MELHORANDO O ENCAMINHAMENTO DE MENSAGENS EM REDES D2D COM MÚLTIPLOS SALTOS

MICHAEL DOUGRAS DA SILVA

MELHORANDO O ENCAMINHAMENTO DE MENSAGENS EM REDES D2D COM MÚLTIPLOS SALTOS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Antônio Alfredo Ferreira Loureiro

Belo Horizonte - Minas Gerais
Junho de 2018

MICHAEL DOUGRAS DA SILVA

IMPROVING D2D MULTI-HOP MESSAGE FORWARDING

Dissertation presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Antônio Alfredo Ferreira Loureiro

Belo Horizonte - Minas Gerais

June 2018

da Silva, Michael Dougras

Improving D2D Multi-Hop Message Forwarding / Michael Dougras da Silva. — Belo Horizonte - Minas Gerais, 2018
xxi, 19 f. ; 29cm

Dissertação (mestrado) — Federal University of Minas Gerais
Orientador: Antônio Alfredo Ferreira Loureiro

1. d2d. 2. multi-hop. 3. network. 4. forwarding. I. Título.

[Folha de Aprovação]

Quando a secretaria do Curso fornecer esta folha,
ela deve ser digitalizada e armazenada no disco em formato gráfico.

Se você estiver usando o `pdflatex`,
armazene o arquivo preferencialmente em formato PNG
(o formato JPEG é pior neste caso).

Se você estiver usando o `latex` (não o `pdflatex`),
terá que converter o arquivo gráfico para o formato EPS.

Em seguida, acrescente a opção `approval={`*nome do arquivo*`}`
ao comando `\ppgccufmg`.

Se a imagem da folha de aprovação precisar ser ajustada, use:
`approval=[`*ajuste*`][`*escala*`]{`*nome do arquivo*`}`
onde *ajuste* é uma distância para deslocar a imagem para baixo
e *escala* é um fator de escala para a imagem. Por exemplo:
`approval=[-2cm][0.9]{`*nome do arquivo*`}`
desloca a imagem 2cm para cima e a escala em 90%.

Dedication text goes here.

Acknowledgments

Here goes aknowledgement text. texto muito importante

"Not everything that counts can be counted, and not everything that can be counted counts."

(William Bruce Cameron - "Informal Sociology: A Casual Introduction to Sociological Thinking")

Abstract

Abstract text goes here.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Mobile devices have become smaller and more powerful over the years. Smartphone based computing and communication has become ubiquitous. Connected services and applications like social networks, instant messaging apps, content distribution systems and games, for example, have imposed several traffic growth in the mobile network. To support this demand we have been improving the network infrastructure. In indoor environments this strategy is feasible, because usually they are smaller and we have more control over the number of devices. However when we are dealing with outdoor environments this approach can be unfeasible, because in this case we have a dynamic scenario in which the number of devices and consequently the traffic demand can be unpredictable.

In the current mobile communication model devices must contact a base station to communicate with other devices or services. Recent works have explored a new communication model in which mobile devices can bypass the base station to communicate directly with near devices. This model is called Device to Device communication or D2D [Yang et al., 2013]. This approach has several applications, for example offload traffic from base stations [Yang et al., 2013, Nunes et al., 2016c, Aijaz et al., 2013, Pyattaev et al., 2013, Andreev et al., 2014, Bastug et al., 2014], to provide proximity based services [Lin et al., 2014a] and to provide extended network coverage under emergency scenarios [Babun et al., 2015].

The D2D communication shows itself as a sensible solution to evolve the mobile network system and improve the devices experience. However there are some characteristics of this scenario that makes the regular Internet communication protocols unapplicable. One of the major characteristics is the absence of end-to-end path between two nodes. This occurs because messages are transmitted when there are contacts between two or more nodes, and these contacts are defined by people mobility. This fact by itself makes the IP protocol routing mechanism unapplicable, which makes the routing problem one of the major topics of study [Misra et al., 2016]. Other characteristic is the intermittently connections also caused by nodes mobility. In the regular IP protocol nodes after receiving a packet immediately forwards it to the next hop based on its routing table. Due to intermittently connections, nodes can't forward the message immediately in some cases.

