data that any user of the network could access. We used insider knowledge only to verify the correctness of our deductions.

II. Ethics and Anonymization

Tapping in on traffic on a network generally raises privacy concerns, even if it is broadcast traffic, which means data is trivially accessible by everyone on the subnet and the listener is actually part of the intended recipient group. Local legislation might or might not allow this kind of analysis and can require certain procedures and precautions.

Therefore, before we started our large-scale experiment, we used our lab to sample broadcast and multicast data. During these experiments it became apparent that personally identifiable information (PII) can appear in some of these messages. This required us to use anonymization techniques on the data beyond securely hashing information such as MAC and IP addresses. In particular, real names were frequently found in hostnames. We therefore tokenized hostnames found in all protocols sending them, removed tokens from a list of well-known terms as explained later and hashed all remaining tokens which potentially could contain PII. This removed PII from the collected data and made the later analysis much more difficult.

We consulted with the legal department of the German research network (DFN) and received a detailed statement from them. Meanwhile, they have published their work on this particular legal topic in [2]. We believe, based on this statement, our experiment conforms to German data privacy law and user privacy was protected during these experiments. Also, for the experiments on the IETF meeting network, we have consulted with a number of experts in the field, parties familiar with local law and the IETF leadership, in particular the chair of the IETF. The experiment was announced to the participants of the meeting and only a selection of the available SSIDs announced during the meeting were analyzed to give meeting participants the opportunity to opt out of the experiments. There was a lively discussion on the meeting mailing list following the announcement of the experiment [3]. We actually tried to do this experiment at the meeting 93 in Prague, but we were not able to receive a definitive answer on some of the legal questions in time [4]. Also, at that time there was a very lively discussion on the privacy implications of such an experiment [5].

Given the concerns expressed whenever our experiments were announced, we believe it is important to analyze current broadcast protocols in a privacy-preserving manner, to have a more informed debate about the issue and to understand what broadcast data currently reveals in terms of sensitive information. In particular since the data collection we performed can be trivially done by an attacker just by passively listening, without having to have special privileges or having access to a special location in the network.

III. Data analysis

We base most of our analysis on the larger data set collected at the university. There are two reasons for this. For one, given the larger time window of the analysis, the results are more expressive. Additionally, on the IETF network, broadcasts were not distributed over the air interface and only multicast messages were seen on the network. Therefore, when the data set is not explicitly identified below, we will be talking about the university's data. We will explicitly mention when the IETF data is being addressed. The smaller data set is nevertheless quite valuable since represents an interesting second data point with a very different user base—highly knowledgeable domain experts.

On the university network, the total volume of broadcast data seen during the six month time period did amount to about 40 GB, counting every byte starting with the Ethernet header. Broadcast data peaked at around 1 GB per day with a low of 3.5 MB and an average of 215 MB per day. Analyzing this data—even live—does not require special hardware and can clearly be done using a low-performance device. As expected for a university network, most traffic could be seen on weekdays during term time, while the weekends and non-term days showed far less activity.

The data contained about 35,000 different MAC addresses of which we suspect a maximum of 21,000 to be from real devices. The other MAC addresses were either from apparent experiments by students or from VPN clients that were part of the same broadcast domain for about half the time of our data collection. Luckily, it was fairly easy to identify these and to remove them from the data set.

Of all the packets observed, approximately 90% were UDP packets. To find the protocols worth analyzing, we looked at which protocols accounted for the most UDP packets sent. They were: mDNS (16%), SSDP (15%), LLMNR (13%), NetBIOS (7%) and Dropbox LAN Sync Discovery Protocol (7%). We focussed our attention mainly on mDNS, NetBIOS and Dropbox after inspecting the information these protocols could reveal. Including the other protocols as part of this analysis remains future work.

The hourly multicast traffic of the IETF network (both monitored SSIDs combined) in the 36 hour measurement window did amount to nearly one GB. During that time, about 2,600 MAC addresses were observed. Out of the protocols above, the largest fraction of the multicast traffic consisted of mDNS as well (36%)—remember that no broadcast traffic was recorded on the IETF network and therefore some protocols were missing.

