IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

An Efficient Selfishness Aware Routing in Delay Tolerant Networks

N.Senthilkumar[*1], Dr. T.V U. Kiran Kumar[2]

[*1,2,3,4] Bharath University, Chennai, India

senthil.analyst86@gmail.com

Abstract

Delay Tolerant Networks (DTNs) enable data transfer when mobile nodes are onlyintermittently connected. DTN routing usually follows store-carry-and-forward mechanism. Therefore,the willingness of nodes to relay messages for other nodes plays a significant role in the routing process.Moreover, since the resources in mobile devices are generally limited, carriers of mobile devices maybe unwilling to relay messages for other nodes in order to conserve their scarce resources. Whenconsidering routing in DTNs, such selfish nodes have to be considered. Existing routing algorithmsdetects routing misbehavior can be caused by selfish nodes that are unwilling to spend resources suchas power and buffer on forwarding packets of others, or caused by malicious nodes that drop packets tolaunch attacks. To mitigate routing misbehavior by limiting the number of packets forwarded to themisbehaving nodes. Many challenges considers of selfishness of users who prefer to relay data forothers with strong social ties. Such social selfishness of users is a new constraint in network protocoldesign. Proposed work carries Social Selfishness Aware Routing (SSAR) algorithm to allow userselfishness and provide better routing performance in an efficient way. SSAR considers both users'willingness to forward and their contact opportunity, resulting in a better forwarding strategy.

Keywords: Detection, Disruption Tolerant Networks, Mitigation, Routing Misbehavior, Security.

Introduction

Delay Tolerant Networking (DTN) is an approach to computer network architecture that aims to address the technical issues in heterogeneous networks that experience lack of continuous network connectivity. DTN enable data transfer when mobile nodes are only intermittently connected. Due to lack of consistent connectivity, DTN routing usually follow store-carry-and-forward. In Delay Tolerant Networks, selfish or malicious no desmay drop received packets. Selfishness in our context can be expressed in two ways. Firstly nodes may deny copying and storing data, which are of no interest to them and destined to a third node. Secondly even if they accept to acquire such data, they may refuse to infect another node with them, i.e. relay data to other nodes. To address the problem, this work has a distributed scheme to detect packet dropping. In this work, a node is required to keep a few signed contact records of its previous contacts, based on which the next contacted node can detect if the node has dropped any packet.

Since misbehaving nodes may misreport their contact records to avoid being detected, a small part of each contact record is disseminated to a certain number of witness nodes, which can collect the appropriate contact records and detect them is behaving nodes. This scheme also consists of to mitigate routing misbehavior by limiting the number of packets forwarded to them is behaving nodes. Proposed SSAR is carried out to identify the nodes and performs efficient SSAR routing with its bandwidth capacity, contact opportunity, node willingness.

Mitigation Misbehavior

There are two types of nodes: misbehaving nodes and normal nodes. A misbehaving node drops the received packet seven if it has available buffers, but it does not drop its own packets. It may also drop the control messages of our detection scheme. Let us assume a small number of misbehaving nodes may collude to avoid being detected, and they may synchronize their actions via out-band communication channels. A normal node may drop packets when its buffer overflows, but it follows our protocol. In some DTN applications, each packet has a certain life time, and then expired packets should be dropped whether or not there is buffer space. Such dropping can be identified if the expiration time of the packet is signed by the source. Such dropping is not misbehavior.

To detect misreporting, the contacted node also randomly selects a certain number of witnesses for the reported records and sends a summary of each reported record to them when it contacts them. When

two nodes contact, they generate a contact record which shows when this contact happens, which packets are in their buffers before data exchange, and what packets they send or receive during the data exchange. The record also includes the unique sequence number that each of them assigns for this contact. The record is signed by both nodes for integrity protection. A node is required to carry the record of its previous con tact, and report the record to its next contacted node, which will detect if it has dropped packets since the previous contact.

The misbehaving node in Figure 1.1 is required to generate a contact record during each contact and report its previous contact record to the contacted node. Based on there ported contact records, the contacted node detects if the misbehaving node has dropped packets. The misbehaving node may misreport to hide its misbehavior, but forged records cause inconsistencies which make misreporting detectable.

