Efficient Matching of Services with Users in Opportunistic Network Environments

Abstract: Opportunistic Networks are a specific type of wireless ad hoc network where there is an absence of a continuous end-to-end path. The proliferation of mobile devices with Wi-Fi capability creates opportunities to forward packets by utilizing nodes as they present themselves. Such a dynamic networking environment enables services to be advertised by propagating from device to device, in order that all users in an area receive them. However, excessive propagation of service advertisements consumes energy from mobile devices, whilst also degrading the users' experience if they receive adverts for services that are misaligned with their personal interests. In this article we propose an architecture for a protocol and an algorithm that facilitates the matching of relevant service adverts with interested recipients in an Opportunistic Networking environment, whilst serving to minimize energy consumption.

Keywords: Ad hoc network; opportunistic network; matchmaking; service oriented computing

I. INTRODUCTION

The average mobile phone and tablet that is now available has more computing power than even a year ago. In particular, the inclusion of Wi-Fi capability enables such devices to be used as part of a Wi-Fi based information sharing network without needing to modify the existing mobile telephone infrastructure. In addition to this, the ability to create decentralized wireless Ad-Hoc networks means that there is no reliance upon any form of network infrastructure, such as switches, routers, access points and servers.

Ad-hoc networks are wireless networks where the nodes communicate directly with each other, when in range. As devices join together to form the ad-hoc network, messages can be sent, received and relayed between devices.

Each device, or node, within the network can act as the originator of the message, the destination of the message, or as a router that passes the message onto other nodes on a path between the source and destination. As such, a message can be routed from node to node until it reaches the intended recipient node.

If it is not possible to clearly identify a path between a sending node and an intended recipient node, then alternative, Opportunistic Network (ON) topology is required. An ON facilitates the propagation of messages amongst nodes that are within a physical proximity that allows successful transmission and reception of messages. Whenever two nodes come within range they pass messages to each other, due to the potentially constant mobility of nodes, messages can be propagated to nodes that were initially out-of-range at the point of message creation. ONs utilize a store-carry-and-forward paradigm where each node holds a copy of the message, whilst also propagating to other nodes as they come within range. The objective of an ON is to move messages as quickly as possible whilst minimizing any load upon the network itself.

We briefly illustrate some of the fundamental characteristics of an ON by showing the topology at three different points in time (Figs. 1.1-1.3). Within each figure, a wireless device (such as a mobile smartphone, tablet, smart watch, etc.), is represented by a small solid dot and a number.

Each dot maps to a node, which is in the center of a larger circle that indicates the wireless range of that device. A black line indicates which devices are within range, therefore presenting the potential for a path to be created between two or more nodes.

If at t=1 (Fig 1) device p69 sends a message to device p178, there is no direct path evident, so devices p67 and p109 will store the message.

At t=2 (Fig 1.2), some movement of nodes (devices) has occurred, yet there is still no continuous path to p178 However, the message is passed to other nodes that are in range such as p161, p152, p187, etc.

Finally at t=3 (Fig 1.3) there is now a path between p161, p169 and p100 as a result of further node mobility and so the message can now be delivered to p178.

The rest of this article is organized as follows. First, we introduce ON by way of a case study, to explain the concepts and challenges pertinent to this research in relation to the use of message propagation to advertise services. Second, we present the current state of the art in relation to ON, and consider key opportunities to advance understanding in this field.

Section IV explores the development of a protocol wrapper architecture, in conjunction with a matchmaking algorithm that aims to facilitate efficacy in message passing. Section V depicts the preliminary simulation results, before the final conclusions are described.

II. BACKGROUND AND CASE STUDY

A consistent theme of this research is to make use of a real-world scenario to both explain the challenges presented to mobile networked devices, as well as to explore the potential of such environments for the emergence of new business opportunities.