A. Dropbox

The Dropbox desktop application uses the Dropbox LAN Sync Discovery Protocol to speed up the synchronization of shared directories that are present within the local network. Such folders do not have to be synced from the Dropbox servers but from other Dropbox users nearby to save bandwidth. This option is enabled by default and leads to multiple broadcast packets on UDP port 17500 every 30 seconds. The datagrams sent contain a unique identifier for the Dropbox installation of one user, the so-called *host_int*, which makes it possible to track this user even if the IP address or MAC address changes. This also enables us to identify two interfaces, e.g. a WiFi and a LAN interface, to belong to one device. The broadcast packets also contain *namespaces*, which is a list of unique IDs for the shares this user has. In other words: if two users share one directory, they announce the same ID in the namespace list.

For the six month period, 2,560 Dropbox user installations with 9,361 shares overall could be observed. The users of the analyzed network are mostly students, thus we suspected that many of them would use the shares for specific lectures or the whole course of studies, meaning they would show up as a community if we drew the Dropbox users and shares as a graph. As the complete data covers two semesters with vacation time in between, we decided to analyse only the Dropbox data from the first three weeks of the summer semester beginning on the 16th of March 2015. We removed all shares which showed up in only one namespace list as they are not of interest for an analysis of the social graph of the network's user base, i.e. these shares might have multiple users but only one within our network. After that, we removed the users which only had such a share. This left us with 712 users and 718 shares, for which we drew a directed graph. This graph is an abstract representation of a social graph between the users of the network. It shows who is sharing data with whom, but it does not reveal who these users are, which course they attend or why they have a connection.

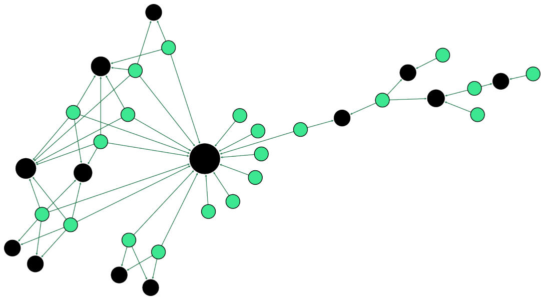

Fig. 1. Dropbox community graph

In order to group users with stronger ties into groups, we applied the Louvain method to identify 101 communities within this graph [6]. The largest of those had 48 users and 31 shares (with 42 users for the largest share). We chose six communities to investigate further for our analysis based on the number of users, shares and modularity. One of those communities is shown in figure 1. The black nodes represent shares, the other nodes represent users connected to the shares they announce. We numbered all communities and use those numbers throughout this paper to identify them. The graph in figure 1 shows community 54.

At this point we tried to identify publicly available information that could help us to further enrich those communities with information. We e.g. crawled the university's publicly accessible course schedules and tried to match online times of community members and course schedule times. Unfortunately, this did not yield good results in general.

B. Hostnames

The abstract social graph the Dropbox protocol reveals has interesting properties in itself. E.g. one can see how interconnected students are. But to make it really interesting, the nodes in the graph need to be correlated with actual user names. This clearly is information that no sane protocol designer would broadcast in the clear. A number of protocols however broadcast the hostname of the device. NetBIOS over TCP/IP and mDNS are prime examples for such protocols, we analyzed and evaluated these.

Approximately 7,600 unique hostnames were announced from 10,500 different MAC addresses. We processed these hostnames to remove duplicates (e.g. "John-Does-iPad" and "John-Does-iPad-2") and recurring strings such as "iphone", "macbook" and others. A little more than 5,300 hostnames remained for us to consider. To anonymize this list before analysis, we separated each hostname into tokens (e.g. "John-Doe" into "John" and "Doe") and stored them after hashing them securely first.

The most interesting finding in our analysis was that most users name their device after themselves using either their first, last or both names as part of the hostname. For our analysis, we only had hashed information available. In order to come to this conclusion, we needed a second data source from which we could extract names of people at the university, hash those too, and match for equality. This source is introduced in the next section. In addition, other interesting information can be extracted from the hostname, e.g. the language the user speaks. Apple users reveal particularly liberally their language by having names such as "iPhone von John Doe", where the "von" indicates a German-speaking user. A lot of hostnames reveal the device vendor, the type of device (phone, laptop etc.), the device model but also locations, faculties, functions (e.g. file server), or login names, initials and nicknames. We observed this information in control experiments conducted with a larger number of students who gave their consent to our experiment before connecting to an access point provided by us.

