# Content Source Selection in Bluetooth Networks

Liam McNamara
Dept. of Computer Science
University College London
UK London WC1E 6BT
l.mcnamara@cs.ucl.ac.uk

Cecilia Mascolo
Dept. of Computer Science
University College London
UK London WC1E 6BT
c.mascolo@cs.ucl.ac.uk

Licia Capra
Dept. of Computer Science
University College London
UK London WC1E 6BT
l.capra@cs.ucl.ac.uk

***Abstract*—Large scale market penetration of electronic devices equipped with Bluetooth technology now gives the ability to share content (such as music or video clips) between members of the public in a decentralised manner. Achieved using opportunistic connections, formed when they are colocated, in environments where Internet connectivity is expensive or unreliable, such as urban buses, train rides and coffee shops.**

**Most people have a high degree of regularity in their movements (such as a daily commute), including repeated contacts with others possessing similar *seasonal movement* patterns. We argue that this behaviour can be exploited in connection selection, and outline a system for the identification of long-term companions and sources that have previously provided quality content, in order to maximise the successful receipt of content files.**

**We utilise actual traces and existing mobility models to validate our approach, and show how consideration of the colocation history and the quality of previous data transfers leads to more successful sharing of content in realistic scenarios.**

## I. Introduction

An extremely large percentage of people already possess devices such as portable music players, PDAs and mobile phones. These devices are quickly improving both in terms of supported functionalities and storage capabilities. Current portable consumer electronics can already store days worth of music; new SDHC flash cards can store up to 32GB, so whereas previously only a few audio files could be carried around, it is now feasible to hold a person's whole music collection or a set of videos. The recent release of the *Microsoft Zune* music player brings a new dimension to the use of electronic media players. Their WiFi network interfaces allow communication of data (such as music files) between Zune devices. The use of *Digital Rights Management* (DRM) technology allows possibly copyrighted data to be shared without infringing the ownership rights of the copyright holder. DRM restricted music is useful for music distributors as it allows their product to be sampled by people, encouraging them to then purchase it. Besides music, several other data types are freely available to share, such as sample clips, movie trailers, and Creative Commons licensed data.

Crucially, an increasing number of portable devices are now equipped with wireless network interfaces, so that ad hoc networks can be formed, opening the door to a wide range of decentralised and ubiquitous content exchanges, including: file sharing, interactive games, and information updates (e.g., news headlines, traffic congestion updates, etc.), when centralised connectivity is expensive or unreliable. People spend a considerable amount of time traveling (e.g., to work or school), and enjoying leisure time (e.g., in coffee shops, bars). While traveling, such as during a daily commute through a city, people enjoy listening to music or watching movies to entertain themselves in what would otherwise be unused time. The availability of wireless network interfaces on portable devices opens the door to spontaneous, though not always reliable, communication between these devices. Despite their lack of administration and organisation, these connections can still be utilised for useful data transmission and content sharing. However, a key issue for users of these devices is how to decide whom to interact with among this variable plethora of interconnected peers.

Traditionally, finding 'who to interact with' has mainly been seen as a problem of understanding who is offering the service required. Most service discovery and selection frameworks, developed for both traditional distributed systems and ubiquitous systems, focus on how to describe services, how to formulate and spread queries, and then match queries with service descriptions [1]. However, in this formulation of the problem, the ad hoc nature of pervasive interactions is not taken into account, in particular the following two challenges have been overlooked: first, how to identify, among the providers offering a desired service, those that are likely to be connected long enough for the service provision to complete; second, how to select trustworthy providers that will actually deliver the service/content as promised.

The concept of the *familiar stranger* [2], someone that we may often see but do not personally know, is a useful property for the creation of digital relationships. In everyday life people often follow *seasonal movement* patterns, traveling very similar routes regularly and visiting the same places (e.g., the same journey to work, entering a coffee shop, going to their local pub, etc.). People possessing similar patterns will be more likely to be regularly colocated with each other. Device connection history could thus be used to improve future data communication. Even if nodes did move randomly, which could be the case for some specific applications [3], there would still be some locality information to be leveraged, so that the future of colocations could be predicted and exploited. A colocation-aware host selection framework could thus use this knowledge to select those providers that will most likely remain connected to the client for the duration of the service. Various (co)location or mobility aware routing

algorithms, such as GPSR and CAR [4], [5] already exist; however, not much has been done to exploit this knowledge in service provision [6], [7]. These approaches mostly attempt to overcome mobility and unstable connections to hide the nature of the network.

Class 2 Bluetooth devices have a range of 10 meters, which is suitable for communication between devices in relatively stationary proximity, but will not provide stable connections during free movement. In urban areas (where around half the world's population live) a lot of time is spent in close proximity to many other people, e.g., when in mass transit systems, offices or commercial districts. Most connections formed during these periods are highly transient, and are prone to being particularly unreliable. They vary from many short connections (people passing by in the street) to a few long ones (friends traveling/shopping together). Devices would obviously benefit from identifying the colocations that are not expected to last long, to avoid setting up connections which would lead to failed (incomplete) data transmissions.