These problems were discussed and solutions were proposed in the context of Disruption Tolerant Networks (DTNs) [Fall, 2003]. Regular networks are based on the *store-and-forward* paradigm, in which nodes after receiving a message immediately forward it to other connected node that can help to deliver the message to the destiny. DTN networks define the *store-carry-and-forward* paradigm, in which a node after receiving a message, it stores it in a persistent buffer, and carries it until there is a proper contact to forward the message.

The *store-carry-and-forward* paradigm solves the primary problem of communication, making it possible to transmit a message from a node to another, however there is still some problems to solve. One of the major problems is the connection establishment. In D2D networks the communication process is highly dependent of nodes cooperation and resource sharing. Regarding the connection establishment there is a discussion of the best interface to use. Some works propose the spectrum sharing between D2D connections and regular mobile network connections [Lin et al., 2014b]. The major benefit of using this approach is the higher network performance and the major challenge is to define a good spectrum allocation mechanism to evict communication interference between closer devices and base stations. Some works propose the use of secondary interfaces like WIFI and bluetooth [Mao et al., 2015]. The major benefit of this approach is the reduced interference with closer connections and simplified connection establishment mechanism. The major problem is the reduced network performance.

With a defined communication interface and the data transmission paradigm *store-carry-and-forward* it is possible to forward messages in the network using only D2D communications. However there still an interesting problem to solve: the routing protocol. As said earlier in this scenario there is no predefined end-to-end path between two nodes, so the regular IP routing mechanism does not work [Misra et al., 2016]. There are several proposals to solve this problem, in which the major solutions are based on flooding variations, probability functions and social context exploration. We will discuss the major solutions in depth in the next chapter. One thing to notice is that the algorithms based on utility functions (probability and social context based strategies, mainly) tend to concentrate the network traffic in some nodes with higher utility values, penalizing them [Chilipirea et al., 2013]. This is a remarkable problem, because in networks that use the *store-carry-and-forward* paradigm nodes need to allocate a buffer to store messages until there is chance to forward it. If the traffic is too high, some nodes can have problems of buffer overflow [Silva et al., 2015].

This work has two major contributions. First we propose a new buffer management strategy called *Space Time Drop* or just ST-Drop that aims to solve the problem of buffer management under high traffic demands. The proposed solution shows a notable performance when combined with social aware and probability based routing algorithms, outperforming classic approaches. The second contribution is the definition of a distributed implementation of a new social aware routing algorithm called Groups-NET proposed in [Nunes et al., 2016c]. We propose a distributed algorithm for detecting and manage groups. The initial experiments show that the algorithm outperforms the BubbleRap algorithm on network overhead metric with compatible delivery ratio. The work is organized as follows. In chapter 2 we discuss the main approaches and algorithms to forward messages in D2D networks. This chapter introduce important concepts to understand the proposed solutions. In chapter 3 we discuss the classic approaches of buffer management and introduce the proposed algorithm ST-Drop. In chapter 4 we introduce the distributed implementation of the Groups-NET algorithm and present some initial results. And in chapter 5 we conclude the work with a summary of the main results and discuss future works to expand the research.

Chapter 2

Message Forwarding D2D Networks

In D2D networks messages are forwarded using intermediate nodes as relays through the *store-carry-and-forward* paradigm. A node after receiving a message, stores it in a persistent buffer until there is a proper contact to forward the message. The big problem here is how to define when to forward a message upon a contact. If too many copies of a message are generated, the network performance degrades because nodes energy and memory consumption get high. With fewer message copies, fewer paths are explored, so in general the delivery probability degrades as well because there is no predefined end-to-end path between two nodes. This chapter presents the main forwarding protocols proposed to deal with this problem.

Forwarding protocols built on top of D2D networks can be classified in single-hop and multi-hop protocols. Single-hop protocols uses only the direct contacts of a node to provide services. Multi-hop protocols use intermediate nodes to expand the service coverage. This work focus on multi-hop protocols mainly, but in this chapter we take a quick overview of single-hop protocols too.