Enriching the social graph with this data helped identifying nodes in it. Also, mining information from social networks of well-known nodes like a list of friends can lead to the identification of previously only partially identified nodes, for example if only the first name is present in the hostname or initials. We did this manually for some members of our research group and students of the control experiments and were surprisingly successful. But to automate this for the gathered hostnames we needed to find another data source.

C. Additional data set

The LDAP server of the university is accessible from the campus network and LDAP searches are not restricted. This allowed us to automate the above process. We extracted all user data the server offered which included the login name[1], first and last name[1], email address[1], faculty, course of study, status (student, professor, etc.) and the date when the password has last been changed. We collected more than 8,400 user records this way. Out of those users, 1,300 had a unique first name, i.e. no other user had that particular first name. In contrast 4,564 last names were unique.

Using the list of 5,300 hostnames and the LDAP information, we searched for all hostnames that contained an existing first name, last name or both as part of the hostname. Interesting for us was whether the name itself would uniquely identify the user or how many LDAP users would remain as potential users of the device. As an example, if "John"(hashed) was part of the hostname, we counted the number of people called John in the LDAP data. If only a single John existed, we could identify the user uniquely. If five Johns existed, we could at least narrow the potential users down and combine it with e.g. the Dropbox data, online times, course of study of the host's Dropbox contacts etc. to finally identify the correct John.

[1] This field was only stored hashed for our analysis.

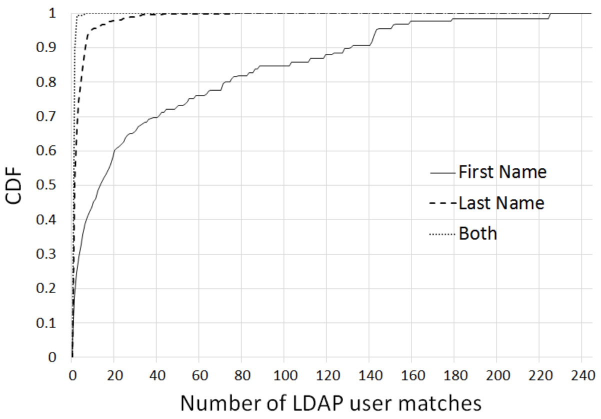

Fig. 2. CDF for LDAP user matches based on the hostname

We found approximately 2,900 first names as part of hostnames, where we did not count first names twice, e.g. "John-Does-iPhone" and "John-Mays-PC" would only count once as the occurrence of "John" in hostnames. Out of those, around 500 would uniquely identify the LDAP user. The CDF of name matches is shown in figure 2. 17% of the first names used in hostnames did uniquely identify the user of a host. A quarter of the first names matched either uniquely or at most appeared twice. The biggest number of matches observed was 244, i.e. there is a first name used in hostnames that matches 244 LDAP users.

We also found 929 last names used as part of a hostname. The probability of uniquely identifying these users is—not surprisingly—much higher. Over 50% of the last names used as part of a hostname would uniquely identify the user of the host. The highest number of LDAP users that match a last name found as part of a hostname was 77.

Finally, in case a full name was part of a hostname, the probability for uniqueness is around 90%. 293 full names were used, which was still surprisingly high. Ambiguities were—of course—much rarer. But still, there were two full names that matched six LDAP users.

Since mDNS was the main contributor of this data, we were able to do this analysis also on the IETF network data set. We still needed a list of potential users of the network in order to create a list of hash values to compare against our list of hashed hostname tokens. LDAP was not available for this on the IETF network, but the attendees list is publicly available. We extracted first and last names from that list and applied the same method as described before. Overall, 1,487 people attended the meeting. On one SSID (SSID1) we observed 1,115 different MAC addresses and on the other (SSID2) we saw a total of 1,500 MAC addresses during the 36 hour measurement time window. The total volume of mDNS traffic was around 220 MB (SSID1) and 140 MB (SSID2). Of the hostnames found in the mDNS messages, on SSID1 6 (4) contained both first and last names, 98 (133) contained last names and 247 (195) contained first names of IETF participants for SSID1 (respectively SSID2). Of those 148 (98) would lead to a unique identification of the device owner.