Open networks, such as public wireless spaces, have been recognised as easily open to abuse, where malicious users selfishly try to maximise their own utility, or aim to disrupt the utility of others. For example, providers may maliciously falsify their service descriptions, in order to attract clients, and then deliver the service incorrectly or at a much lower standard. Because multiple identities could be created at zero cost (Sybil attack), and because of continuous changes in the network topology, similar abuses can be perpetrated fairly easily without ever being 'caught'. To discourage such behaviours, distributed reputation management systems have been proposed that enable devices to reason about the trustworthiness of other peers by means of past experiences and recommendations [8]. As discussed above, ubiquitous systems are often characterised by seasonal movement patterns of groups of devices. We thus argue that a reputation-aware content source selection protocol could be highly effective in estimating a peers' future behaviour, isolating malicious peers and thus further reducing the chances of service provision failures.

The contribution of this paper is twofold:

- A mechanism for content provider selection based on expected colocation and trust information;
- An evaluation of the mechanism through the use of realistic mobility models and real traces.

The paper is further structured as follows: Section II illustrates a public transport scenario that we will use to describe our approach. Section III provides a description of the mechanisms behind the approach. The simulation setup and results are documented in Section IV, followed by a discussion (Section V), related work (Section VI) and some concluding remarks (Section VII).

## II. Scenario

In order to exemplify our approach to content source selection, we introduce a typical pervasive computing scenario where the previously made observations hold: commuters moving through a large city by means of a mass transport system (e.g., buses/trains), carrying a device (e.g., phone/PDA/music player) that can play audio/video files and has Bluetooth capability. Users want to download and share content while they commute, for example, to enjoy in their journey to/from work or school. During periods of host-to-host connectivity, devices may transfer music tracks or video clips between each other. However, while some commuters may be cooperative, many will display selfish behaviour, only sharing content that will benefit them (e.g., pushing advertisements).

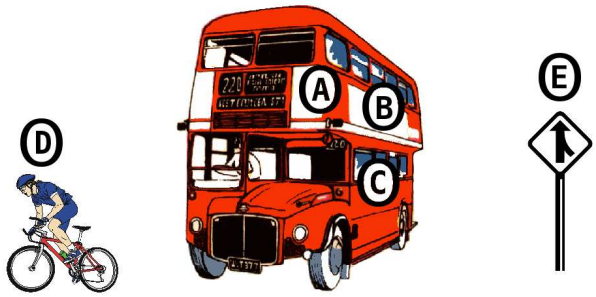


Figure 1. City commuting diagram.

Fig. 1 depicts a road in a large metropolitan city at rush hour, filled with people traveling on various forms of transport. The user Alice (A) is traveling on her daily commute to work and has been on her regular bus for the last three stops; she wants to receive media for some entertainment. User Bob (B) is also commuting on the same bus, having boarded two stops ago. A tourist Carol (C) is traveling to a museum on a route she has never been on before and she has been aboard since the start of the route. Dave (D), another worker in the city, is cycling along his usual route to work, which is similar to the bus route. Advertising board E, located next to the road, pushes advertisements in response to any requests for content.

Alice does not personally know Bob, but having been on the same bus before due to their similar commutes, they have exchanged files successfully in the past. Carol has never been to this city before, and has never previously been in contact with any of the other participants. Alice has also had interactions with Dave before, but he is rarely proximate to her for long periods of time because he cycles. Most attempts by Alice to download a file from him have terminated mid-way through, as they have moved out of communication range. This is a big problem for Alice as she wasted battery power and time that could have been spent on a successful download. This also caused useless network contention. Alice has also previously received advertisements, misrepresented as content files, from board E; she discovered this upon inspecting (e.g., trying to listen to/display) the file. Having identified the source of the transmitted data she did not find beneficial, her device stored a negative reputation about E, in order to avoid accepting content from E in the future. Also, Alice has only just come within communication range of the board, as it is a

stationary device, and it will quickly be out of range again.

As noted in this scenario, colocation time and trustworthiness are critical parameters upon which to base the selection of content provider. In terms of colocation, the aim is to filter out devices that are not really moving with us, like the case of Dave in our scenario; such devices stay colocated for very short periods and then move away. On the contrary, for devices actually moving with us (like Bob and Carol), colocation time has a certain minimum threshold (e.g., time between two or more bus stops). We can use this threshold to distinguish between 'stably' colocated devices and more transient ones.

The other parameter worth reasoning about is the provider's trustworthiness. Some sources may maliciously distribute unrequested content (e.g., advertisements) in order to maximise their own utility. Even without being malicious, the content could have simply been incorrectly labeled or encoded in an unknown format and be useless. To reduce interaction failures due to unsatisfactory provider behaviour, nodes should favour those providers they have interacted with in the past, and with which transfers have been successful. Despite the openness and dynamicity of the environments we are considering, we argue that some trust knowledge can be built; this is because of the likeliness of seasonal mobility patterns on most users (e.g., commuters on public transport make repetitive routes at similar times).