2.1 Single Hop Protocols

Single hop protocols eliminate the routing problem by exploring only the direct neighbors of a node. Protocols in this category usually explore people group or cluster formation to provide local services for data offloading. One example of solution that uses this approach is a protocol called WiGroup proposed in [Wang et al., 2015]. This protocol defines user groups in which there is a group owner that is responsible for connecting to the base station and actuate as an intermediate access point. The other members of the group uses the group owner to access network service. A similar solution is proposed in [Zheng et al., 2014] which uses social relationship analysis in the group formation algorithm.

The major benefits of these solutions are the reduced number of connections to the base station and it is possible to define cache strategies at group level. Single hop protocols are a sensible solution for data offloading. They are specially good for using in events with high number of people in the same area, like in stadiums and big concerts [Wang et al., 2015]. However, in environments with high mobility they do not perform well, because in this scenario groups are unstable. To handle theses scenarios solutions based on multi-hop protocols are recommended.

2.2 Multi Hop Protocols

Multi hop protocols use intermediate nodes to forward messages in the network. These protocols can be used for *unicasting* communication, in which a message is forwarded from a source to a single destination node. And can also be used for *multicasting* communication, in which a message is forwarded from a source to multiple destination nodes. The major multi hop protocols were defined in the context of Disruption Tolerant Networks (DTNs). Forwarding algorithms usually explore the unicasting communication to validate the solution, because if unicasting works well, the protocol can be extended to work in multicasting mode [Misra et al., 2016].

Multi hop protocols can be classified as single copy and multi copy. Single copy protocols, as the name suggests, keeps just one copy of a message in the network as in classic networks. First Contact and Direct Delivery are the major protocols in this category [Misra et al., 2016]. In the First Contact protocol messages are forwarded every time a node get in contact with other nodes. The message is forwarded until it reaches the destination. After forwarding a message the node drops the local copy. In Direct Delivery a node carries a message until it get in contact directly with the destination node. These protocols are too simple and shows low performance in dynamic scenarios.

Multi copy protocols create multiple copies of each message to explore multiple paths. This approach is sensible, because since there isn't a predefined path, exploring multiple paths can improve delivery ratio. However creating a high number of message copies in the network will increase nodes' power and memory consumption, what in practice can degrade the network performance. So a good forwarding algorithm should achieve a good balance of delivery ratio and message copies overhead. Message delivery is not guaranteed is this scenario, so the delivery ratio refers to the percentage of created messages delivered to the destination. The message overhead refers to the proportional number of copies per created message. It is also important to notice that multi hop protocols are designed to work with content that does not have strict delivery time constraints. Traditional networks have acceptable delivery time of few milliseconds, in contrast in D2D multi hop we consider delivery times of some days, a week or higher depending on the used approach.

The Epidemic protocol is the simplest multi copy algorithm [Vahdat et al., 2000]. It uses the flooding strategy, in which at each encounter nodes exchange all messages they have. This protocol achieves the best possible delivery ratio, however it also has the worst message overhead, which in turns makes it unfeasible for real scenarios. There some approaches derived from epidemic that uses a controlled flooding strategy. The best example is the Spray and Wait protocol [Spyropoulos et al., 2005], in which the source node forwards a predefined number of message copies during its contacts. Nodes that received the message them uses the direct delivery approach to forward the message to the destiny. There is also a variation called Binary Spray and Wait [Spyropoulos et al., 2005] in which the source node starts with L messages. At each encounter the source forwards $\frac{L}{2}$ copies until there is only 1 copy left, then it uses the direct delivery strategy. Other nodes takes the same approach of the source, but they start with the number of copies received. Both approaches show good delivery ratio with low overhead, however the authors discuss that this approach have low performance in scenarios with low mobility, because nodes with only a copy will carry the message until they encounter with the destination. So they propose a modified version called Spray and Focus [Spyropoulos et al., 2007], in which nodes with only one copy left can forward the message like a single copy protocol, dropping the local copy. The forwarding decision is based on an utility function that uses a timer that counts the last time a node had a contact with the destination. The authors have shown that this Spray and Focus has better performance than Spray and Wait in scenarios with lower mobility.