The above was all done automatically using scripts, meaning some names are probably missed due to hostnames with initials or nicknames (e.g. "Bill" for "William"). These are easy to parse for a human but much more difficult to parse in software. With more effort, the above numbers can therefore be increased further. Our control experiments showed that our automated analysis on anonymized data indeed misses some names a human would easily identify as a nickname or similar. Also, without anonymization, a second data source such an LDAP server or the IETF attendees list is not necessary for a real attacker to infer identities.

D. Combining the data

With the mapping of Dropbox users to LDAP users, we could attach (hashed) names to some of the nodes in our six chosen communities. For two of them, we could not identify any users as the hostnames only revealed ambiguous first names (community 36 and 56). In one community (19) we could identify two students, one used the full name and the other had a unique first name. These two users had the same course of studies, so we searched in our LDAP data for the other users in this community where we only had ambiguous first names from the hostnames. This lead to seven additional matches based on the course of studies and the time they had changed their passwords the last time. The LDAP server at our university does not return the date of creation or semester for a user. But the data showed most users do not change their password after they set it once (which probably is at the creation of the account), so we could calculate the semester with the help of the date of the last password change. Now that we knew the course of study and semester for this group, we could also verify that the online-times of the devices matched the semester's schedule after manually identifying lecture times where all students of the course should be online.

We re-did this for the other three remaining communities and decided we needed to verify the results of this analysis since we worked on anonymized data.

E. Data verification

As mentioned before we did multiple control experiments with groups of students who consented to connect to an access point where we recorded and analyzed the broadcast data in the clear. This allowed us to identify the protocols of interest for our analysis, but also revealed the device naming habits of many users as described before. If some sort of social relation between users could be identified (e.g. by a Dropbox community or—as in this case—if they are connected to the same access point), the identification of only one member of this group was instrumental for the identification of multiple other members. In other words, by just identifying a single node in the graph, it was possible to identify nodes for which the hostname information alone was not enough to identify the node uniquely. This happened in every control experiment we performed.

To verify that we in fact did identify the correct course of studies and students with the help of the Dropbox data we made an additional set of control experiments. We visited two lectures, presented our work and showed a community graph we created from the measurement data which we believed contained multiple students in those lectures. To ensure the privacy of each student, we did not include any potentially sensible data. To verify the graph we asked them (on a voluntary basis) to write down the MAC address of their devices, the hostname, and either their initials, full name or only the first name. For community 54, eight students said to use the wireless campus network and the Dropbox desktop application. They also told us they have a Dropbox share together for their semester. Only four of them had their notebooks with them, we could match all of them to the hashed MAC addresses and names we found for their respective community graph. The first names of the students without a notebook present also matched the names we found. We repeated this for another community with similar results.

Additionally, we searched for publicly accessible social network profiles of the voluntary identified users. Some of the profiles we found had their list of friends visible. We could see the majority of the other already identified group members as their friends. After hashing the names in the list of friends we could also identify other students from the community which we previously could not identify because the LDAP search showed multiple students for the found names.

F. Countermeasures

Broadcast/multicast protocols fulfill important tasks such as auto-configuration or service/peer discovery. Switching these services off would clearly not be a satisfying countermeasure to the potential privacy threads detailed above. Often there is no easy technical fix for the protocols themselves as the problem really lies in the ability to combine multiple broadcast sources. While a technical solution is difficult to achieve, there are steps to protect oneself against this kind of analysis which are simple, straightforward and fairly obvious.

In particular, the hostname of a device should never contain any kind of name or personal information which can identify a person—that includes initials, nicknames and IDs. As shown, even the first name alone might be enough to identify a user if cross-referenced with other data sources. Unfortunately, as our data has shown, this is a very common practice. Operating system vendors can actually help advising the user to create better device and service names as many of them make an initial suggestion for a device name. Most users are unaware of the fact that these names will be seen in the clear on the network. Users have to consider the tradeoff between having a unique name as hostname which for example helps to identify a device when using network shares, and a common or random hostname which helps to protect the privacy at the cost of convenience.