## III. Approach

In this section we describe the details of our approach. The first few steps of content source discovery are common to many approaches: a host looking for content emits a query, which is propagated in the 1 hop range of the host. The hosts receiving the request can reply to the query with a content description message[1]. The requesting host waits for a specified period of time and then examines the received replies in order to select a device to rely upon as content provider.

The basic idea behind our *content provider selection process* is to filter out those sources that are either in contact for a short time only (i.e., people just passing by) or that have not been good providers in the past. With reference to our scenario, when Alice begins a content search, she broadcasts a request specifying the type of content that she would like to receive (e.g. for music files, artist/genre/Top 40). Let us assume that devices B, C and D have some content that matches, and that they are willing to share it, so they each reply to A's query. Device E, in an attempt to push advertisements to as many users as possible, always replies to requests positively. If Alice receives no responses, she will wait for a short period and try again; this is not regarded as a failure. All replying hosts are then ranked according to Alice's host selection policy; she will then initiate the transfer with the preferred one. If the connection with the source breaks, due to moving out of range, the partially downloaded data is deleted and Alice will begin another attempt to receive content. Failures due to a device's movement do not impact on the source's trustworthiness, assuming they do in fact depart. However, if Alice considers the file inappropriate upon inspecting it, she records a negative performance against the host that provided the file. Otherwise, the transfer is regarded as a success, and Alice's trust in that provider is increased, indicating she is happy to receive more content from them.

An abstract description of the steps performed by our source selection algorithm can be found below. We split the reminder of this section in two parts, in order to provide details of the two key features of our selection algorithm.

**Algorithm 1** Content download steps.

```
loop
  Broadcast a request for content
  Wait upon a timeout for replies
  Score and rank replying hosts with selection policy
  Perform download from host with highest score
  if Data is valid then
    Increase source reputation
  else
    Decrease source reputation
  end if
  Wait for short period
end loop
```

### A. Colocation Detection

In order to disregard sources that stay colocated with a client device for only a short period of time, our approach considers the current length of colocation as an important measurement. We define a threshold AGE_THRESHOLD beyond which someone is considered 'colocated': if this threshold is not reached, it means that the source is still a risky choice (e.g., it may be a casual contact with someone traveling in the opposite direction). The value of this parameter could be set by the user, in order to allow them to choose the right balance between high overall number of files downloaded (low values of AGE_THRESHOLD) and overall low number of failed attempts (high values of AGE_THRESHOLD); or be adaptive to current conditions. This mechanism is motivated by the fact that, in a transport environment such as the one described in Section II, some peers are likely to be in the same relative location for a while (e.g., they are on the same means of transport), or they will only be colocated for a very short time (e.g., Dave momentarily stopped next to the bus at a traffic light).

In order to implement this idea, each host periodically performs a Bluetooth inquiry procedure to find out which hosts are within communication range; it then maintains a list of 'reachable' hosts, together with their length of colocation. Hosts that have been 'reachable' for longer than the AGE_THRESHOLD value, are then considered 'colocated'. The longer a host has been colocated, the higher the score it gets from our selection algorithm, in the range $[0, 1]$. The value

[1]Note that a reply is not indicative of the fact that the replying host has the content, as hosts could lie if they can benefit from spreading content they possess.

asymptotically approaches 1, with each additional colocation second adding a smaller amount to the score.

In our scenario, Alice will give Carol the highest score, as she was already present on the bus when Alice got on board. Bob, having got on only one stop after Alice, will have a slightly lower score. Neither Dave or E will be ranked, as their colocation times are shorter than the AGE_THRESHOLD and they are thus ignored.

### B. Trust Selection

Content providers are also ranked based on their trustworthiness. In our scenario, the term *malicious* is used to indicate any data source that is not perceived as 'beneficial' to the receiving host. We thus classify as malicious those hosts providing garbage data, pushing advertisements (spam), incorrectly encoding or labeling music tracks, and the like. We do not distinguish among these cases any further, the rationale being that, if content supplied by a host is of no use, then the receiver does not care about the reasons behind it, and just wants to avoid similar problems in the future. We do not treat failures due to broken connections as malicious, as we believe these are not rational actions in our target scenario (e.g., selfish hosts cutting off the connection half-way through a download would save more energy by initially refusing to serve the file).

Note that in this paper we are not attempting to build a new trust system; rather, we take advantage of existing trust approaches to improve a user's selection. Each host holds a trust value for other hosts that they have interacted with, in the range $[-1, 1]$, with $-1$ representing total distrust, $+1$ being absolute trust, and $0$ indicating a neutral opinion; a bootstrapping value of $0$ is used to represent trust in an unknown host. This value will then increase additively for each successful interaction and decrease multiplicatively for each unhelpful interaction with a host, thus favouring hosts with whom successful transfers have been accomplished, while severely punishing malicious ones. A more advanced trust system could be employed, with reputation sharing or trusted third parties, but that is out of the scope for this work. To avoid hosts that repeatedly serve bad content, a minimum trust level can be set to avoid attempting to download from them in future, even if there are no other hosts available.