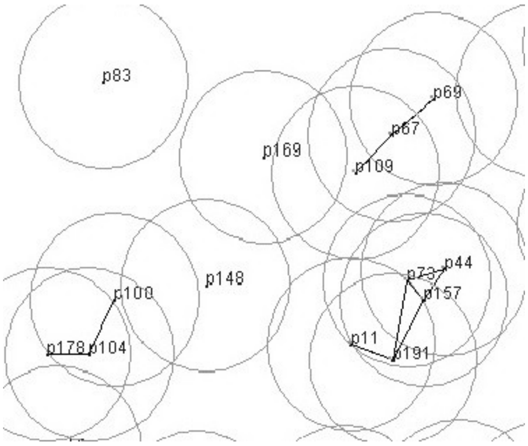

Fig 1.1 network at t=1

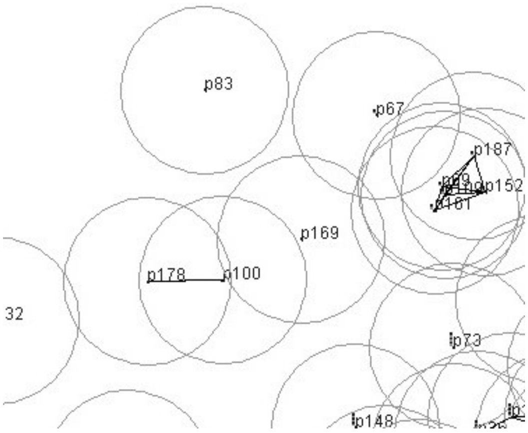

Fig 1.2 network at t=2

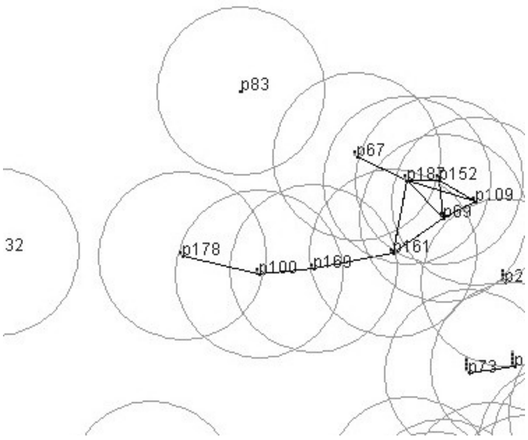

Fig 1.3 network at t=3

Prior work [11,12,13] has established a framework of metrics that provide an indication of the potential for performance optimization of an ON. This framework was developed in the context of a real-world case study, which is briefly summarized as follows.

Consider the scenario of a town center or a retail mall where shoppers and potential customers congregate. It is likely that as each user enters a location, their mobile device can attempt to join an ad-hoc network. Through this network, messages in the form of adverts for services and applications are propagated amongst the connected devices.

Software applications resident on the mobile devices proactively filters any advertisements that present themselves, retaining only those that the user is interested in. Service providers such as shop keepers/restaurateurs etc., have the ability to create and publish to the ad-hoc network adverts for new services.

These adverts propagate through the ad-hoc network to each device currently connected, and are selectively displayed or discarded as per the user's preferences, whilst also relaying messages to other devices that fall within transmission and reception range.

As a user leaves that location, messages stored on their device are carried until another location with an ad-hoc network is reached. These locations are not connected to each other, and there is no central infrastructure except for the ad-hoc wireless system. The propagation of these adverts between locations is achieved through the mobility of users.

The very nature of the intermittently connected network as described, with devices 'carrying' adverts between locations, means that an Opportunistic Ad-Hoc network would be a natural solution.

In the context of this there is a need to be both effective and efficient in matching adverts (provision) to the user's device. There are a number of reasons for this. First, users that are bombarded by unwanted messages are likely to turn their device off.

Second, unlike a wired network where the devices are connected to mains power, a mobile device relies on its own scarce resource (battery). All wireless transmission requires energy, and excessive, irrelevant message propagation will increase the power consumption of mobile devices. Finally, excessive message propagation results in increased traffic for the network.

The overall objective of this work is to facilitate the provision of providing the relevant adverts to interested parties without crippling an individual device or the network as a whole.

Consequently, if a device receives multiple copies of the same massage it will significantly reduce the capacity of the battery. Additionally, the amount of storage on the device to store adverts is another constraint so as not to interfere with the normal operation of the device.

With regard to the network it too has a limit to its capacity, referred to as bandwidth; this indicates the maximum number and the size of the messages that can be passed across the network. As the number of messages reach this limit the network becomes congested and further messages are unable to be sent, resulting in overload.

In order to establish the extent by which messages can be efficiently provisioned in an ON environment, a number of characteristics need to be monitored and evaluated such as:

- Quantity of adverts arriving at the interested parties;
- Volume of network traffic generated;
- Time taken for interested parties at various points to receive the advert;
- Proportion of interested parties that were reached;
- Amount of energy consumed by a device/the whole network.

III. RELEVANT LITERATURE

Typically, wireless network infrastructure makes use of multiple nodes (access points) to define paths through which packets can travel. In contrast, an ad-hoc network is a collection of autonomous nodes that communicate wirelessly without any pre-existing infrastructure. They are able to configure themselves dynamically without any external intervention.

The nodes can act as both end systems and intermediary systems to forward packets to other nodes. As a consequence, they are able to take the form of a multi-hop wireless network, allowing for end-point to end-point communication even when those nodes are out of wireless range.

The actual path taken by packets varies as nodes become available. In an ad-hoc network, a node may move beyond the range of all other nodes, breaking any existing communication path.

One solution is to make use of passing nodes that are moving in the correct direction to carry messages to the out-of-range node. This method is referred to as opportunistic routing [3], which is the basis of ON.

As described earlier, ONs utilize a store-carry-and-forward paradigm, with the objective of moving the message as quickly as possible to a destination whilst minimizing any load on the network itself.

There have been a significant number of protocols created to support ad-hoc networks, but all require an effective end-to-end path. In ONs, the lack of a defined end-to-end path has required the development of different protocols.