More advanced protocols reduce the number of message copies adding a decision mechanism to define if a node should send a message copy or not when an encounter occurs. The Prophet protocol is a good example of such protocols [Lindgren et al., 2003]. Prophet uses nodes' contacts history to measure the probability of a given node to deliver the message. When two nodes get in contact, they will exchange only the messages for which the other node has a greater delivery probability. The basic idea in Prophet is to assign higher probabilities of meeting again to pairs of nodes that have met more recently. Through experiments the authors show that Prophet outperforms previous solutions.

In recent years, researchers have been exploring social aware forwarding strategies. The idea is that, since contacts are driven by human mobility, if we can understand social relationship patterns we can improve protocols. A good example of such algorithms is Bubble Rap [Hui et al., 2011]. It is a social-aware algorithm that exploits the concept of community and network node's popularity. In this algorithm, each node is assigned to at least one social community. Social communities are defined as sets of more densely interconnected nodes (groups of nodes that have more contacts among themselves). The popularity of the nodes is measured by the number of distinct contacts a given node in the network has along the time. The basic idea of Bubble Rap is to forward the message to nodes with greater global popularity until it reaches a member of the destination communities. Then the message is forwarded based on the local popularity (node popularity considering only members of the destination community) until it reaches the destination node. Through experiments the authors show that Bubble Rap outperforms previous solutions including the Prophet protocol. Other recent social aware protocol is the Groups-NET [Nunes et al., 2016b]. This solution explores the idea of regularity of people encounters in groups to forward the messages. This protocol is explored in depth in chapter 4.

Chapter 3

Buffer Management Strategies

Chapter 4

Message Forwarding in D2D Networks

In this chapter we present an overview of message forwarding algorithms for D2D networks. The first section presents the main contributions to the topic, discussing the state-of-art forwarding algorithms. Then we introduce an implementation proposal of Groups NET algorithm [add-reference] in a distributed environment. The implementation detail is explained in two parts: the first covers the group detection algorithm and the second the forwarding implementation itself.

4.1 Introduction

Introduce the challenge of forwarding messages in this scenario of dynamic nodes movement.

The main solutions were initially proposed in the context of DTNs (Delay Tolerant Networks) or are derived from them, since the two types of networks share some intrinsic characteristics.

Add a brief description about message forwarding in D2D.

Add description for the following algorithms:

* Epidemic routing * Prophet * Bubble Rap * Original Groups NET

4.2 The Distributed Groups NET

In this section we propose a distributed implementation of Groups NET forwarding algorithm. The solution is comprised of two stages: The first stage is group detection, in which nodes use neighborhood discovery to infer social group meeting using a local algorithm. The second stage is message forwarding, in which nodes use the generated group data to decide when to forward a message or not. The following sections describe in details the two stages.

4.2.1 Mobile Group Detection

The mobile group detection algorithm proposed here is derived from the ideas presented in [Nunes et al., 2016a]. A group is defined as a set of people that get in contact on a given point of space and time. The authors show that group encounters have some level of regularity along the time (mainly at daily and weekly basis). This property can be used by forwarding algorithms.

In the original proposal the authors use a approach to detect groups based on global network knowledge. They basically build a contact graph considering a predefined time interval, and for each graph is used the Clique Percolation community detection algorithm. Each community detected by the algorithm is considered a group meeting in that time window. Then a metric called *Group Correlation Coefficient* is used to detect instances of the same group across multiple time windows.

With this approach the authors show that groups usually have regular encounters along the time (specially in daily and weekly basis). But they also discuss that the detection approach is unfeasible for a distributed environment, because it assumes a global network knowledge. However, authors also discuss that implementing a distributed solution for this should be a simple task, because in the local scope nodes can use neighborhood discover strategies and process regular neighbors encounters to decide their group encounters.

In this work we expanded this idea to build a distributed solution for group detection.It is composed of four processes that are executed concurrently by each node using basically neighborhood inspection and some basic additional parameters. In the following sections each process is explained.