With reference to our target scenario, Alice has a highly positive trust value for Bob as a consequence of a successful history of interactions, while she has a neutral trust valuation for Carol, as there have been no previous interactions between them. This will neither improve nor decrease her ranking score resulted from colocation analysis. Dave has successfully provided a few content files to Alice before, and as such has a slightly positive trust rating. Due to the pushing of inappropriate content, any previous transfers with E have resulted in a negative outcome, and its trust rating has been decreased each time, leaving it with a negative score.

The combined analysis of colocation time and trustworthiness to gauge host quality will lead to Alice selecting Bob as the content source, as he will have the highest score. In general, the two parameters are analysed separately and then combined according to this formula: $\delta \times coloc\_score + (1 - \delta) \times trust\_score$. The value $\delta$ can be chosen in a range $[0, 1]$, in order to favour trust or colocation more.

## IV. Evaluation

In this section, we first describe our simulation settings; we then provide a detailed analysis of the results we have obtained to date.

### A. Simulation Settings

We have used the discrete event simulator Omnet++ [9] to model hosts attempting to download generic content files. We assume that each host in our simulation has at least one file that matches the request (given a storage capacity of 1000s of content files on modern portable devices, this is not an overly simplifying assumption). To ensure realistic results are gained, connection logs from real life Bluetooth devices were used. Also, synthetic mobility models are employed, in order to experiment with parameters (e.g., host density/speed), which we would not have the flexibility to tune when working on real traces only.

More precisely, we have worked with four different connectivity data inputs:

- *Unitrans Run3 traces* [10], collected on the Unitrans bus system at University of California, Davis, USA in early 2006. Scanning devices were fixed into 33 buses giving a trace of all the bus routes throughout a 5 day period. The Bluetooth inquiry scan phase lasted for 5.12s, which should allow detection of 99% of surrounding discoverable hosts [11]. This was repeated every 2 minutes, giving quite a fine granularity of device detection.
- *Cambridge/Haggle-SR-10mins-Students traces* [12], collected during January 2006 in the city of Cambridge, UK. Bluetooth sightings were recorded by 36 users, carrying small Class 2 Bluetooth enabled devices (iMotes), for just under 12 days, in office and conference environments. The inquiry scan length was also 5.12s with 10 minutes between scans, arguably a coarse granularity that will not catch very many short colocation periods.
- *Random Waypoint* [13], which is probably the most widely used mobility model in mobile simulations. Contrary to the characteristics of our target scenario, this model assumes that nodes move without any correlation. We use this model as a worst case scenario, to prove the advantages of our approach even in unfavorable situations. The traces were generated by *setdest* from the CMU Monarch project.
- *Community Mobility Model* [14] which is a model founded on social network theory. It allows connections of hosts to be grouped together in a way that is based on social relationships between the individuals. This grouping is then mapped to a topographical space, with movements influenced by the strength of social ties, that may also change with time. The properties of the synthetically generated traces have been validated against real traces from the Haggle project.

Bluetooth version 1.1 allows symmetric uploads and downloads at the highest transfer mode of DH5, that is, 433.92Kb/s each way. However, this is just the specification throughput, and it will usually not be achieved in real environments, due to interference in the 2.4GHz ISM radio band (such as from WiFi or other Bluetooth devices). Indicative experiments between two mobile phones in a fairly typical urban area (with some interference from WiFi access points etc.), gave throughput as low as 120Kb/s. We thus performed a set of experiments with a selection of transfer rate ranges: 0-100, 100-200, 200-300, 300-400, 433.92-433.92 (uniformly randomly distributed). To capture environments ranging from the greatest interference up to the specification rating, unless otherwise stated the range $[200, 300]$ was used.

For a Bluetooth device to be able to detect surrounding devices, and be discovered itself, it must change into the *Inquiry* and *Inquiry Scan* substates regularly. During these substates, transmission/receipt of data cannot occur. The length and frequency of the inquiry phase was predetermined for the real traces being used, according to the study parameters. All the experiments had an inquiry dwell time of 5.12s, which was repeated every 10 minutes in the Haggle traces, and every 2 minutes for the others. We took this into account in the experiments, by reducing the data transfer rates according to the appropriate substate length and frequency. Also, while content data is being transmitted to a client, the provider becomes busy and unable to process any other requests for downloads.

In terms of content files, we assume their size to be similar to an MPEG 1 Layer 3 (MP3) audio file, with average bitrate 240Kb/s and length approximately 3 minutes. This results in files of around 5MB. There are many other audio file formats in popular usage (Ogg Vorbis, AAC, WMA), which give a relatively similar file size, obviously the exact file size will be dependent on the audio length, encoding type and encoding program. File sizes are defined according to normal distributions with means of 4, 5, 6, 7 and 8MBs with a standard deviation of 1MB. This covers the range from low quality MP3s to small videos.

