International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in; editorijoer@gmail.com

Vol.5., Issue.4, 2017
July-August

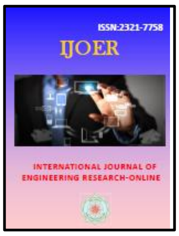
REVIEW ARTICLE

ISSN: 2321-7758

FASTER CONTENT SHARING OVER SMARTPHONE BASED DELAY-TOLERANT NETWORKS

Dr SAI MANOJ KUDARAVALLI[1], Dr. P CHIRANJEEVI[2]

[1, 2]Associate Professor, Department of Computer Science and Engineering
Amrita Sai Institute of Science and Technology Paritala, Krishna Dist, AP, India
e-Mail:- ksmasist@gmail.com



M.KAMRAJU

ABSTRACT

Delay Tolerant Network (DTN) service and protocol stack and presents an implementation of it on the Android platform that is called "Bytewalla". It allows the use of Android phones for the physical transport of data between network nodes in areas where there are no other links available, or where existing links need to be avoided for security reasons or in case the Internet is shut down by a government authority like it happened in some Arab countries during the spring of 2011. With the growing number of smartphone users, peer-to-peer ad hoc content sharing is expected to occur more often. Thus, new content sharing mechanisms should be developed as traditional data delivery schemes are not efficient for content sharing due to the sporadic connectivity between smartphones. To accomplish data delivery in such challenging environments, researchers have proposed the use of store-carry-forward protocols, in which a node stores a message and carries it until a forwarding opportunity arises through an encounter with other nodes. Most previous works in this field have focused on the prediction of whether two nodes would encounter each other, without considering the place and time of the encounter. In this paper, we propose discover-predict-deliver as an efficient content sharing scheme for delay-tolerant smartphone networks. In our proposed scheme, contents are shared using the mobility information of individuals. Specifically, our approach employs a mobility learning algorithm to identify places indoors and outdoors. A hidden Markov model is used to predict an individual's future mobility information. Evaluation based on real traces indicates that with the proposed approach, 87 percent of contents can be correctly discovered and delivered within 2 hours when the content is available only in 30 percent of nodes in the network. We implement a sample application on commercial smartphones, and we validate its efficiency to analyze the practical feasibility of the content sharing application. Our system approximately results in a 2 percent CPU overhead and reduces the battery lifetime of a Smartphone by 15 percent at most. The implementation of a store and forward messaging application and a Sentinel Surveillance health-care application (SSA) that runs on top of Bytewalla are presented together with a few usage scenarios. Our conclusion is that the integration of DTN links in the general IP-network architecture on mobile phone platform is feasible and will make it easier to integrate DTN applications into communication-challenged areas. To our knowledge our implementation of the bundle protocol is the first on the Android platform.

Keywords: Delay Tolerant Networks, Android, Mobile phone, Store and forward networks, wireless communication, location dependent and sensitive, pervasive computing

Literature Survey

In this section, we analyze the problem of content sharing in smartphone-based delay-tolerant networks and describe the solutions. As stated in the Introduction, we focus on store-carry-forward networking scenarios, in which the nodes communicate using DTN bundle architecture. Some smartphones in the network store content that they are willing to share with others. All smartphone users are willing to cooperate and supply a limited amount of their resources, such as bandwidth, storage, and processing power, to assist others. Our goal is to allow users to issue queries for content stored on other smartphones anywhere in the network and to assess the chances of obtaining the information needed. We assume that smartphones can perform searches on their local storage, and we find the relevant results for a given query to facilitate searching. Conceptually, the content sharing process is categorized into two phases: the content discovery phase and the content delivery phase. In the content discovery phase, a user inputs requests for content in a content sharing application. The application first searches for the content in local storage, and if not found, the application generates a query message based on the user's request. The query is then spread in the network based on a specific forwarding decision and search-termination technique. When the content is found, the content delivery phase is initiated, and the content is routed toward the query originator. Given many unknown factors that influence the efficiency of content sharing, including the naming of content, availability of bandwidth and energy resources, and network connectivity, these two phases primarily decide the performance of content sharing. For example, a user needs to use free Internet on Fairview Avenue around 1 pm and wants to know the location of available open Wi-Fi hotspots and the Internet speed of Wi-Fi APs. In this scenario, the user is setting spatiotemporal limitations on the delivery of the content. Although the content may be well indexed, and there may be plenty of resources for sharing, if the content cannot be discovered and delivered to the requester at the right time and at the right place, the user will not be satisfied with the sharing scheme.