4.2.1.1 Device's Local Group Detection

This process is responsible for processing nodes' local contacts and decide when a group meeting happens. The algorithm is very simple. Each node keeps two lists of devices. The first is the *friends* list (lets call it FL) that keeps track of current nodes considered as friends, i.e, members of a group that have a recent encounter. The second list is the *strangers* list (lets call it SL), that keeps track of recent nodes' contacts that are not yet considered as friends. Each entry in these lists have a contact counter that keeps track of consecutive contacts with that node, and also have a inactive counter that keeps track of consecutive periods of time without contact with that node.

The node inspects its neighborhood at each predefined *time interval* and based on current neighbors it updates the friends and strangers list. This update is based on two predefined parameters: the *friend threshold* and *inactive threshold*. Lets call these parameters as FT and IT respectively. Based on these parameters the following steps are executed:

- At each time interval a node collects its current neighbors (lets call it CN).
- The first step is to update the counters of friends list. For each current neighbor that is present in the friends list, we increase the contact counter and reset the inactive counter. For each node present in the friends list that is not a current neighbor we reset the contact counter and increase the inactive counter.
- The second step is to update the strangers list. For each current neighbor that is present in the strangers list we reset the inactive counter and increase the contact counter. If the contact counter reaches the friend threshold the node is promoted to friends list. For each node that is present in strangers list but is not a current neighbor we reset the contact counter and increase the inactive counter. If the inactive counter reaches the inactive threshold the node is removed from the strangers list. Nodes that are current neighbors and are not present in both friends and strangers list are added in the strangers list with contact counter equals 1.
- As a final step, we check the number of nodes in the friends list that are considered inactive. A node is considered inactive when the inactive counter reaches the inactive threshold. If more than 50 percent of the nodes in the friends list is inactive, the friend list is archived and considered as a group meeting that happened in the past. When this happens we clear the friends and strangers list and restart the algorithm.

In the end this algorithm creates a list of local detected group meetings. This list is used by the local group combination process to detect instances of the same group with multiple encounters along the time.

4.2.1.2 Local Group Combination

In this step each mobile device process groups discovered in the previous step to discover groups that have multiple encounters along the time. The algorithm is based on a metric called *Group Correlation Coefficient* defined in [add-reference]. This metric basically computes the proportion of nodes shared between two sets.

Each node keeps a list of *combined groups*, that keeps track of groups with multiple encounters along the time. For each local group detected in the previous process, we check if there is a combined group with Group Correlation Coefficient greater or equals 0.5. If there is, the local group is combined into the combined group adding its members to it and registering a new encounter. If there isn't, a new combined group is initialized with the local group data.

The combined groups generated in this process are used by neighborhood inspection to compose the final groups considered in the forwarding algorithm.

4.2.2 Neighborhood Inspection

Besides inspecting its current neighborhood, at each considered time interval each node also collects its neighbors combined groups. The collected groups are combined with the groups generated by the local group combination process using the same approach described previously. Groups that have correlation coefficient greater of equals 0.5 are merged. This step is used to expand a node local vision with its neighbor network knowledge. So, in the end the local combined groups are updated with neighbors detected group information.

4.2.2.1 Group Graph Creation

In this step we build the graph of groups using the combined groups created by the neighborhood inspection process. The graph is created using the approach described in [Nunes et al., 2016a]

Chapter 5

Conclusion

Put some text here

Bibliography

[Aijaz et al., 2013] Aijaz, A., Aghvami, H., and Amani, M. (2013). A survey on mobile data offloading: technical and business perspectives. *IEEE Wireless Communications*, 20(2):104--112.

[Andreev et al., 2014] Andreev, S., Pyattaev, A., Johnsson, K., Galinina, O., and Koucheryavy, Y. (2014). Cellular traffic offloading onto network-assisted device-to-device connections. *IEEE Communications Magazine*, 52(4):20--31.

[Babun et al., 2015] Babun, L., Yürekli, A. İ., and Güvenç, I. (2015). Multi-hop and d2d communications for extending coverage in public safety scenarios. In *Local Computer Networks Conference Workshops (LCN Workshops), 2015 IEEE 40th*, pages 912--919. IEEE.