In the first set of experiments we describe, all hosts are assumed to be non-malicious. However, to analyse the impact of the presence of malicious nodes in the ad hoc network, nodes which exhibit non-cooperating behaviour must be introduced. When testing resilience against malicious behaviour, up to 50% of nodes were set to be malicious. A malicious node is one that *always* serves malicious content. Though a simplification, the reasoning for this is that spamming hosts will blindly push advertisements, and hosts that have their media library in an unrecognisable format will also always serve unsuitable content. When receiving data from a malicious host, the transfer completes as normal; it is only after receipt of the complete file that the data is recognised as malicious. After a node has served two malicious files to a given client, it is blacklisted and that client will not accept content from it in the future.

Four different selection behaviours were tested when choosing which host to attempt to download from:

- *RANDOM*: devices make no effort to discern which provider is best, and simply select a random one out of the current available neighbours. This provides a lower bound, useful for analysing the improvements from our algorithm;
- *COLOCATION*: only the colocation-aware aspect of our protocol is used to choose a provider, with AGE_THRESHOLD equal to 0.15. This ignores connections newer than 140 seconds (two inquiry scans in the Unitrans experiment);
- *TRUST*: only the trust rating of nodes is used to rank and select a content source;
- *TRUST+COLOC*: both metrics are used to score the providers, and are given equal weight when computing the final score (i.e., $\delta = 0.5$). Hosts that are not 'colocated' or trustworthy enough are still ignored.

### B. Results

In this section, we analyse some of the results obtained during our simulations. In particular, for each of our four traces (Unitrans, Haggle, Random Waypoint and Community Mobility Model), we compare the content download success rate of the selection techniques listed above, when varying parameters: *transfer rate*, *file size* and *malicious rate*. The success rate is computed as the number of successfully completed downloads, over number of attempts; ignoring requests that were not satisfied due to no neighbours being present, as no selection methodology could overcome this problem.

*1) Transfer Rate:* Fig. 2 shows the effect of the wireless connection throughput upon the success rate. The file size parameter was set at 5MB, with no malicious hosts. Each source of trace input is represented, showing the success percentage of attempted downloads when using the RANDOM and COLOCATION selection schemes (given that there are no malicious nodes in this case, we are not considering trust-based selection schemes). As the graph shows, the success rate dramatically increases when using colocation reasoning, in all four traces. Unitrans experiences an improvement of around 25%. The Haggle traces already have a high success rate when using RANDOM; however, when using COLOCATION the improvement is still over 10% across all transfer rates. Quite surprisingly, RWP benefits reasonably from COLOCATION, despite hosts moving at random. As expected, CMM shows a huge improvement, going from below 30% to always over 80% success rate.

The different gains are due to features of the traces themselves, in particular, average colocation duration among any two hosts. If hosts are generally colocated for long periods of time, then selection is less important and downloads should be initiated as early as possible. On the contrary, if there are many transient connections constantly appearing, these must be ignored, in order to reduce failure rates, even at the expense of delaying the beginning of a download. We have measured the average colocation duration and, as expected, the lower this value, the higher the benefit of COLOCATION

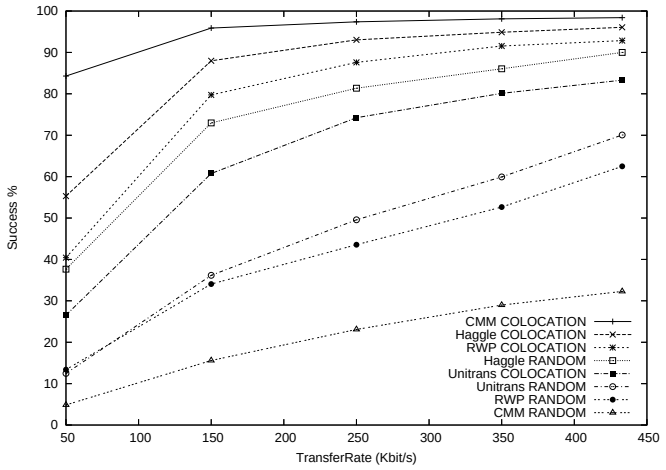


Figure 2. Transfer rate impact.

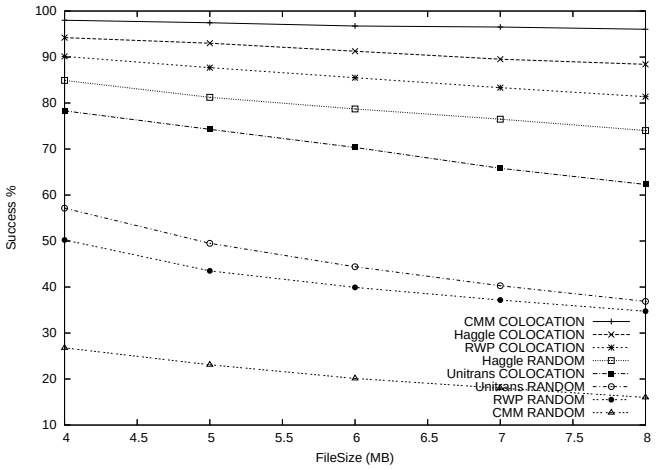


Figure 3. File size influence.