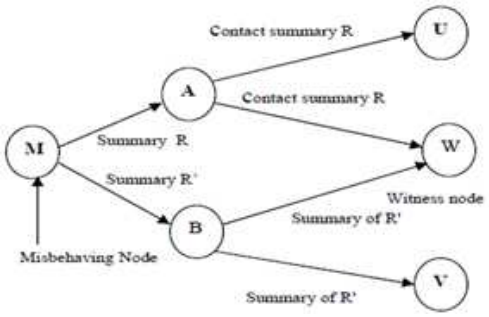


Figure 1.1:Misbehaving node Detection.

To detect misreporting, the contacted node also randomly selects a certain number of witness nodes for the reported records and sends a summary of each reported record. The witness node W that collects two inconsistent contact records can detect them is reporting node nodes for the reported records and sends a summary of each reported record to them when it contacts them. The witness node that collects two inconsistent contact records can detect the misreporting node.

Misreporting Node Detection

To hide the dropping from being detected by the next contacted node, will not report the true record of the previous contact. However, when there is no collusion, cannot modify the true record since it is signed by the previous contacted node. Also, cannot forge a contact record because it does not know the private key of any other node. Thus, the only misreporting it can perform is to replay an old record generated before the previous contact. This misreporting is referred to as replay record. Note that other types of misreporting are possible when collusions exist.

Detection

To detect the inconsistency caused by misreporting, for each contact record generated and received in a contact, a node selects W random nodes as the witness nodes of this record, and transmits the summary of this record to them when it contacts them. It selects the witness nodes from the nodes that it has directly contacted. In this manner, each node can collect some record summaries for which its a witness. When it receives a new summary, it checks the summary against the already collected summaries signed by the same node to see if the signer has violated any of the two consistency rules. If a violation is detected, then it further verifies the signatures included in the inconsistent summaries. If the signature verification succeeds, the signer is detected as misreporting.

ALARM

After detection, the witness node floods an alarm to all other nodes. The alarm includes the two inconsistent summaries. When a node receives this alarm, it verifies the inconsistency between the included summaries and the signature of the summaries. If the verification succeeds, this node adds the appropriate misreporting node into a blacklist and will not send any packets to it. If the verification fails, the alarm is discarded and will not be further propagated. A misreporting node will be kept in the blacklist for a certain time before being deleted. In this each node detects packet dropping locally based on the collected information. Moreover, the detection scheme an effectively detect misreporting even when some nodes collude.

To mitigate routing misbehavior, we tryto reduce the number of packets sent to the misbehaving nodes. If a node is detected to be misreporting, it should be blacklisted and should not receive packets from others. However, if a misbehaving node does not misreport, we cannot simply blacklist it because it is dropping packets, since a normal node may also drop packets due to buffer overflow. In the following, we focus on how to mitigate routing misbehavior without affecting normal nodes too much when misbehaving nodes do not misreport. The detection scheme works in distributed way; i.e., each node detects packet dropping locally based on the collected information. Moreover, the detection scheme can effectively detect misreporting even when some nodes collude. Analytical results on detection probability and detection delay were also presented. Based on our packet dropping detection scheme, this scheme to mitigate routing misbehavior in DTNs.

Proposed Work

In the proposed work the selfish node forward packets for nodes with which they have social ties but not others, and such willingness varies with the strength of the social tie. Proposed work carries a Social Selfishness Aware Routing (SSAR) algorithm to cope with user selfishness and provide good routing performance in an efficient way. To select ineffective forwarding node, SSAR considers both users' willingness to forward and their contact opportunity, and derives a metric with mathematical modeling and machine learning techniques to measure the forwarding capability of the mobile nodes. SSAR allows users to maintain selfishness and achieves good routing performance with low transmission cost.

SSAR

As being selfish, they are unwilling to forward packets for those with whom they have no social ties in order to save their own storage and power resources. For convenience, the above social and selfish behavior will be referred to as social selfishness. Social selfishness has not been addressed before. It will not work well since some packets are forwarded to nodes unwilling to relay, and will be dropped. The selfish side of users, where selfish nodes are stimulated to forward packets for all other nodes to maintain high-performance. However, these schemes go to another extreme; i.e., it assumes that a node is not willing to forward packets for anyone else. For convenience, such selfishness is called individual selfishness.