[Bastug et al., 2014] Bastug, E., Bennis, M., and Debbah, M. (2014). Living on the edge: The role of proactive caching in 5g wireless networks. *IEEE Communications Magazine*, 52(8):82--89.

[Chilipirea et al., 2013] Chilipirea, C., Petre, A.-C., and Dobre, C. (2013). Energy-aware social-based routing in opportunistic networks. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pages 791--796. IEEE.

[Fall, 2003] Fall, K. (2003). A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27--34. ACM.

[Hui et al., 2011] Hui, P., Crowcroft, J., and Yoneki, E. (2011). Bubble rap: Social-based forwarding in delay-tolerant networks. *IEEE Transactions on Mobile Computing*, 10(11):1576--1589.

[Lin et al., 2014a] Lin, X., Andrews, J., Ghosh, A., and Ratasuk, R. (2014a). An overview of 3gpp device-to-device proximity services. *IEEE Communications Magazine*, 52(4):40--48.

[Lin et al., 2014b] Lin, X., Andrews, J. G., and Ghosh, A. (2014b). Spectrum sharing for device-to-device communication in cellular networks. *IEEE Trans. Wireless Communications*, 13(12):6727--6740.

[Lindgren et al., 2003] Lindgren, A., Doria, A., and Schelén, O. (2003). Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE mobile computing and communications review*, 7(3):19--20.

[Mao et al., 2015] Mao, Z., Wang, X., and Jiang, Y. (2015). Contact-duration aware transmission scheduling in wifi direct enabled mobile social networks. In *2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 1581--1586. IEEE.

[Misra et al., 2016] Misra, S., Saha, B. K., and Pal, S. (2016). Opportunistic mobile networks.

[Nunes et al., 2016a] Nunes, I., Vaz de Melo, P., and A.F. Loureiro, A. (2016a). Group mobility: Detection, tracking and characterization. In *IEEE ICC 2016 International Conference on Communications (ICC'16 SAC-8 SN)*, Kuala Lumpur, Malaysia.

[Nunes et al., 2016b] Nunes, I. O., Celes, C., de Melo, P. O., and Loureiro, A. A. (2016b). Groups-net: Group meetings aware routing in multi-hop d2d networks. *arXiv preprint arXiv:1605.07692*.

[Nunes et al., 2016c] Nunes, I. O., de Melo, P. O. S. V., and Loureiro, A. A. F. (2016c). Leveraging d2d multihop communication through social group meeting awareness. *IEEE Wireless Communications*, 23(4):12–19. ISSN 1536-1284.

[Pyattaev et al., 2013] Pyattaev, A., Johnsson, K., Andreev, S., and Koucheryavy, Y. (2013). Proximity-based data offloading via network assisted device-to-device communications. In *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pages 1--5. IEEE.

[Silva et al., 2015] Silva, A. P., Burleigh, S., Hirata, C. M., and Obraczka, K. (2015). A survey on congestion control for delay and disruption tolerant networks. *Ad Hoc Networks*, 25:480--494.

[Spyropoulos et al., 2005] Spyropoulos, T., Psounis, K., and Raghavendra, C. S. (2005). Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252--259. ACM.

[Spyropoulos et al., 2007] Spyropoulos, T., Psounis, K., and Raghavendra, C. S. (2007). Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *null*, pages 79--85. IEEE.

[Vahdat et al., 2000] Vahdat, A., Becker, D., et al. (2000). Epidemic routing for partially connected ad hoc networks.

[Wang et al., 2015] Wang, Y., Zhang, H., Tan, C. C., and Sheng, X. D. (2015). {WiGroup}: A lightweight cellular-assisteddevice-to-device network formation framework.

[Yang et al., 2013] Yang, M. J., Lim, S. Y., Park, H. J., and Park, N. H. (2013). Solving the data overload: Device-to-device bearer control architecture for cellular data offloading. *IEEE Vehicular Technology Magazine*, 8(1):31--39.

[Zheng et al., 2014] Zheng, Z., Wang, T., Song, L., Han, Z., and Wu, J. (2014). Social-aware multi-file dissemination in device-to-device overlay networks. In *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, pages 219--220. IEEE.