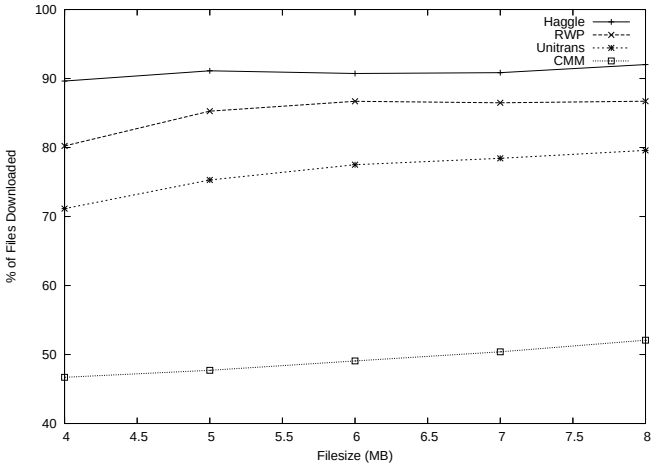


Figure 4. File download reduction.

versus RANDOM. Haggle, for example, has a mean colocation duration of 1178 seconds, which is long enough to transfer a 5MB file at 35Kb/s, so that even RANDOM gives good results. The Unitrans hosts were only colocated for a mean duration of 400 seconds, and so a 5MB transfer would need just over 100Kb/s throughput. CMM hosts were only colocated for a mean of 302 seconds, which would require at least 140Kb/s to transfer 5MB in time, thus the higher reduction in failures when using COLOCATION. All traces show a quick decline under 150Kb/s mean throughput, due to many colocations being too short to complete the (slow) file transfers.

*2) File Size:* When using the generic concept of *content*, the size of files being transferred could vary significantly. Fig. 3 shows the results from different file sizes with a medium transfer rate of $[200-300]$Kb/s and no malicious nodes. As before, we compare two source selection techniques: RANDOM and COLOCATION. The results obtained are similar to those gathered when varying transfer rate: in particular, huge gains are obtained when using COLOCATION over RANDOM, and again this is especially so when the average colocation duration is low. As the graph shows, the download success rate can be kept above 60% for all traces, even when the content being shared is much larger than standard music files, and starting to enter the range of video-clip file sizes.

The dramatic reduction in failed downloads comes at a cost. In fact, in an attempt to filter out sources which may not be stably colocated with us, some false negatives occur, that is, some sources are rejected when they would have provided successful transfers. If such sources are the only ones colocated with a device, the download is not attempted at all, thus reducing the overall number of files downloaded. In other words, less downloads are attempted, but the success rate of these attempts increases. Fig. 4 shows the percentage of files that are downloaded using COLOCATION, with respect to those that have been downloaded using RANDOM selection. The Haggle traces only show a small reduction in the number of downloads, again as a consequence of high average contact time (i.e., it is often the case that a stable node is available to perform a file transfer). The reduction in downloaded files for Unitrans and RWP is smaller, but still perfectly acceptable, given the gain obtained in reducing incomplete transfers. The biggest reduction occurs for the CMM traces, where only around half the files that RANDOM would have procured are being received. This indicates that the AGE_THRESHOLD was set too high, causing too many connections that would have eventually succeeded being ignored. Note that, as the size of the files increases, the relative throughput of the various traces improves, as the RANDOM selection performance deteriorates.

*3) Malicious Rate:* Fig. 5 shows the impact of the presence of malicious nodes serving inappropriate data upon performance. Three selection policies are compared here: RANDOM, TRUST and TRUST+COLOC. Unitrans, RWP and CMM all show a small but consistent improvement when using TRUST rather than RANDOM, whereas with Haggle the improvement increases as the malicious rate does. Quite interesting is the difference between TRUST and TRUST+COLOC: even with a very high proportion of nodes serving malicious content, it is still colocation reasoning that

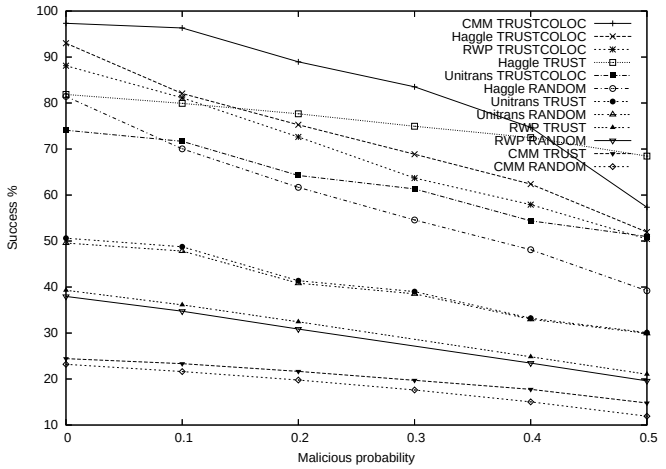


Figure 5. Malicious provider impact.