1) Opportunistic Network Routing

There are two main classes of opportunistic routing; context-oblivious and context-aware routing. Each class has its own way of forwarding messages. Context-oblivious routing tends to flood the network with copies of the message making it available for all the nodes in the network. Every node in the network has the opportunity to forward the message to the destination. This strategy provides a large pool of available resource to forward the message to its destination, with the disadvantage of flooding the network with redundant message duplication, causing network congestion.

Alternatively, context-aware routing relies upon knowledge of the users' behavior, as recorded by their device. This category is further divided into another two sub-categories; mobility based routing which utilizes the mobility information of the nodes for message forwarding, and social routing which exploits the social relationships of the nodes to route the message.

The mobility information includes a description of the mobility pattern of nodes, the connectedness of a node, and the relative position of the nodes in the network.

Social routing focusses upon the behavioral knowledge about the node's environment and its social contact with other nodes. Unlike context-oblivious routing, both mobility-based and social context-aware routing do not flood the network with message duplications and so they serve to reduce network congestion.

Epidemic routing [15] is an example of a context oblivious protocol. Epidemic (or infection) routing is based upon a flooding scheme, whereby a node with a message forwards that message to all nodes that it meets while in motion.

This continues until a specified number of hops are achieved or the message lifetime expires. This protocol is effective in that it achieves node coverage with low latency; however, it is less effective at constraining the load placed upon a network as a whole.

Not only is the message forwarded to every node in the area, but nodes that have already received the message will continue to be transmitted to. This causes congestion and is wasteful in terms of other scarce resources such as bandwidth, storage, and power.

Whereas Epidemic routing is context-oblivious (it uses no context information in forwarding to nodes), other protocols are context-aware. Such an example is the Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) [10].

PRoPHET makes the assumption that node mobility is not random and that there is a rationale behind their movements. Every node is assigned a probability that it will come into contact with a certain node; the probability increases when it connects with that node and reduces as a function of time otherwise.

When nodes connect they swap the predictabilities of the message destinations they carry. The message is passed only if the passing node has a higher probability of delivering it.

An alternative to this is Bubble Rap[6] which is a social network protocol. This is a context based system where the context is described as the social community the users are part of. Communities are defined by the pattern of contacts between nodes, which are ranked by their sociability, a measure that is based on the nodes they are most frequently in contact with.

When a message is sent the protocol looks for nodes of the same community. If a node carrying the message comes into contact with a node of the same community as the destination, the message is passed. Alternatively, if the new node is not in the same community, but has higher ranking than the current node, the message will be passed.

Verma & Srivastava [16] identified that context-aware protocols have limitations when the context information is missing, causing a high overhead, extended message delay and poor delivery performance.

The second significant aspect of this research is the need to automate a match between users and advertisements for services that are of interest.

2) Matchmaking

Matchmaking algorithms have been used extensively in research into cloud based services and grid systems. These can be complex processes that generally include three entities, a) the requester, b) the provider and c) the matchmaking service.

Matching is most often about comparing the offerings of different service providers in order to find the ‘best’ match [5,9]. The matchmaking service makes use of methods such as probability, set or graph theory to obtain the best match by considering a significant number of factors. In order to be able to compare services they use a complex service specification language, such as the Unified Service Description Language (USDL) or the Agent Capability Description Language (ACDL) [14].

An alternative system is currently used in Location Based Services, which is designed to use geo-location information to provide users with location specific information, such as the location of specific entities such as Restaurants etc. These systems tend to be based upon a request to an external, often cloud based, matchmaker service component which interprets the user profile and produces matches based on filter queries [18].

Another method to provide location based services is that referred to as publish/subscribe, where the user subscribes to topics published within range of the device [4]. Although these systems are designed to use Ad-Hoc networks for the device to device communication, the matching of messages is done using a web based service.

An alternative proposal is that of the use of community brokers [17]. In such a scenario the nodes are divided into ‘communities’ which are grouped together due to their social behavior. One node that can easily reach all the others in the group is then designated the broker for that group. All subscriptions and adverts are then routed through that one node for that particular group.

The primary aim of this research is to achieve the efficient matching of service advertisements to users in an ON environment. It is likely that this will be achieved in part by comparison of an incoming message to a device, with a user profile that is resident upon the device, in order to produce a result that aligns with the interests and intentions of the user.

In order to accommodate this, we have deemed that the message content language must permit aggregation so that new instances of information can be added as and when they emerge. For the purposes of this work, advertisements are expressed as keywords for matchmaking [9].

IV. EXPERIMENTAL WORK

To enable performance comparisons to be drawn, this research will reference appropriate benchmarks, such as Epidemic routing in the case of network routing.

1) Experiment Design

As described earlier, the base scenario that has been used so far is that of mobile users within a retail shopping mall, that possess devices with wireless network capability.

Within the geography of the shopping mall, users move randomly within the area. Also, users enter and leave the area (and therefore the network) at random.

The mobility of users will be represented via the Random Waypoint model [1][7]. The Random Waypoint model attempts to capture the movement of humans, and each node is given random coordinates in the simulation area (waypoint). Each node moves at a constant velocity directly towards the given waypoint.