Secondly, restricting the data visible to non-friends on a social network profile is helpful if one does not want outsiders to recreate a social graph. This is obviously true for the social network alone, but combining this data with trivially accessible information on the network reveals much more personal information.

A third countermeasure could be the deactivation of the Dropbox LAN sync protocol in the Dropbox desktop application (or any other broadcast-utilizing protocol or application), but we do not recommend this as a general solution. This protocol helps reducing the traffic volume if a share can be found in the local network. But it is important to keep in mind that leaving this option activated might reveal connections to the persons a user shares folders with.

Besides users, network operators should clearly not have directory services such as LDAP broadly accessible. In addition, broadcasts can be easily controlled (blocked) on the air interface on modern access points which would make this kind of analysis harder (as has been done on the IETF network). The same does not apply to multicast, in particular in IPv6-enabled networks.

Finally, broadcast protocol designers should make sure that as little information as possible can be gleaned from the broadcast messages. If temporary IDs or a control back-channel over a server can be used to achieve the same goal these options should be taken into consideration. We have written an Internet draft [7] summarizing our main findings. Together with the IETF we attempt to detail a set of considerations for broadcast protocol designers, which include e.g. the ability to control broadcasts features on e.g. an SSID level, tighter control of message frequencies, advice on the use of persistent identifiers and others.

IV. Related work

There is an existing body of work that has investigated the properties of campus network traffic and the Dropbox protocol.

The paper from Singh et al. analyzes broadcast data from a campus network to classify packets as normal and anomalous traffic [8]. This work examines—like ours—only broadcast data, but considers only the IP addresses of a sender to classify the sender as genuine, malicious or unidentified. The content of the broadcast packets is not further analyzed and no cross-protocol analysis is done. Kotz/Essien analyze the usage patterns in a large campus wireless network by observing the complete traffic passively for eleven weeks [9]. They found that “residential traffic dominated all other traffic”. In contrast to our campus, the wireless network included residential buildings, therefore we can expect a different usage pattern in our data. However, a follow-up paper by Henderson et al. analyzing the same campus-wide network shows that the usage patterns changed dramatically from web traffic to peer-to-peer and streaming traffic [10]. We did not consider usage patterns as part of our analysis as we only used broadcast data. Balachandra et al. did a similar analysis of user behavior and network performance but for a public wireless LAN [11].

Given the popularity of Dropbox, it is not surprising that it has been analyzed before. Kholia/Wegrzyn e.g. analyzed the security aspects of Dropbox by reverse engineering frozen Python applications [12]. They describe how the *host_int* is received by the Dropbox application from the Dropbox server on startup and that it can—like we have done—be obtained from the Dropbox LAN sync protocol. Drago et al. characterize Dropbox by a passive measurement of all Dropbox protocol exchanges, i.e. not only broadcasts, to quantify its impact on the network [13]. This work does not explicitly address privacy issues but describes that Dropbox announces unique identifiers for the user (*host_int*) and its shares (*namespaces*).

Generally speaking, using non-broadcast traffic requires the observer to be on-path, which is not possible in the general case. We only used data that any device on the network can trivially access without being in a special location or having to rely on certain privileges.

V. Conclusions

An increasing number of protocols make use of broadcasts for various purposes. These protocols analyzed in isolation do usually not reveal much useful information that would give away personal information of a network user, i.e. do not reveal the real identity of the user. This can change when combing information from multiple broadcast protocols and publicly available data like we have done for this paper to gather as much data as possible about the network's user base, i.e. personal information. Even though initial identification of some users came directly from carelessly set hostnames, others could be identified when looking at the social graph the Dropbox LAN sync protocol reveals. We were surprised by how much data we could collect and how easy it was to identify a large number of users of the network as well as their social contacts. This is indeed troublesome, since broadcasts are trivially accessible by anyone in the broadcast domain. Furthermore, there are protocols that include IDs with which a user can be tracked quite precisely since the ID does not change often and the protocol broadcasts that ID with a high frequency. "Traditionally" such IDs were e.g. the MAC address of an interface, but it turns out that the MAC address alone today can be misleading sometimes since e.g. sleep proxies respond to queries on the behalf of other devices. With multiple unique IDs to choose from, tracking users becomes increasingly simple and once the connection between any of these IDs to real users is made, it becomes more and more easy to identify others using these as a starting point.