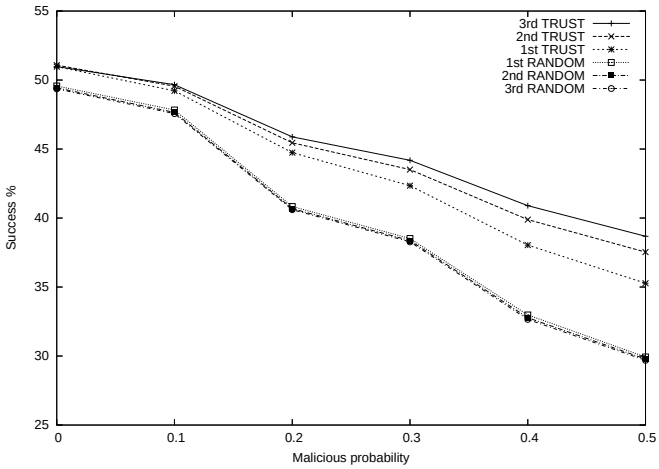


Figure 6. Unitrans malicious host impact when repeated.

improves the success rate the most, with at least a 20% gain when COLOCATION is added to the selection process. This indicates that, even in mobile networks with malicious agents, it is usually disconnections that cause most failures, not inappropriate content being served. The only exception to this is with Haggle and larger malicious rates, when pure TRUST performs better than TRUST+COLOC.

When reasoning about trust, a major restriction of the actual user traces is their limited duration: a matter of days is not enough to build a reliable reputation store. In the Unitrans traces, hosts only meet a mean of 5.9 times, and with the Haggle traces just 3.7. Note that this is not the average number of times a pair of hosts interact, but purely the number of times they have met, so the actual chances for building an accurate trust valuation from completed downloads is very low. To approximate cyclic patterns of movement, we tried repeating the Unitrans traces two and three times over, while caching the collected trust valuations from one iteration to the next. Though this will change some of the subtler features of the data, the result of 'training' the system is akin to having some history from a previous week of commuting. Fig. 6 shows the benefit of using TRUST selection when repeating the data set. The benefits are greater when the proportion of malicious hosts increases, as having past information about nodes' behaviour becomes essential; also, as expected, the benefits are higher when increasing the number of repetitions, as more knowledge is built. In particular, using three repetitions and a large proportion of malicious hosts, the TRUST approach gives an 8.8% improvement. The RANDOM selection approach obviously gives similar results no matter how many repetitions there are, as no historical information is used.

## V. Discussion

Overall our results show how a host may choose to drastically reduce the amount of failed attempts at content transfers if they are willing to sacrifice some of the resulting throughput of files in the system. The value of the AGE_THRESHOLD parameter (used to identify transient connections) is of key importance, and the optimum point of failure reduction while still receiving enough files is dependent on the underlying patterns of connections. When applied to the Unitrans traces (file size=5MB, transfer rate=[200,300]Kb/s, no maliciousness) we only had 26% of the disconnections while still receiving 75% of the files. The other real trace set, Haggle, suffered just 29% of the disconnections with receipt of 91% of files. Hosts can therefore choose to reduce the amount of network contention they cause, and battery power they expend while sharing content; important factors for small mobile devices. Making the threshold adapt to the current environment's average colocation length would ensure flexibility of the system. The experiments showed that different traces have very different performances when attempting automated Bluetooth content sharing. This would apply equally to actual deployments in different environments, and on different user's devices. Our attempts to estimate the colocation and suitability of content from a provider showed very promising results, with large increases in the success of attempted downloads.

The inclusion of artificial history information for the Unitrans traces (by re-running the traces twice or three times) increased the benefit of using a trust based system. This indicates the presence of familiar nodes in the surrounding area provides a means to overcome some of the unreliability of mobile ubiquitous communications.

There are however limitations related to all the traces we used: Unitrans are traces of the sightings made by particular buses, and although useful and interesting, do not represent the exact interaction patterns an individual person would face (except maybe a bus driver). The Cambridge/Haggle traces were from a small city, and the subjects in the study were also students with less regularity than $9-5$ to their movements, and a lot of colocation with their own devices during night-time. So although providing appropriate data they are not entirely representative of the particular scenarios we are envisioning. Namely, an environment of a large city with a high density of nodes and group movement such as on public transport. In this kind of environment the approach should behave even better

than what we have already seen here.

## VI. Related Work

Much work has focused on the challenges of achieving services, including multi-hop routing, in mobile ad hoc networks (MANETs) [15]. There has been a focus on developing multi-hop protocols for dissemination of service queries and replies in pervasive settings [16], [17]. With respect to these works, our approach takes a different perspective; in particular, we have stated that there exist many classes of pervasive computing applications (such as the public transport scenario we have focused in this paper) for which it is important that the requester and the provider are traveling together, in order for a service provision to successfully complete. In this paper we do not assume any transitivity in the service provided, which is allowed to happen strictly between two hosts.