Symbol Notation

$U_i(d)$	Utility function of node i for destination d
R	Communication range
δ	Time instance – minimum duration for discovering a meaningful place (e.g., 10 minutes).
t	Current time
T	Query/Content lifetime (e.g., 3 hours)
l	Location information, $l = \{lat, long, \varepsilon, \mathcal{A}\}$
lat	Latitude
$long$	Longitude
ε	Error bound (accuracy)
$\mathcal{A}$	Access point set – includes the AP name, signal, etc.
$\|l_1 - l_2\|$	Euclidian distance between locations l_1 and l_2
p	Path – consists of two locations and the transition duration, i.e., $p(l_1, l_2)$.
$\mathbb{M}$	Mobility information – set of locations
$\mathbb{L}$	Meaningful places – set of locations l
$\mathbb{P}$	Path information – set of paths p
$\vec{\sigma}_t$	Vector of accelerometer readings σ_x, σ_y, and σ_z for the x, y, and z axes, respectively, at time t

DELAY TOLERANT SERVICES AND PROTOCOLS

Several evolving wireless networks such as terrestrial civilian networks connecting mobile wireless devices, including mobile phones, PDAs, or wireless sensor networks (in water or land) or space-networks, such as the InterPlaNetary (IPN) Internet Project do not conform to the Internet's underlying assumptions which are: Continuous, Bidirectional End-to-End connection, short round trips and consistent symmetric data rates between source and destination and low error rates on each link. Such networks are characterized with intermittent connectivity, long or variable delay, asymmetric data rates, and high error rates. Therefore, connecting them to the Internet requires the intervention of a service that can translate between incompatible networks characteristics and which can provide a buffer for mismatched network delays. The Delay-Tolerant Networking (DTN) helps to address the above mentioned technical issues. The DTN concept was first conceived within the Inter-Planetary Network Research Group charter (IPNRG) of the Internet Research Task Force (IRTF), to deal with the challenges in high delay environments. The DTN architecture defines an overlay layer on top of the TCP/IP architecture between the transport and the application layer of the network on which it is hosted. This layer forms an overlay that uses persistent storage to solve network interruption related problems and provides functionalities similar to the Internet layer described in the original ARPANET/Internet designs. The overlay network approach is represented by the Bundle Protocol (BP) (RFC5050). The basic idea is that each packet transmitted is called a "bundle" and contains all of the signaling as well as the data required to transit the transport layer which is referred to as the bundle convergence layer. Therefore, the bundle architecture operates as an overlay network, whereby DTN nodes are identified by Endpoint Identifiers (EIDs), which are the bundling equivalent of addresses. Bundles are routed in a store and forward manner between participating nodes over varied network transport technologies (including both IP and non-IP based transports). Other DTN protocols include the Licklider Transmission Protocol (LTP) which is a point-to-point protocol designed to be a potential convergence layer to support the bundle protocol, though it can also be used in other contexts. It is primarily designed for the true high-latency case of deep space communications; to be usable as a convergence layer for the bundle protocol. But it can also be used above traditional connectionless transport layer like UDP in terrestrial contexts, including sensor networks using data mules.

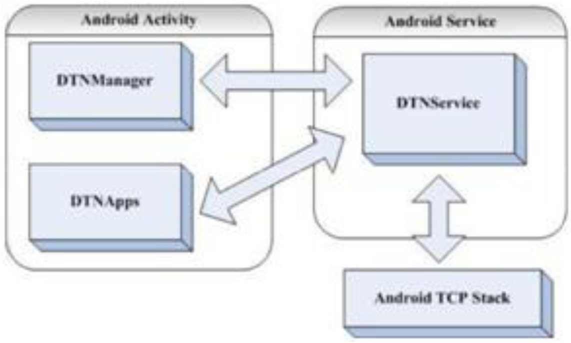
Network Architecture:

Top of the DTNService. Two sample DTN applications developed are DTNSend and DTNReceive. DTNSend is a DTN application allowing users to send text messages over DTN. DTNReceive is a DTN application allowing users to receive text messages over DTN. Because both of them are front end applications similar to DTNManager, they are mapped to the Android activity.