At this point the node pauses and a new waypoint is defined, together with a new random velocity. Simulations were executed using Epidemic and other non-context aware routing protocols.

Whilst this refers to the base scenario, there are also more specialized scenarios that can be envisaged.

For instance, when user leaves a location, they may be carrying advertisements for services that will be recognised by subsequent connections to ad-hoc networks in other locations.

In this way an originator in one location, having identified that a significant amount of custom comes from another location, could target that location. For example, a chain of retail outlets could propagate a voucher that is redeemable in any one of the bricks and mortar stores.

The propagation of these adverts between locations is achieved through the mobility of users; it is the mobility of users that make the connections in an ad-hoc fashion.

The simulations assume that users move based on the Shortest Path Map Based Movement model [2]. The Shortest Path Map Based Movement model is one of a number of Map-based movement models, where the movement of the node is constrained to a path as defined in a set of map data.

In the Map-based model, nodes are able to move randomly along any path, whereas in the case of the Shortest Path Map Based model, nodes follow the shortest route to a point on the map.

This point on the map is chosen either as a random point on the map or from a list of Points of Interest. In this case there will only be a single Point of Interest provided. This opens up the possibility of defining a range of simulations based on a number of context aware routing protocols.

In this research the matching of services to user's needs is to be done in an effective and efficient manner, this including not only the transmission protocol but also the matchmaking algorithm itself.

In systems where an external broker is used to carry out the matchmaking there is a significant overhead due the time required to communicate to a third party, as well as the extra network load imposed.

However, given the very nature of an ad-hoc network without any infrastructure, a community broker node would not be viable due to the extra processing and therefore power dissipated in the process. In such a case it would be logical for all the processing needs to be confined within the individual device. This arrangement has the additional benefit of reducing concerns regarding user privacy concerns as the user's profile remains within the device and is not communicated to a third party.

2) Network Layer

In the TCP/IP model the protocol sits at the network layer. Messages are passed up to the application layer via the transport layer for processing. The proposed matchmaking service might require messages to be passed from the protocol layer up to the application layer. It can be assumed that widespread adoption of the service amongst users would result in myriad user preferences being instantiated. The end result would be that a significant number of these messages would ultimately be discarded, resulting in wasted processing of raw messages that did not conform to the preferences of the user.

A more efficient method might be to carry out a first pass of the matchmaking at the protocol layer. This would allow messages that the user is broadly interested in to be filtered, and any message types that have not been seen before, to be processed and passed up to the system application layer for user interaction.

Further processing could then be carried out at the application layer, such as adding a newly discovered preference into the profile.

How much processing takes place at each level depends upon the speed of the main processor and the time between received messages. Too many processing cycles would result in dropped messages, and too little processing would reduce the accuracy of the matchmaking.

In order to keep the matchmaking algorithm in close proximity to the protocol layer, a protocol wrapper has been adopted to prevent any direct modifications to the underlying network protocol. Fig. 4.1 illustrates this.

In order to simulate this process The ONE simulator was selected since it was designed specifically for Opportunistic or Delay Tolerant Networks. The ONE simulator is designed to emulate various routing protocols and movement models [18].

The upper layers of the communications stack are not implemented other than a link that can be used to pass messages up to the next layer. The way the network layer is implemented in the ONE Simulator, is similar to that required in that the network layer is already in two parts.

The lower part is the implementation of the specific protocol and above that fits a thin wrapper that provides the common interface for the various protocols. It is in this layer that the testing and buffering of messages takes place for the node, by adding the matching algorithm at this point so that testing can take place.

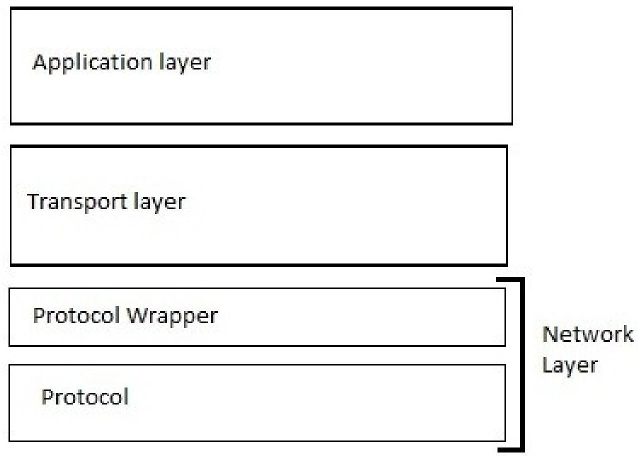

Fig 3.1 TCP/IP stack modified with protocol wrapper.

3) Matching Algorithm

The matchmaker algorithm needs to be relatively simple in order to minimize an additional processing overhead, and will need a clear profile to work with. Even in this there needs to be a level of sophistication built into the matchmaking algorithm otherwise the system will not gain acceptance with the users. For example the system will need to work on multiple levels i.e. a general descriptor and then a modifier. An example would be a keyword of say of restaurant then the second level keyword that would define what type i.e. French, Chinese, Mexican, etc. In addition when an advert for a restaurant that isn't in the list of existing restaurants is received user input will be required. So for instance the device receives an advert for a Korean restaurant which isn’t in the profile, it will be passed to the user so that the user can respond, the user would accept or rejects the advert, and the profile would be updated automatically. This would provide a learning system that could build-up a user’s profile on the fly, that is rather than the user been required to initially input a profile the device would learn the user’s preferences based on their response to adverts.