The protocols in use are only partly to blame. Their design can certainly be improved in ways to better hide information, but the biggest problem is the use of names or other identifiers in hostnames which is a common practice for a large fraction of users.

In the particular network we analyzed, we had additional "support" of an LDAP server, which certainly helped in automating the identification of users, their current semester (using a heuristic) and course of study. With some extra work, this information (and sometimes much more) was often accessible through other means such as social networks. It remains future work to apply automation using e.g. Google's people search API or the Twitter API for this process. The other network we analyzed had no LDAP server we could use for this purpose but a public meeting attendees list was available. Generally speaking, this helped us to analyze the network's user base, but a real attacker likely does not even need this information. In particular if the attacker tries to attack a certain owner's devices, the victim is already known and the challenge is to figure out which device is owned by the victim. Broadcast data, as shown in this paper, can make this a trivial undergoing.

On a final note, the analysis presented in this paper is still ongoing. Not all protocols have been included and the degree of automation can still be improved to being able to work on live data streams. Also, automatically including external data sources using the APIs mentioned before remains part of our ongoing efforts, which is difficult though as anonymized data cannot be used.

Acknowledgment

This work has received funding from the European Union under the FP7 Grant Agreement n. 318627, project "mPlane".

References

[1] J. Mogul, "Broadcasting internet datagrams," Internet Requests for Comments, RFC Editor, STD 5, October 1984, http://www.rfc-editor.org/rfc/rfc919.txt.

[2] H. Sporleder, "Dein name ist programm," *DFN Infobrief Recht*, pp. 16–18, Nov. 2015.

[3] "multicast/broadcast experiment at ietf94 (email thread)," Nov. 2015. [Online]. Available: https://www.ietf.org/mail-archive/web/94attendees/current/msg00490.html

[4] "No network experiment during the meeting (email thread)," Jul. 2015. [Online]. Available: https://www.ietf.org/mail-archive/web/93attendees/current/msg00331.html

[5] "Network experiment during the meeting (email thread)," Jul. 2015. [Online]. Available: https://www.ietf.org/mail-archive/web/93attendees/current/msg00121.html

[6] V. Blondel, J. Guillaume, R. Lambiotte, and E. Mech, "Fast unfolding of communities in large networks," *J. Stat. Mech*, 2008.

[7] R. Winter, M. Faath, and F. Weisshaar, "Privacy considerations for ip broadcast and multicast protocol designers," Working Draft, IETF Secretariat, Internet-Draft draft-winfaa-intarea-broadcast-consider-01, March 2016, http://www.ietf.org/internet-drafts/draft-winfaa-intarea-broadcast-consider-01.txt.

[8] R. S. Raman Singh, Harish Kumar, "Traffic analysis of campus network for classification of broadcast data," *International Conference on Intelligent Infrastructure*, 2012.

[9] D. Kotz and K. Essien, "Analysis of a campus-wide wireless network," *Wirel. Netw.*, vol. 11, no. 1-2, pp. 115–133, Jan. 2005, http://dx.doi.org/10.1007/s11276-004-4750-0.

[10] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '04. New York, NY, USA: ACM, 2004, pp. 187–201, http://doi.acm.org/10.1145/1023720.1023739.

[11] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan, "Characterizing user behavior and network performance in a public wireless lan," in *Proceedings of the 2002 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '02. New York, NY, USA: ACM, 2002, pp. 195–205, http://doi.acm.org/10.1145/511334.511359.

[12] D. Kholia and P. Wegrzyn, "Looking inside the (drop) box," in *Presented as part of the 7th USENIX Workshop on Offensive Technologies*. Berkeley, CA: USENIX, 2013. [Online]. Available: https://www.usenix.org/conference/woot13/workshop-program/presentation/Kholia

[13] I. Drago, M. Mellia, M. M. Munafo, A. Sperotto, R. Sadre, and A. Pras, "Inside dropbox: Understanding personal cloud storage services," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 481–494. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398827