Content Delivery

When query matching content is discovered, the content carrying node should send only a subset of the results. This requirement is necessary to limit the amount of resources used both locally and globally for transmitting and storing the responses, and to remove potential duplicates. The query originator assigns a limit for both the number of replications and the volume of content that should be generated. When nodes need to forward a query message, the limitations included in the query message are used to make a forwarding decision. If the volume of the content exceeds the response limit, the node needs to select which ones to forward. For example, when AP hotspots are huge in number, the set of AP hotspots with the most reliable Internet speed is chosen and forwarded to the query originator. In this paper, we propose a general solution for content delivery, and we examine how to deliver contents rather than which content to deliver. Nodes broadcast a one-hop beacon message to indicate their existence. This beacon message contains the list of queries and the list of content headers the node is carrying. Both queries and content headers include the destination node's (i.e., query generator node) mobility information. Based on the mobility information in the content headers, the nodes decide to exchange actual contents. Allowing all nodes to carry the same content is wasteful, so DPD uses the multiple-copy case Spray and Focus [10] with a special utility function to limit the number of content replicas. Spray and Focus routing has two phases: the spray phase and the focus phase. In the spray phase, for every message originating at a source node, L message copies are initially spread, forwarded by the source and possibly other nodes receiving a copy, to L distinct "relays."

MOBILITY LEARNING AND PREDICTION

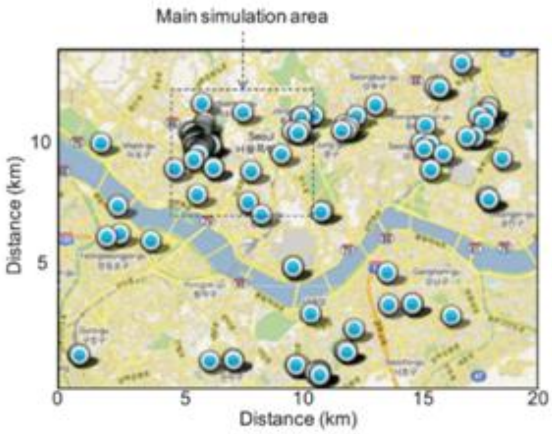


In this section, we describe the core component of the proposed content sharing scheme which involves mobility learning and prediction. Although contents are not coupled to user locations, our content discovery and content delivery mechanisms use the location information of the user for accurate content sharing. Specifically, in (2), we use the location information of two nodes: a node that makes a forwarding decision and a node that has generated a query. Therefore, we utilize a smartphone-based context provider called LifeMap [21], which is specifically designed to support location-based services in indoor/outdoor environments. Also, in (3) and (4), the predicted location of nodes is used to estimate the distance between these nodes at a given time. We extend LifeMap with a Prediction module, which is a special component to predict the future locations of nodes. The Content Provider module of LifeMap is modified to extract the temporal features of locations. We examine these modules, and for the description of other components, readers may refer to the original papers

Mobility Learning

In daily life, people typically visit a number of places, but not all of these are meaningful for learning people's mobility. Indeed, DPD requires the discovery of locations where content sharing can be performed. Content sharing is successfully performed in places where smartphone users stay long enough, as perceiving the existence of other nodes and message exchanging requires several minutes depending on the size of the message, the bandwidth, and the network interface. Hence, we are basically interested in discovering places where the user stays longer than a certain duration (i.e., meaningful places) and the context in user movement (i.e., paths).