In a simple way, if the system received an advert for a restaurant of type Chinese, which is already in the user’s profile, it would therefore be a matched and the advert would be passed onto the user.

Alternatively, if the user’s profile shows a dislike for French and an advert for a French restaurant arrives, the profile would indicate that this is not a preferred option and the advert would not be passed to the user. Then there would be the special case where there is currently nothing in the profile, which would be passed to the user. The users response would then cause it to be added to the profile as either a like or dislike.

An additional concept would be where there were two adverts for restaurants which are both shown as likes on the user's profile. The matchmaker algorithm (see appendix 1 for the detailed algorithm) could possible then make a decision based on the distance to the two restaurants or in a simple case display both adverts.

The message is passed as a list of keywords together with a number that signifies the number of keywords in the message (levels).

The "user profile" currently consists of a number of sets of keywords, one for each level to be matched. Also there are two complete groups of sets, one for "likes" the other for "dislikes", and example set is shown below;

For example, a three level information database could have the form shown in figure 3.2.

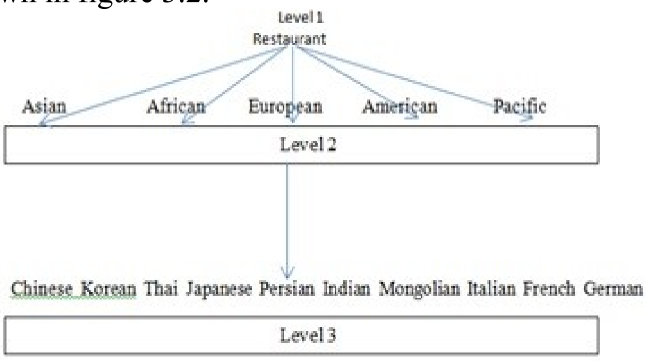

Fig 3.2 Database tree diagram

Whilst, for a given customer enquiry this information structure can be refined to include "likes" and "dislikes" to represent the preferences. As an example consider the preference lists:

Likes:	[Restaurant;	Level 1
	Asian, African, Pacific;	Level 2
	Chinese]	Level 3
Dislikes:	[European, American	Level 2
	Japanese, French]	Level 3

Giving the revised structure shown in figure 3.3.

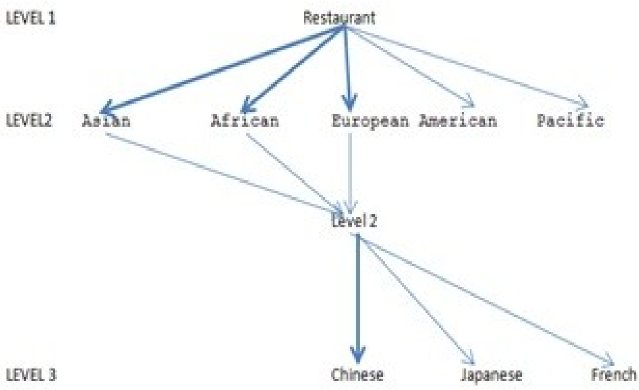

Fig 3.3 Likes and dis-likes diagram

The algorithm (see Appendix 1) firsts enters a while loop that will step through all the levels unless the current level that is been tested is either in the hate group or does not exist in either group. If all levels in the message match those in the likes group the pass_level will equal the levels and the message will be displayed. If any level matches an entry in the hate group decn will be set to 2 and the message will be ignored. While if at any level there is no match with either group the message will be displayed for user decision.

Sample messages together with the result from the search are shown below

message 1	[Restaurant; American; Brazilian]	would fail at level 2,
message 2	[Restaurant; African; Moroccan]	would be referred at level 3.
Message 3	[Restaurant; Asian; Chinese]	would pass at level 3.

V. SIMULATION RESULTS

The matchmaking algorithm has been simulated in MATLAB. To be able to evaluate the effectiveness of the search, in relation to the number of levels to be considered, three simulations were carried out.

In each simulation a node attempted to contact a neighboring node (mobile device) and the message compared with this neighboring device's user profile ('likes' and 'dislikes' lists). This determined the level (1, 2 or 3) and message status at which the search was completed (success, refer or reject), further these results were used to investigate the cost/benefit of the depth of search used (level of information to be considered).

The three messages simulated were:

Simulation Run 1 message
Restaurant 'likes' High probability
Asian
Indian

Simulation Run 2 message
DIY 'likes' Medium probability
Chain store
Carpets

Simulation Run 3 message
Food 'likes' Very High probability
Farmers market
Indian

In this simulation each mobile device could have up to 5 members in each of their 'likes' and 'dislikes' sets. Tables 5.1-5.6 illustrate the results of each of the simulations. For each of the keywords in the message, a weighting was applied to indicate the user's preference in their device profile.