In terms of Bluetooth network exploitation for service provision, Bluespots [18] is a public transport based content distribution system. Communication is based on an interaction pattern of users connecting to a hub installed in a bus, rather than peer-to-peer. Content that is deemed to be popular (e.g., music, news sites) is hosted on the hubs and is made available to the public. This centralisation not only causes network contention issues, but also restricts the flexibility of what data can be shared. Furthermore, this work does not consider the effect of peoples movement or the serving of inappropriate content and has single points of failure. Bluetorrent [19] is a peer-to-peer file sharing system using Bluetooth. It is similar to the operation of Bittorrent, where files are split into small pieces then downloaded and shared by clients. The main focus in this work is to overcome the problem of many short connections that occur between independently moving hosts, without considering group movement. APs are used to seed and spread selected content, requiring the creation of this infrastructure and management of the injection of content into the system. This work relies on enough people serving the same version of a file to gain the advantage of *swarming*.

## VII. Conclusions

We have presented an approach to content exchange for opportunistic networks and described a public transport scenario in which the approach could be applied. We have used Bluetooth traces and realistic mobility models to evaluate the approach. To further our exploration of ad hoc content sharing, we plan to perform our own data collection. Specifically of Bluetooth traces from consumer electronic devices, focusing on traveling on a mass transport system. We wish to gain traces over longer periods to investigate whether sufficiently useful trust values can be accumulated to improve the amount of content made available to users. We also plan to implement the content sharing system in a Java MIDlet and Python script, to ensure deployability on a wide enough selection of devices.

**Acknowledgements**: We would like to acknowledge the support of the European Union through project PLASTIC and of EPSRC through project CREAM.

## References

[1] J. Liu and V. Issarry, "QoS-Aware Service Location in Mobile Ad-Hoc Networks," *Mobile Data Management (MDM)*, p. 224, January 2004.

[2] E. Paulos and E. Goodman, "The Familiar Stranger: Anxiety, Comfort, and Play in Public Places," in *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM Press, April 2004, pp. 223–230.

[3] C. Tuduce and T. Gross, "A Mobility Model based on WLAN Traces and its Validation," *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1, pp. 664–674 vol. 1, March 2005.

[4] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *MobiCom '00: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM Press, August 2000, pp. 243–254.

[5] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks," in *Proceedings of the IEEE 6th International Symposium on a World of Wireless, Mobile, and Multimedia Networks (WoWMoM 2005). Taormina, Italy.* IEEE press, June 2005.

[6] M. M. B. Tariq, R. Jain, and T. Kawahara, "Mobility Aware Server Selection for Mobile Streaming Multimedia Content Distribution Networks," in *Web Content Caching and Distribution: Proceedings of the 8th International Workshop*, October 2004, pp. 1–18.

[7] J. Hartman, "Implementing a Service Discovery Mechanism for Mobile IPv6 Environments," Master's thesis, Institute of Communications Engineering at Tampere University of Technology, August 2004.

[8] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The Eigentrust Algorithm for Reputation Management in P2P Networks," in *Proceedings of the 12th International Conference on World Wide Web (WWW'03)*. New York, NY, USA: ACM Press, 2003, pp. 640–651.

[9] "OMNeT++ Discrete Event Simulator System - version 3.3 released October 2006." [Online]. Available: http://www.omnetpp.org

[10] J. LeBrun and C. Chuah, "CRAWDAD data set ucdavis/unitrans (v. 2006-11-01)," Downloaded from http://crawdad.cs.dartmouth.edu/ucdavis/unitrans, Nov 2006.

[11] J. P. Kharoufeh, "Bluetooth Inquiry Time Characterization and Selection," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1173–1187, 2006, member-Brian S. Peterson and Senior Member-Rusty O. Baldwin.

[12] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket Switched Networks and the Consequence of Human Mobility in Conference Environments," in *Proceedings of the SIGCOMM 2005 Workshop on Delay Tolerant Networking*, 2005.

[13] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996, vol. 353.

[14] M. Musolesi and C. Mascolo, "A Community based Mobility Model for Ad Hoc Network Research," in *Proceedings of the 2nd ACM/SIGMOBILE International Workshop on Multihop Ad Hoc Networks: From Theory to Reality (REALMAN'06)*. ACM Press, May 2006.

[15] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages," in *MobiHoc'03*, 2003.

[16] Z. J. Haas, M. R. Pearlman, and P. Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks- IETF MANET Internet Draft," July 2002.

[17] F. Sailhan and V. Issarny, "Scalable Service Discovery for MANET," *Pervasive Computing and Communications (PerCom)*, pp. 235–244, 2005.

[18] J. LeBrun and C.-N. Chuah, "Bluetooth Content Distribution Stations on Public Transit," in *MobiShare '06: Proceedings of the 1st International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking*. New York, NY, USA: ACM Press, 2006, pp. 63–65.

[19] S. Jung, U. Lee, A. Chang, D.-K. Cho, and M. Gerla, "BlueTorrent: Cooperative Content Sharing for Bluetooth Users," *Percom*, pp. 47–56, March 2007.