Social Selfishness Aware Routing(SSAR) protocol, in which a node only forwards packets for those with social ties, and it gives priority to packets received from those with stronger social ties when there are not enough resources. Since each node only forwards packets for part of the nodes, it is important to know how this will affect the routing performance. To achieve high performance, SSAR considers both user willingness and contact opportunity when selecting relays. It combines the two factors through mathematical modeling and machine learning techniques, and obtains a new metric to measure the forwarding capability of a relay. With SSAR, a packet will most likely before warded to the relay that has strong willingness to forward as well as high direct or indirect/transitive contact opportunity with the destination. Social selfishness but also try to maintain good routing performance under social selfish behavior. Social selfishness is a kind of user demand that should be satisfied.

Willingness Table

Each node maintains a table that contains its willingness values for other nodes in the network. In this table, each item has the format of [node ID, value i]. The value of willingness is a real number within [0, 1], where0 means unwilling to forward and 1 means the most willing to forward. A node's willingness value for another node depends on the social tie between them. The stronger the social tie is, the larger the willingness is. Node only needs to setits willingness value for each other user with whom it has a social tie, and set a default willingness value (e.g., 0) for all other (possibly unknown) nodes without any social tie.

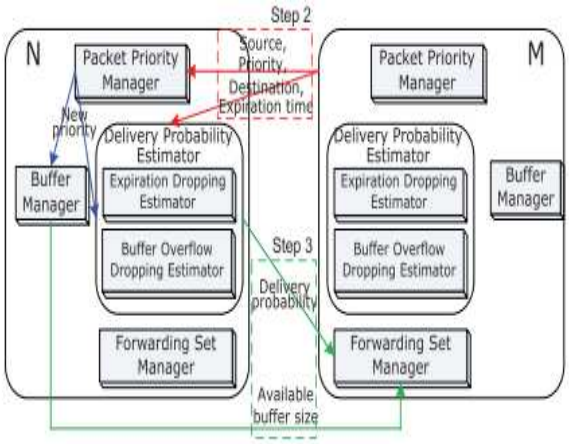
Protocol Description

i). After neighbor discovery, node N and M deliver packets destined to each other in the decreasing order of priority.

ii). Suppose M still buffers some other packets. Then M sends N a summary list of h sourced, destination ID, expiration time, priority for these packets.

iii). From the source ID and priority information, N calculates the new priority value for each packet in the list. Based on the new priority, the destination ID and expiration time, N calculates its delivery probability and available buffer size for each packet, and returns them to M.iv). M determines a candidate set of packets for which N has higher delivery probabilities.

v). considering the available buffer size information, M further decides which candidates to transmit by solving the
MKPAR formulation.



Packet Priority

When a node receives a packet from a previous hop, it assigns a priority to the packet. The priority determines if this node will relaythe packet (i.e., the priority is positive) or not(i.e., the priority is zero). To be socially selfish, the node only forwards the packet if it is from anode with a social tie. There are two cases: First, the source of the packet has a

social tie with this node, and hence forwarding the packet means helping the source. Second, the previous hop has a social tie with this node, no matter the source has a social tie or not.

In this case, the previous hop has taken over the responsibility (probably from its own social tie) to deliver the packet. Thus, even if the source does not have a social tie with this node, this node should still relay the packet to help the previous hop. Actually, this is motivated by the real-world phenomenon that people usually would like to help a friend's friend. The priority should also measure the social importance of the packet to this node. For example, when other conditions are the same, packets from the node with a stronger social tie should have a higher priority.

Let Pcurr denote the new priority of a packet in the current hop, and Pprev denote theold priority of the packet in its previous hop. Let ωsrc and ωprev denote the current hop' s willingness for the packet source and the previous hop, respectively.