Simulation 1				
Level	Likes probabilities		Dislikes probabilities	
1	0.3	High	0.1	Low
2	0.1	Medium	0.1	Low
3	0.45	Very High	0.05	Very Low

Table 5.1 Simulation run 1.

Simulation 2				
Level	Likes probabilities		Dislikes probabilities	
1	0.1	Medium	0.1	Medium
2	**0.05**	**Low**	0.3	High
3	0.3	High	0.2	High

Table 5.2 Simulation run 2.

Simulation 3				
Level	Likes probabilities		Dislikes probabilities	
1	0.55	Very	0.2	Medium

		High		
2	0.25	High	0.3	High
3	**0.05**	**Low**	0.05	Very Low

Table 5.3 Simulation run 3.

The next tables show the effects (proportion of correct decisions) had the search been halted at each level and the implied work carried out by level of search employed.

Level	Simulation 1	Simulation 2	Simulation 3
1	0.71	0.90	0.46
2	0.98	0.99	0.87
3	1.00	1.00	1.00

Table 5.4a Summary of results from all simulation runs.

The results from the first simulation P_likes(H, M, VH) show that halting the matching algorithm after two levels (only) see figure 3.4 results in a 2% maximum error, where an error occurs when a message has been passed incorrectly.

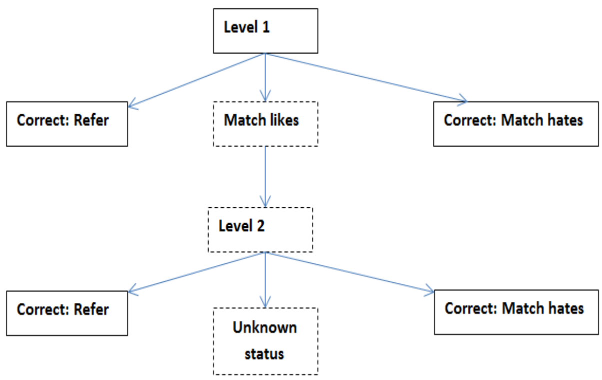

Fig 3.4 Two level matching

An interesting result highlighted by these simulations is that the depth of search required, so that most decisions will be correct, seems to depend upon the proportion of devices expected to like the level 1 category. Table 5.4b shows that the greater the probability of a message being liked at level 1 the greater the depth of search required to reach the correct decision (to pass or not to pass the message).

Simulation	P(level 1 liked)	Percentage of correct decisions halting at	
		Level 1	Level 2
2	0.1	90	99
1	0.3	71	98
3	0.55	46	87

Table 5.4b Summary of results from all simulation runs

Thus additional knowledge concerning the 'liked' status of the options (probability) within the query could indicate an appropriate depth of search required to give (mostly) correct decisions, to pass or not pass the message.

Additionally the simulation indicated how the 'cost' measured in terms of the number of tests carried out is related to the liked status of the top level option within the query. The number of option comparisons for each level of search when N devices are interrogated are shown in table 5.5, showing that a message where the top level category is very popular will cause more work than a message where the top level category is unpopular, a theoretical consideration of an m-level search is given in appendix 2.

Case	P(L1 liked)	Level 1	Level 2	Level 3
2	0.1	N	1.10N	1.11N
1	0.3	N	1.29N	1.31N
3	0.55	N	1.54N	1.68N

Table 5.5 Summary of results from all simulation runs.

From figure 5.5 it can be seen that as the number of levels increase the 'cost' increases significantly moving from level 2 to level 3. Which indicates a reduction of efficiency as more levels are required to match user requirements. In order for the efficiency not to deteriorate significantly this leads to only having 2 levels of matching at the protocol layer.

1) Measurement Framework

The protocol benchmark used for comparison of the performance of opportunistic network protocols is the Epidemic protocol. The measurement framework for the protocol has already been established in prior work[11,12,13]. However, a similar benchmark does not exist for the process of service matchmaking, and it is important for the validation of experimental findings to establish pertinent metrics.

An initial set of characteristics to monitor is likely to comprise the following:

- Quantity of messages that have been successfully matched with a user's preference;
- Time taken to achieve a match;
- Volume of traffic imposed upon the network;
- Latency- the time taken to match a service advertisement to a user.
- Quantity of processor cycles consumed by a mobile device within the network. This will help infer the quantity of energy consumed by the process.

Once these characteristics have been identified and verified, other challenges can be evaluated.

In particular this will permit investigation into the effects of differing user and device behavior. A user may arbitrarily decide to simplify the matchmaking process by deliberately using broad terms to express their preferences. Such users may be satisfied with a general set of advertisements appearing upon their devices.

Conversely, a user may develop very specific preferences that require several levels of validation before a successful match is achieved. This will necessitate more processing cycles, increased network traffic and as a result, more energy from the device power source.

The application developer may also wish to either delegate the matchmaking service to the protocol wrapper, or alternatively use the functionality provided and extend it by augmenting new functionality.