Evaluation:



Implementation

The application generates hotspot information that provides available Wi-Fi access points in an area. Fig. 5 shows the user interface of an open Wi-Fi finder application. Users may use two kinds of view modes: 1) map view and 2) list view. In the map view, the user's current location, which is obtained from LifeMap, and nearby hotspots are shown. The icon of hotspots differs in accordance with the number of APs available in the hotspots. In the list view, a list of hotspots, which includes a hotspot's location name, address, distance from the current position, and the strongest AP signal strength, is shown. In the map or list view, the user may see detailed information on the selected hotspot and obtain direction to reach the hotspot. Also, the user may search APs in a specific location by selecting a search item in the menu. A search dialog (Fig. 5c) appears, and the user is prompted to provide the search location, region (radius), and delivery deadline. If no hotspot information is found in the selected region, the DTN content discovery DPD is initiated to discover and deliver the requested hotspot. When the requested hotspot is found, the user is informed about the discovery, and detailed information on the discovered hotspot is provided, as shown in Fig. 5d. Hotspot information contains the availability of Internet access, the approximate downlink speed, the signal strength, and so on. Additionally, the application provides users the capability to extend the local database using the URLs from the central servers or database files. Users set the sharing permission and level to protect their privacy in application settings. 5.2 Implementation Analysis An application for smartphones should exhibit the following characteristics in order to be pragmatic: lightweight in terms of CPU and memory overhead, and efficient in resource utilization and energy consumption. Specifically, ad hoc content sharing applications should be carefully designed as such applications have to continuously run in the background. With the following experimental analysis, our goal is to identify the feasibility of content sharing applications on smartphones.

Conclusion

In this paper, we proposed an efficient content sharing mechanism in smartphone-based DTNs. We attempted to utilize the advantages of today's smartphones (i.e., availability of various localization and communication technologies) and appropriately designed the protocol. In designing a content sharing algorithm, we focused on two points: 1) people move around meaningful places, and 2) the mobility of people is predictable. Based on this proposition, we developed a mobility learning and prediction algorithm to compute the utility function. Thus, in contrast to conventional methods, the proposed sharing mechanism does not require contact history. We learned that contents indeed have geographical and temporal validity, and we proposed a scheme by considering these characteristics of content. For example, distributing queries for content in an area 20 miles from the location of the content searcher has only a 0.3 percent chance to discover the content while generating 20 percent extra transmission cost. Also, the time limitation on query distribution reduces transmission cost. Most important, the proposed protocol correctly discovers and delivers 87 percent of contents within 2 hours when the contents are available only in 30 percent of nodes in the network. The implementation of our system on Android platform indicates that the scheme results only in a 2 percent CPU overhead and reduces the battery lifetime of a smartphone by 15 percent at most. Finally, we believe our system still has room for improvement. Specifically, the use of asymmetric multicore processors and efficient sensor scheduling is needed to reduce the energy consumption of smartphones' sensors. Further, since location is the key element of the proposed solution, user privacy should be carefully considered. We plan to address these issues in our future works.

Bibliography

[1]. K. Fall, S. Farrell, "DTN: An architectural retrospective", IEEE Journal on selected Areas in Common, Vol.26, no.5.pp. 828-826, June 2008.

[2]. A. MacMahon, S. Farrell. "Delay-And Disruption-Tolerant Networking," IEEE Internet Computing, vol. 13, no. 6, pp. 82-87, Nov/Dec. 2009.

[3]. V. Cerf, A Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, H Weiss "Delay-Tolerant Networking Architecture", IETF RFC 4838, Apr. 2007.

[4]. K. Scott and S. Burleigh, "Bundle Protocol Specification", IETF RFC 5050, Nov. 2007.

[5]. International Telecommunication Union, Press release. http://www.itu.int/newsroom/press_releases/2010/06. html. Barcelona, 15 February 2010, last accessed on 2011-03-22.

[6]. ICT4RD project: http://www.ict4rd.ne.tz/, last accessed 2011-03-25.

[7]. M. Zennaro, et all: Water Quality Wireless Sensor Network (WQWSN): An Application to Water Quality Monitoring in Malawi. ICPP Workshops, Vienna, Austria, Sept 2009.

[8]. A. Pentland, R. Fletcher, A. Hasson. Daknet: Rethinking Connectivity in Developing Nations. IEEE Computer Society Press, Los Alamitos, CA, USA. 2004.

[9]. Sami Network connectivity Project. http://www.epractice.eu/cases/saminetwork, Last accessed 2011-03-24.

[10]. N4C project, http://www.n4c.eu/. Last accessed 2011-03-22.

[11]. InterPlanetary Internet Project, http://www.ipnsig.org/home.htm , last accessed on 2011-03-21.

[12]. F. Warthman, Delay-Tolerant Networks (DTNs): A tutorial.