Then the current hop calculates the new priority in the following ways (Note that the initial priority of a packet is set as 1 by the source node.):

i). If neither the source nor the previous hophas a social tie with the current hop, then Pcurr = 0.

ii). If the source has a social tie but the previous hop does not, then Pcurr = ωsrc.

iii). If the previous hop has a social tie but the source does not, then Pcurr = Pprev.ωprev.

iv). If both the source and the previous hop have a social tie with the current hop,

v). Pcurr = max {ωsrc, Pprev.ωprev}.

The priority assignment method and the buffer management policy can enforce social selfishness. Packets that traverse different social links will receive different forwarding service. SSAR allows users to behave in the socially selfish way and improves performance by considering user willingness, resource constraints, and contact opportunity when selecting relays. SSAR's gain in packet delivery ratio and cost does not come for free its packet delivery delay is longer, because it does not forward packets to the relays that have a good contact opportunity but low willingness. However, the packets forwarded to these relays also have a high risk of being dropped due to the low willingness.

Buffer Manager

A node manages buffers based on packet priority: (i) Packets with priority 0 will not be buffered; (ii) When buffer overflows, packets of low priority are dropped first. The second rule indicates that a new incoming packet can preempt the buffer occupied by a lower-priority packet. The buffer policy together with the priority assignment method allows nodes to be socially selfish.

Delivery Probability Estimator

It estimates a node's delivery probability for a packet, which is used to mea-sure the node's forwarding capability for that packet. When two nodes are in contact, each packet is for-warded from the node with a lower delivery probability to the node with a higher delivery probability. Traditionally, the quality of a relay is measured solely based on contact opportunity, which can be the relay's direct contact opportunity to the destination node or the transitive contact opportunity provided by the relay's contacted nodes or both. SSAR measures the delivery probability of a node based on both of its contact opportunity to the destination and its willingness to forward. It is straightforward that a node with a low contact opportunity should not be a relay. Interestingly, a node with a high contact opportunity but low willingness should not be a relay either.

Conclusion

In this work we have studied the problem of data dissemination in delay tolerant networks. By considering selfish nodes with social behavior, In this existing work, using summary report the witness node detects packet dropping in DTNs. The detection scheme works in a distributed way. i.e., each node detects packet dropping locally based on the collected information. Moreover, the detection scheme can effectively detect misreporting even when some nodes collude. But this will not route through the selfish node. But the Social Selfishness Aware Routing (SSAR) protocol allows social selfishness and provides better routing performance in an efficient way. SSAR allocates resources such as buffers and bandwidth based on packet priority which is related to the social relationship among nodes. To maintain the routing performance, SSAR will quantify the relay's willingness to evaluate its forwarding capability and thus reduces the packet dropping rate.

References

[1] Bin Chen, Mun Choon Chan. A Credit-Based Incentive System for Disruption Tolerant Network. IEEE Infocom, 2010.

[2] Boldrini, M. Conti, A. Passarella, Content place: social-aware data dissemination inOpportunistic networks, in: Proc. MSWiM,2008.

[3] E. Bulut and B. K. Szymanski, "Friendship Based Routing in Delay Tolerant Mobile Social Networks," in Proc. of IEEE GLOBECOM, 2010.

[4] E. Daly and M. Haahr, "Social Network Analysis for Routing in Disconnected

Delay-Tolerant MANETs, ” Proc. ACM Mobi Hoc, 2007.

[5] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, “Maxprop: Routing for Vehicle based disruption Tolerant Networks,” in Proc. IEEE INFOCOM,2006.

[6] K. Fall, “A Delay-Tolerant Network Architecture for Challenged Internets,” in conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM ’03.

[7] K. Scott, S. Burleigh, ”Bundle Protocol Specification,” IETF Network Working Group, November 2007.[8].Lindgren, A. Doria, and O. Schelen, Probabilistic routing in intermittently connected networks, ACM SIGMOBILECCR, 2003.

[8] Mei and J. Stefa, “Give2get: Forwarding in social mobile wireless networks of selfish individuals,” in Distributed Computing Systems (ICDCS), 2010 IEEE 30thInternational Conference on, 2010,

[9] S. Jain, et al., “Routing in Delay Tolerant Network,” ACM SIGCOM’04 , Portland, Oregon, 2004