Such an approach could imbue a device with a greater degree of sophistication and autonomy in terms of managing the service advertisement management process. It is feasible that this could also be a dynamic behavior, that responds to environmental and system effects such as a low battery, where power needs to be conserved.

One further aspect is to investigate the effect of user mobility patterns upon the message propagation and matchmaking process, since a user's lack of a preference for a service advertisement should not prevent messages spreading in an ON environment.

The preliminary results for the matchmaking algorithm are promising and the next step is to implement the algorithm within the protocol wrapper architecture (described in Section IV) in The ONE simulator in order to start optimizing the process.

VI. CONCLUSIONS

Opportunistic Networking has considerable potential for new business models, particularly since it avoids the need for traditional infrastructure, yet there is a proliferation of users with mobile, Wi-Fi enabled devices.

Whilst such devices have typically been smart phones, the wider acceptance and resulting dependence upon 'smart' technology means that emerging devices, such as smart watches and such like, serves as enablers for greater connected-ness between users and business organizations.

The realities of ON can already be realized through ad-hoc network connections and a variety of network protocols. However, the potential of more sophisticated uses of this technology also presents challenges for the research community, such as the need to be able to efficiently manage and provision network services in an ON environment.

This research makes use of prior work to establish a measurement framework for ON, and augments this by proposing a means by which the propagation of messages can be managed both by users, and also more transparently by the mobile devices themselves, through the use of a matchmaking algorithm and protocol wrapper.

The protocol wrapper enables matchmaking to be added to the ON protocol of choice, without requiring modification of the underlying protocol. This architecture permits a more autonomous approach to managing the matching of service advertisements to the preferences of users, whilst also providing access to application developers who may choose to extend the functionality even further at the application layer.

Preliminary simulation of the matchmaking algorithm indicates that in excess of 61% of service advertisements can be discarded at the first level, which suggests that there may be a measurable reduction in power consumption as a result of reduced processing on mobile devices. The simulations also indicate that the most efficient point to halt the matching process is at level 2 of the matching. As mobile devices reduce in size and become more pervasive and embedded, this is of particular interest to the research and business communities.

REFERENCES

[1] Camp, T., Boleng, J., and Davies, V. A. Survey of Mobility Models for Ad Hoc Network Research. Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2002, vol. 2, pp. 483–502.

[2] Choffnes, D. R., and Bustamante, F. E. An integrated mobility and traffic model for vehicular wireless networks. Proceedings of the 2nd ACM International Workshop on Vehicular Ad-hoc Networks, 2005, pp.69-78.

[3] Conti, M.; Giordano, S.; May, M. & Passarella, A., From opportunistic networks to opportunistic computing, Communications Magazine, IEEE, 2010, 48, 126-139.
[4] Eugster, P. T.; Garbinato, B. & Holzer, A., Location-based Publish/Subscribe, Proceedings of the Fourth IEEE International Symposium on Network Computing and Applications, IEEE Computer Society, 2005, 279-282
[5] Guan, T.; Zaluska, E. & De Roure, D., A semantic service matching middleware for mobile devices discovering grid services, Proceedings of the 3rd international conference on Advances in grid and pervasive computing, Springer-Verlag, 2008, 422-433 .
[6] Hui, P., Crowcroft, J., & Yoneki, E., BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks, In Proceedings of the 2nd ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch), 2007, 241-250.
[7] Johnson, D. B., and Maltz, D. A., Dynamic source routing in ad hoc wireless networks. In Mobile Computing. Mobile Computing, Kluwer Academic Publishers, 1996, 153-181.
[8] Keränen, J. Ott, T. And Kärkkäinen, T., The ONE simulator for DTN protocol evaluation, in: Proceedings of 2nd International Conference on Simulation Tools and Techniques (SIMUTools'09), Rome, Italy, ICST New York, NY USA. ISBN 978-963-9799-45-5, 2009, pp 55:1-55:10.
[9] Kuokka, D. & Harada, L., Matchmaking for information agents, Readings in Agents, Morgan Kaufmann, 1995, 672-678.
[10] Lindgren, A., Doria, A., and Scheln, O., Probabilistic routing in intermittently connected networks. Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003), 2003, 19-20.
[11] Smith, A. & Berry, S., Evaluation of a Framework for Measuring Efficency in Opportunistic Ad-hoc Networks, Emerging Intelligent Data and Web Technologies (EIDWT), 2012 Third International Conference on, 2012, 61-65.
[12] Smith, A. & Hill, R., Towards a Framework for the Evaluation of Efficient Provisioning in Opportunistic Ad Hoc Networks, Proceedings of the 2011 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, IEEE Computer Society, 2011, 32-36
[13] Smith, A., Hill, R., 'Measuring Efficiency in Opportunistic Ad Hoc Networks'. International Journal of Interconnected Networks, World Scientific Publishing, 2011, vol. 12, iss. 3, pp. 32-36.
[14] Sycara, K.; S, W.; M, K. & Lu, J., LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace, in Cyberspace. Autonomous Agents and Multi-Agent Systems, 2002, 173-203
[15] Vahdat, A. & Becker, D., Epidemic routing for partially connected ad-hoc networks.Technical Report CS-200006, Duke University, 2000.
[16] Verma, A. & Srivastava, A., Integrated Routing Protocol for Opportunistic Networks, International Journal of Advanced Computer Science and Applications, Vol. 2, No.3, 2011.

[17] Yoneki, E., Hui, P., Chan, S-Y., and Crowcroft, J., A socio-aware overlay for publish/subscribe communication in delay tolerant networks. Proceedings of the 10th ACM Symposium on Modelling. Analysis and Simulation of Wireless and Mobile systems (MSWiM), 2007, 225-234.
[18] Yu, S.; Al-Jadir, L. & Spaccapietra, S., Matching User's Semantics with Data Semantics in Location-Based Services, 1st Workshop on Semantics in Mobile Environments (SME 05), 2005.

Appendix 1:

```
decn            ← 0     % decision
test_level      ← 1     % current level tested
pass_test       ← 0     % current level passed

while (test_level ≤ levels) AND (decn=0)
        If((ISEMPTY({new_message_type(test_level)}∩
                              {likes(test_level)})=TRUE)
                If((ISEMPTY({new_message_type(test_level)}∩{hates(
                                                      test_level)})= FALSE)
                        decn ←2
                        %dont want to display this message
                else
                        decn ←3
                        %category not seen, do we want to add it to likes or
                hates
                end
        else
                pass_test ← pass_test +1

        end
        test_level ← test_level +1
end

if pass_level = levels
        "display message"
End

if decn = 2
        "message ignored"
end

If decn = 3
        If message STATUS is good
                %"display message", add to likes?.
                %new information at level   'pass_level + 1'
        else
                "message ignored"
        end
end
```

"STATUS is good", depends on parameters to be derived.

Appendix 2:

An investigation of the basic case provides a mathematical basis for truncating the depth of search dependent upon the form of the query.

A2.1 Formal definition of the search conditions

The basic search considers only those problems defined formally by:

L_{ij} = "True" option i at level j is liked, and

H_{ij} = "True" option i at level j is not liked

It then follows that the following describe the matching procedures

$[H_{ik} = \text{"True"}] \rightarrow [H_{i,k+1} = \text{"True"}]$

If an option not liked at level k implies that lower level options are not relevant in this search,

and $[L_{ik} = \text{"True"}] \rightarrow [L_{i,k-1} = \text{"True"}]$

hence to be required to interrogate an option at level k option the higher level options in the query will have been liked.

finally if

$[L_{ik}]$ and $\neg[L_{i,k+1} \vee H_{i,k+1}] = \text{"True"}$

the query will be referred at level k+1.

A2.2 Analysis

If p_{ij} is the probability that option i is liked at level j, and q_{ij} is the probability that option i is not liked at level j (note, $r_{ij} = 1 - p_{ij} - q_{ij}$, is the probability of referral at level j given that the query has been liked at higher levels).

One level search

If a one level search (only) is implemented it follows that

$$P(\text{correct decision}) = (1 - p_{1j}) + \prod_{k=1}^{n} p_{ik}$$

and if p_{1j} is small ($p_{1j} = \varepsilon$)

$$P(\text{correct decision}) = 1 - \varepsilon\left(1 - \prod_{k=2}^{n} p_{ik}\right), \text{ giving}$$

$(1 - \varepsilon) < P(\text{correct decision}) < 1$, close to 1.

But if p_{1j} is large (p_{1j} = P) it follows that

$$P(\text{correct decision}) = 1 - P\left(1 - \prod_{k=2}^{n} p_{ik}\right)$$

$$(1 - P) < P(\text{correct decision}) < 1$$

Thus it follows that if p_{1j} is small a one level search (only) would be appropriate and efficient.

Two level search

The second level will only be reached if the query has "passed" at the first level, hence it follows that if the search were halted at this level then:

$$P(\text{correct decision}) = (1 - p_{2j}\, p_{1j}) + \prod_{k=1}^{n} p_{ik}$$

and if p_{2j} is small ($p_{2j} = \varepsilon$)

$$P(\text{correct decision}) = 1 - \varepsilon\, p_{1j}\left(1 - \prod_{k=3}^{n} p_{ik}\right),\ \text{giving}$$

$$(1 - \varepsilon) < P(\text{correct decision}) < 1.$$

Thus it follows that if p_{1j} is not small and p_{2j} is small a two level search (only) would be appropriate and efficient.

m level search

Assuming that the probabilities p_{ij} are not small, for i=1 to m-1, it follows that

$$P(\text{correct decision}) = \left(1 - \prod_{k=1}^{m} p_{ik}\right) + \prod_{k=1}^{n} p_{ik}$$

and if p_{im} is small then it would be efficient and effective to halt the search at this level.

Thus a simple rule is to stop at the level when P(likes) is very small, an option within the query is unpopular hence reducing the cost of the search process.